Forest Fire Monitoring Using Multiple Unmanned Air Vehicles

Timothy McLain  
*Mechanical Engineering Department, Brigham Young University, mclain@byu.edu*

Randal W. Beard  
*Department of Electrical Engineering, Brigham Young University, beard@ee.byu.edu*

*See next page for additional authors*

Follow this and additional works at: [https://scholarsarchive.byu.edu/facpub](https://scholarsarchive.byu.edu/facpub)

Part of the [Mechanical Engineering Commons](https://scholarsarchive.byu.edu/facpub)

**Original Publication Citation**  

**BYU ScholarsArchive Citation**  
McLain, Timothy; Beard, Randal W.; Holt, Ryan S.; Egbert, Joseph W.; Bradley, Justin M.; and Taylor, Clark N., "Forest Fire Monitoring Using Multiple Unmanned Air Vehicles" (2006). All Faculty Publications. 1941.  
[https://scholarsarchive.byu.edu/facpub/1941](https://scholarsarchive.byu.edu/facpub/1941)

This Conference Paper 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.
FOREST FIRE MONITORING USING MULTIPLE UNMANNED AIR VEHICLES

Ryan S. Holt, Joseph W. Egbert, Justin M. Bradley, Randal W. Beard*, Clark N. Taylor, and Timothy W. McLain

College of Engineering and Technology
Brigham Young University
Provo, UT, 84602
*Corresponding author: beard@ee.byu.edu

ABSTRACT

The ability to gather and process information on the condition of forest fires is essential to fighting the fires in a cost-effective, safe, and efficient manner. While high-altitude, long-endurance (HALE) unmanned air vehicles (UAVs) are currently used for fire surveillance; they are an expensive and scarce resource. As a proposed alternative, low-altitude, short-endurance (LASE) UAVs offer lower costs, quicker response times, and high-resolution information. In recent years, advances in solid-state sensor and autopilot technology have made LASE UAVs a feasible alternative. This paper overviews a current research project conducted by Brigham Young University and NASA Ames. In the project, we are developing a solution to the fire monitoring problem where multiple LASE UAVs autonomously coordinate their efforts to provide real-time information on the state of the fire. We will overview our experimental UAV platform, describe a potential concept of operations using multiple LASE UAVs, and discuss recent experimental results.

1. INTRODUCTION

The ability to gather and process information on the condition of forest fires is essential to fighting the fires in a cost-effective, safe, and efficient manner. Fire crews at the site benefit by more detailed and up-to-date information concerning the current state of the fire. They are able to manage the fire quicker and in a safer manner. In some areas, low-orbiting satellite images have been utilized in the detection of forest fires. However, the low-resolution images taken from these satellites provide little or no information other than the location of the fire. Additionally, the time it takes for a satellite to orbit around and update the image is approximately 9 hours. Therefore, these images are not very useful to those actually fighting the fire. However, the idea of using imagery to help combat forest fires could be very effective if the resolution of the images was increased and the time between updates was reduced.

Recently, efforts have been made to employ UAVs, such as the ALTAIR, in the information collecting process (Ambrosia, Wegener, Brass, & Schoenung, 2004). These platforms fly at high altitudes while using high-precision sensors to gather information concerning the state of the fire. The information is then transmitted miles away where it is processed and relayed to ground crews. High-resolution images are delivered to the fire crews with a delay of a few minutes. Fire crews are able to analyze the conditions of the fire from the images and determine the safest and most efficient approach to combating it. HALE UAVs are a significant improvement to satellite images. However, HALE systems are expensive and few in number, restricting the availability of such systems during peak fire season. These limitations encourage the development of other platforms that are smaller, lower in cost, and more readily available.

Low-altitude, short-endurance (LASE) UAVs are able to gather high-resolution images by flying close to the fire’s edge. Because of low costs, their availability is much greater than the HALE systems, making them ideal for forest fire situations. In addition, multiple LASE systems can be deployed on a single mission to cooperatively provide
real-time data to the firefighters. They share the task of collecting information about the fire, quickly delivering the data in an effective format to fire crews along the fire’s perimeter. The information reveals where the firefighters should be and informs them to retreat if conditions are hazardous. Because of their small size, LASE systems can be carried in a backpack or stored in the back of a truck, making them available for quick deployment to begin data acquisition.

NASA is currently working on research projects to implement fire monitoring using LASE UAVs (NASA-Ames Research Center, 2000). However, a number of challenges need to be solved before LASE systems can be used to monitor fires. In (Casbeer et al., 2005), some of these challenges are presented. First, the UAVs should be able to adapt their flight path to changes in the fire perimeter using limited real-time information. Second, in order to compensate for the limited fuel supply, UAVs must know when to return for refueling. Moreover, to increase the information update rate, UAVs need to work cooperatively. One additional challenge not introduced in (Casbeer et al., 2005) is the formatting and presentation of the gathered images to make them useful to firefighters.

The Multiple Agent Intelligent Cooperation and Control (MAGICC) Lab at Brigham Young University has a main research focus on developing LASE UAVs and formulating algorithms for intelligent and autonomous flight. The Lab has created a stable experimental platform where algorithms and designs can be tested and refined. Many of the developed algorithms have application to the fire monitoring problem.

In this paper we present an effective concept of operations for LASE UAVs to be used in monitoring forest fires. In order to demonstrate the effectiveness of our algorithm, we extend the fire-tracking problem to the similar road-following problem. This allows us to test the algorithm in real-world scenarios. We introduce a new cooperative control algorithm utilizing multiple UAVs to monitor the perimeter of a fire. By using multiple agents, we are able to decrease the latency of the information - the time it takes to deliver the data to the base station - and increase the update rate. We also propose a method for storing and presenting the gathering information that limits the amount of storage space needed. The entire concept of operations builds upon the experimental platform and its existing capabilities developed by the MAGICC Lab.

The paper is organized as follows: Section 2 briefly overviews the hardware and software of the experimental platform and describes the existing capabilities developed by the MAGICC Lab. In Section 3, we explain the concept of operations, breaking the solution into three parts: fire perimeter tracking, cooperative team tracking, and information formatting. We present our simulation and hardware results in Section 4. In Section 5, we conclude the paper with comments on future work.

2. OVERVIEW OF EXPERIMENTAL PLATFORM

This section overviews the platform used by the MAGICC Lab to test algorithms. The platform can be broken down into two areas: hardware and software. The MAGICC Lab had made many developments in the area of UAVs both is hardware and software. The hardware consists of an airframe with an autopilot and sensors. The software includes a program for simulation and a program for real-world flight. We briefly overview each area and then describe some of the technologies developed by the lab that are pertinent to solving the fire-monitoring problem. The flow chart of the interaction amongst the hardware is shown in Figure 1.

Hardware

The airframe used is a Zagi design with a wingspan of 4 feet and weight of 11.5 ounces. With this design, the agent can fly at a cruise speed of 13 m/s with a flight time of 40 minutes. In addition, the control needed for this type of airframe is robust to disturbances, making it a good test vehicle.

The control center for the UAV is the Kestrel Autopilot manufactured by Procerus Technologies. It has a 30 MHz Rabbit micro-controller with 512 kilobytes of RAM and 512 kilobytes of flash memory. The autopilot is 2 inches by 1.5 inches and weighs 16.65 ounces, making it ideal for small aircraft. The autopilot has a rate gyroscope and an accelerometer directed along each of its x, y, and z-axes, a differential pressure sensor, and an absolute pressure sensor; each is used in attitude estimation. The autopilot, sensors, and motor operate on battery power.
A wide variety of external sensors has been implemented in conjunction with the autopilot to refine the attitude estimation. Global Positioning System (GPS) units measure position, ground-track, and groundspeed, and are applicable to any situation where GPS lock is available. Optic flow sensors determine height above ground or distances away from objects depending upon their orientation. Ultrasonic sensors also measure height above ground. Magnetometers assist in measuring true heading. A lightweight laser range finder detects the surrounding environment and identifies obstacles in the flight path.

Cameras are an important addition to the experimental platform and allow for image-directed control. Both color and near-infrared cameras are used to perform task-specific missions such as target localization and fire detection. The near-infrared camera is an effective, low-cost alternative to a high-priced infrared camera.

Software

The MAGICC Lab uses two main software tools to help in simulating and testing the developed algorithms. The first one is a flight simulator entitled Aviones and was developed by Morgan Quigley at Brigham Young University. This program emulates the physics of the airframe with six degrees of freedom as well as the communication between the base station and the agent. Aviones can be adapted to fit different styles of planes as well as different environment conditions. The most powerful aspect of Aviones is that the code tested in simulation is identical to the code on the autopilot, allowing for quick transfer from software testing to hardware testing.

Virtual Cockpit is the second piece of software. This program interacts with the plane both in simulation and in hardware. Virtual Cockpit connects to Aviones and allows the user to interface with the simulator in the same way they would in a real-world test scenario. Virtual Cockpit also connects to a communication box that channels information between the aircraft and the computer that is running Virtual Cockpit. This program is a valuable piece in the platform. It allows all the information and control for the plane to be in one place.

Existing Capabilities

With this hardware and software platform, the MAGICC Lab has developed many capabilities in the areas of autonomy, path planning and trajectory generation, cooperative control, and image-directed control. The following is a brief description of those capabilities that are applicable to the forest-monitoring problem.
Autonomy: The autonomy of the UAV is what allows it to be useful in the fire monitoring problem. Developed abilities in this area include autonomous take-off, autonomous landing, attitude estimation, and adaptive control. The take-off consists of the UAV applying full throttle and using airspeed to regulate the pitch angle until the desired altitude is reached. For autonomous landing on target, the operator specifies two points and a breaking altitude. The UAV descends while orbiting around the first waypoint until it reaches the desired altitude, then it breaks out of the orbit and follows a glide slope to the second point. When battery levels are low, the UAV returns to the base station and performs an autonomous landing. Improving attitude estimation involves applying Kalman Filters to refined sensor data. Adaptive control allows the UAV to adjust to any sudden changes it may experience by adapting its gains to maintain stable flight.

Path Planning and Trajectory Generation: There are three main abilities developed in the area of path planning and trajectory generation: waypoint path planning, wind compensation, and collision avoidance. Waypoint path planning allows the operator of the UAV to specify a path for the UAV to track or an area for the UAV to monitor, by simply clicking different points on a map in Virtual Cockpit. The base station transmits the points to the UAV where the autopilot generates a smooth feasible trajectory to follow. Wind compensation allows the UAV to estimate the current wind conditions and adjust its heading and airspeed to compensate for the disturbances caused by the wind. This allows the aircraft to reliably track specified areas in high wind conditions. In collision avoidance, a laser range finder detects when an obstacle is in the path of the UAV. The UAV then plans a new path around the obstacle and back on to the original path.

Cooperative Control: There are many different purposes for having multiple UAVs working together as a team, two of which are: cooperative timing and simultaneous arrival. Cooperative timing is the key to persistently imaging a target. It involves communication amongst the agents to relay their position, heading, and airspeed to one another. From this information, each agent adjusts its velocity and heading in order to arrive at the specified location at the assigned time. A specific example of cooperative timing is simultaneous arrival. In this mission, the agents communicate and come to a consensus of when to arrive at a specified point.

Image Directed Control: Vision-based control has many applications to UAVs. Eye-on-target is one example where the UAV determines its heading by maintaining the target at the center of its viewable window. Another vision-like application is canyon-following using optic flow. Two optic flow sensors, facing outward the sides of the plane, assist in estimating the distance between the UAV and the canyon wall. The UAV then calculates its desired heading based on equalizing the distances on the each side of the plane.

Each of these established abilities can be applied to the fire monitoring problem, and we utilize them in our concept of operations. The developed hardware and software setup give us a stable platform for constructing and testing our solution. We are now ready to introduce our concept of operations for the solution to the fire monitoring problem.

3. CONCEPT OF OPERATIONS

The main objective of the fire monitoring problem is to provide firefighters with useful information about the fire as quickly as possible with minimal human involvement. To satisfy this objective, a UAV must first be able to autonomously track the fire perimeter and collect data. Second, the UAV must provide the data with minimum latency and maximum update rate. This is where a coordinated team of UAVs is employed. Finally, the UAV must format the information in a way that is useful to firefighters without taking up too much storage. With these three stipulations, we break the solution into three parts: fire perimeter tracking for a single UAV agent, cooperative team tracking, and information formatting.

Fire Perimeter Tracking for a Single Agent

In order to track the fire perimeter autonomously, an agent needs to be able to gather real-time information concerning the location of the fire, and from that information devise a flight plan. Images from an onboard camera would provide such information. The agent can locate the edge of the fire by processing the image and then devise control commands that will track the perimeter. The UAV will be able to track the fire perimeter in real-time, if the computational time for the algorithms are sufficiently quick.
(Casbeer et al., 2005) developed an algorithm for tracking the perimeter using onboard video images and tested it in simulation. However, they never discussed the process for locating the fire in the image. Here, we will present an algorithm for determining the location of the fire’s perimeter as well as tracking it.

Testing the algorithm in a real-world environment is difficult since forest fires are not readily available and require authorization to approach. For these reasons, we extend fire tracking to the similar problem of road following, which will allow us to test and refine our algorithms. The analysis from the road-following algorithms can then be applied to the perimeter-tracking problem with minimal adjustment.

In both problems, we are attempting to follow an object that can be modeled as a line - the edge of the fire or a straight road - through feedback from an onboard camera. Once the line has been determined, the method to determine the desired heading is the same in both cases. Therefore, we turn our attention from fire tracking to road following and will refer only to the latter throughout the remainder of the paper.

The road following problem can be separated into two steps with the first being the detection of the road in the image and the second, determining a flight path based upon the location of the road.

**Road Detection:** Detection of a road, using only the current frame, has several issues because of the limited view of the world through the camera. If the road curves too sharply, the UAV will not be able to keep the road in its viewable window. Moreover, road conditions will change over time so the values used for detection will need to be adjusted. The image detection algorithm uses hue-saturation-intensity (HSI) rather than red-green-blue (RGB) pixel analysis. The UAV responds to flight control commands 3 times a second; therefore, the image detection algorithm must run at this speed to send the appropriate commands to the UAV on time.

The flight operator initially enters in specific HSI thresholding values that are used to define pixels that correspond to a road. This is accomplished by using a previously captured image or some other prior knowledge of the area. As the UAV follows the road, the onboard camera captures images. These images are then processed by segmentation using the initial thresholding values. We chose HSI values, as opposed to RGB values, because of their increased robustness to the lighting changes the UAV experiences in flight over time. After segmentation, we erode and fill the image to eliminate artifacts and to distinguish the road boundaries better. Once the road is located in the image, we fit a line to the middle of the road. The UAV will track this line.

After determining the line, we further analyze the image to determine what adjustments must be made to the initial HSI thresholding values that will lead to better detection in the future. This updating is essential for an effective autonomous system to adapt to changes in lighting conditions; otherwise, the user will have to provide the updates himself.

**Perimeter Tracking:** With the road defined by a line, we can determine our command heading by one of two ways. If GPS is available, we can project the line into real-world coordinates and create waypoints, corresponding to the line on the road, for the UAV to follow. The UAV then uses the existing capability of path planning and trajectory generation to create a smooth path to follow. The commanded heading is the heading that will take the UAV onto that path.

If GPS is unavailable, then we are relying completely on what the camera sees to guide the UAV. This results in a unique challenge. The goal is to have the line corresponding to the road bisect the image. However, as the UAV rolls, to follow the path of the road, the camera direction points away from the road. This causes the road to leave the area captured by the image. Therefore, we use a combination of prediction and smart path following to help the UAV maintain the road in the image and follow its path.

By applying these algorithms to the fire-tracking problem, a single UAV will be able to effectively track the fire’s edge, collecting data along the way. The next step is to reduce the latency in delivering the data to the firefighters, and this involves multiple UAVs.
Cooperative Team Tracking

The information delivered by a single UAV is important; however, unless it is updated frequently, the information becomes useless to firefighters along the perimeter of the fire. Conditions change so rapidly that a single UAV would not be able to provide the needed information quickly enough. We discuss a distributed monitoring scheme that involves using a team of UAVs working cooperatively to deliver perimeter information to the base station as often as possible. Our objective is to design a decentralized algorithm that minimizes the latency for the delivery of information while maximizing the update rates of that information. Additionally, we desire the algorithm to account for changes in perimeter length (growing and contracting) as well as the addition/deletion of additional agents.

For analysis purposes and presentation of results, we assume the perimeter of the fire to be a line. Note that the results can be extended to any perimeter homeomorphic to a line without any new analysis. The communication between UAVs around the a large fire may be noisy and/or sporadic. Therefore, we assume a limited communication range for each agent. We also assume that each UAV flies along the perimeter with constant velocity.

The information collected by the UAVs are time-stamped images. Therefore, a latency profile accompanies the data whenever the UAV returns to the base station. We desire to minimize this latency profile. Casbeer et al., (2005), proves that the minimum latency profile occurs when the UAVs take the shortest path from the point of data collection to the base station. This result in having two agents, each monitoring half the perimeter and meeting at the base station and the midpoint of the perimeter. Adding additional agents does not decrease the latency profile any further; however, it does increase the update rate. Casbeer et al., (2005), also shows that the maximum update frequency occurs when the agents equally distribute themselves along the perimeter.

To achieve a perimeter of equally spaced agents, we use the framework of coordination variables. The variables are the key pieces of information quantity such that when all the agents share a consistent view, coordination is achieved. In our algorithm the coordination variables are: the perimeter length, the number of agents on the left side of the perimeter relative to a given agent, and the number of agents on the right side of the perimeter relative to a given agent. When all UAVs come to a consensus on these values, each will be able to determine the perimeter section it should monitor.

The following algorithm ensures that the once the UAVs come to an agreement on the coordination variables, the desired steady state behavior is achieved. The algorithm was first presented in Kingston et al., (2005), and we refer the reader to the source for a complete description of the algorithm. A rendezvous occurs when communication is established between two agents or one agent and the base station.

Algorithm 1: Distributed Spread

```
if rendezvous with neighbor then
    calculate shared border position
    travel with neighbor to shared border
    set direction in order to monitor own segment
else if reached perimeter endpoint then
    reverse direction
else
    continue in current direction
```

Figure 2 shows a mathematical simulation of the algorithm. For this scenario, communication occurs when the points share the same physical location on the line. The arrows in Figure 2(a) indicate the initial direction for each agent, while the plot in Figure 2(b) presents the paths of the agents while following the algorithm. A unit of time in the plot represents the time it takes one agent to traverse the entire perimeter.

Extending this algorithm to UAVs with six degrees of freedom is not trivial and adds additional constraints to the solution. For example, UAVs cannot turn around instantaneously; therefore, we need to adjust the path of the UAV in order to simulate an instantaneous turn. This constrains the communication range of each agent to be at least
double the minimum turning radius of the UAV. Using the existing capabilities of path planning and cooperative timing, we can make the adjustments necessary for the UAVs to fly the algorithm.

With each agent tracking the fire and following the above algorithm, information about the current state of the fire is collected and transmitted to the base station with minimum latency. The more agents available to assist, the quicker the update rate of the data. Now that the data can be effectively delivered to the base station, we need to develop a way of formatting the information.

Information Formatting

The information firefighters need to effectively combat a fire is contained in the captured images from a UAV tracking the fire perimeter. However, the captured video has some significant problems. First, due to the dynamics of a UAV adjusting to flight conditions, the video is often very unstable, jittery, and has noise associated with it. Second, even if the first set of problems were negligible, onboard video footage is not the ideal platform for interpreting information about the detail of an area. It is like looking at the world below through a straw. However, once an area has been view, we can store that information. We can stitch together our current image with the past image, generating a larger image of the area. This is referred to as mosaicking.

A mosaic, besides providing a larger view of the area, can be combined with known geographical information system (GIS) data to provide geo-location, texture, and size information concerning the fire, while reducing redundancy between frames in the video. Moreover, a mosaic can provide accurate information as a less noisy, more concise compilation of the data and is therefore smaller to store and easier to compute than processing large videos.

Producing a mosaic presents at least three major challenges. First, how does each agent process information from its camera? Second, how should the group process information in a collaborative effort. Lastly, how should this information be presented to the end user?

We will assume that communication is limited. Thus, the UAV cannot stream video back to the ground station. Therefore, each UAV needs enough memory and computing power to process its own video stream, build the mosaic of information, and store the results. Each UAV creates its mosaic by using its current position and attitude
estimation information (telemetry) to rotate the image and project it down to create a georeferenced mosaic. Images that cover similar areas are stitched together to give a smooth mosaic. We will utilize the uncertainty information associated with the telemetry to refine the mosaic and obtain a result having higher resolution than the individual frames from the video.

The second challenge is addressed using the cooperative team tracking algorithm described above. Each UAV will build up an individual mosaic. When a rendezvous occurs between two agents, the mosaics are transferred to the UAV that is closest to the base station. Eventually, all the mosaics from the individual planes propagate to the base station, where they are stitched together into one large mosaic representing the current state of the fire. Since a UAV traverses the same path multiple times, each new image will refine the mosaic, removing any noise and uncertainty.

The last challenge is overcome by forming a grid of geolocations using GIS data and then mapping the current mosaic to the grid. This forms a large mosaic where each pixel represents a particular range of GPS coordinates. The resulting map is displayed on a computer screen, allowing GPS coordinates of the fire to be viewed by simply clicking on the map with a mouse. This provides the user with a quick, easy method of determining where objects are in the surveyed landscape.

Another advantage of this format is that we can show a best result at any given time. This means that due to the time of propagation of information each mosaic will have a time stamp associated with it. From this time stamp, the base station creates a best result of the currently surveyed area. As the agents collect more and more information, the mosaic will change and grow. The growth is presented to the user in a topographical map fashion, showing the perimeter of the fire at different time updates. This gives the firefighters a way to interpolate the rate of change of the fire and combat those areas of greater importance.

Fire-perimeter tracking for a single UAV agent, cooperative team tracking, and information formatting are the three main areas that form our purposed solution to the fire monitoring problem. The next section discusses the implementation and experimental results of each area.

4. EXPERIMENTAL RESULTS

In this section, we describe the results of implementing the different parts of our concept of operations. We present the results in the same three areas describe earlier: fire perimeter tracking for a single UAV agent, cooperative team tracking, and information formatting. We conducted simulations in all three areas as well as a real-world experiment for the cooperative team tracking algorithm.

Fire Perimeter Tracking for a Single Agent

To simulate our tracking algorithm, we tested road detection and perimeter tracking individually. Both Aviones and the Virtual Cockpit were used in the simulations. Currently, the UAVs do not have the power to process video onboard. Therefore, the video is transmitted to the base station where it is processed and then control commands are transmitted back. This process is emulated using the Virtual Cockpit.

Road Detection: In order to test our road detection algorithm, we took frames from previously gathered video from a UAV and processed them with our algorithm. Figure 3(a) represents a test image while Figure 3(b) is the same frame after segmentation using the threshold values defined by the yellow box in Figure 3(a). In Figure 3(c), we can clearly distinguish the road, noting that the car and other artifacts are gone. The three red dots in this last image represent the line that will determine the heading angle.

Perimeter Tracking: To test out the tracking algorithm, we created a road on a georeferenced map and loaded the map into Aviones and Virtual Cockpit. The road consisted of straight lines and arcs with a radius of 140 meters, about twice the minimum turning radius of the plane. Throughout the simulation, we collected the true position of the UAV and compared it with the coordinates of the path. The total error, defined as the distance away from the path, is shown in Figure 4. The average error was 4.60 meters away from the path with a standard deviation of __ meters. The maximum error was 12.76 meters.
Using these algorithms, we were able to detect and follow roads accurately. Future work in this area consists of analyzing the computational time of both algorithms, to determine if they are efficient. Additionally, we need to determine the robustness of the system by testing it at its limits, for example, how sharp of a turn can the UAV accurately track? This analysis will allow us to refine our algorithms and give us a metric to compare with other algorithms.

**Cooperative Team Tracking**

In the simulation of our cooperative team tracking algorithm, we programmed a fixed perimeter into Virtual Cockpit. Using the existing ability to follow waypoints, we released two UAV and began the algorithm. After the agents were balanced along the perimeter, we introduced a change in the perimeter length by adding an additional waypoint. Again, after settling, we changed the perimeter length. Once balance was obtained we added an additional agent and changed the perimeter lengths. The results of the simulation are plotted in Figure 5. The y-axis is the position along the perimeter, where 0 through 1 is the original perimeter, while the x-axis is time. Note that the UAVs were able to adjust to the different disturbances and return to a steady-state position.

Figure 3: An example of the road detection algorithm where (a) is the original image take from the onboard camera, (b) is the image after segmentation occurs, and (c) is the final processed image where the road is detected.

Figure 4: A plot containing the road-following tracking error of a simulated flight.
Besides the simulation, we also implemented the algorithm on the existing hardware platform. For this experiment, we launched two UAVs and had them distribute themselves along the perimeter. The wind conditions on the day of the experiment was approximately 30 percent the airspeed of the UAVs. This disturbance made the settling time of the agents a lot longer than in simulation. In addition, the airspeed on one agent was about 1 m/s faster than the other. However, the UAVs were still able to distribute themselves evenly. Figure 6 shows the position plots.

**Information Formatting**

To test our mosaicking algorithm, we simulated a camera taking pictures of a landscape and then build a mosaic from those pictures. This will simulate what happens as the UAVs tracks the perimeter of a fire. Figure 7 represents the landscape or the image from which we are taking simulated pictures.
We now place the simulated camera over the landscape and take pictures as we move the camera. This simulates the movement of a single UAV. With each picture taken, we record the orientation of the UAV at the time when the picture was taken. We use this attitude information to build a complete mosaic of the landscape. Figure 8 shows the mosaic created from the landscape in Figure 7. The simulated camera is moving away from the image up and to the left while slightly rotating the camera clockwise with each new image.

Looking at the mosaic, we can determine the movement of the camera by the way the individual images change. The rotation of the camera is noticed by the rotating images. The increasing enlargement of each frame corresponds to the movement of the camera away from the landscape; the camera is able to capture more in the image at higher altitudes. Note that towards edges of the mosaic, the image is blurry compared to other parts in the mosaic. This blur is resolved to some degree as we obtain more images of the same areas in the landscape. The center areas of the mosaic are more defined because multiple frames cover that area. Thus, we can take information from many individual images and obtain a final mosaic with higher resolution than any of the original images. However, there are still small artifacts that enter into the mosaic. Removing these artifacts has been addressed in (Ready,2006)
5. CONCLUSION

In this paper, we have introduce the research going at Brigham Young University in collaboration with NASA Ames to utilize LASE UAVs to monitor forest fires. We have described the experimental platform developed by the MAGICC Lab and overviewed capabilities of the platform that are pertinent to the fire-monitoring problem. We then presented a concept of operations using multiple LASE UAVs to monitor the perimeter of a forest fire. This concept of operations was divided into three areas: fire perimeter tracking for a single UAV, cooperative team tracking, and information formatting. We showed simulation results from each of these areas and hardware results from the cooperative team tracking algorithm.

From our results, we conclude that our proposed concept of operations is effective and feasible. However, there is more work to perform. The tracking and mosaicking algorithms need to be tested in real-world situations, as well as the integration the three different areas. With these tests and the refinement that comes from testing, we will effectively be able to monitor forest fires using multiple LASE UAVs.

ACKNOWLEDGMENTS

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.

REFERENCES


