12-2017

An Unmanned Aerial System for Prescribed Fires

Evan M. Beachly
University of Nebraska-Lincoln, ebeachly@gmail.com

AN UNMANNED AERIAL SYSTEM FOR PRESCRIBED FIRES

by

Evan Beachly

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professors Carrick Detweiler and Sebastian Elbaum

Lincoln, Nebraska

December, 2017

AN UNMANNED AERIAL SYSTEM FOR PRESCRIBED FIRES

Evan Beachly, M.S.

University of Nebraska, 2017

Advisors: Carrick Detweiler and Sebastian Elbaum

Prescribed fires can lessen wildfire severity and control invasive species, but some terrains may be difficult, dangerous, or costly to burn with existing tools. This thesis presents the design of an unmanned aerial system that can ignite prescribed fires from the air, with less cost and risk than with aerial ignition from a manned aircraft. The prototype was evaluated in-lab and successfully used to ignite interior areas of two prescribed fires. Additionally, we introduce an approach that integrates a lightweight fire simulation to autonomously plan safe flight trajectories and suggest effective fire lines. Both components are unique in that they are amenable to input from the systems sensors and the fire crew. A preliminary study confirms that such inputs improve the accuracy of the fire simulation to better counter the unpredictability of the target environment.

ACKNOWLEDGMENTS

I would like to greatly thank my advisers, Dr. Carrick Detweiler and Dr. Sebastian Elbaum, for all of their help, support, and guidance over the past two years as I developed as a researcher. Having access to your vast experience was invaluable for all of the questions I had during my project and about research in general.

The creation of the UAS-Rx was a collaborative effort, and this research would not have been possible without the work and assistance of the other team members. James Higgins and Christian Laney created an early prototype of the UAS-Rx before I joined, and designed the dropper and circuit board used on the current UAS-Rx. James and Christian taught me a lot as I started working on this project, and were always willing to answer my mechanical and hardware related questions. Becca Horzewski designed a temperature sensor for the UAS-Rx, Ashraful Islam designed the hopper, and Denis Komissarov and Carl Hildebrant assisted with lab tests. I am grateful for their contributions, help, and the opportunity to work with them. Additionally, I would like to thank all of the members of the Nimbus lab, especially Ajay Shankar, for the camaraderie and all of our discussions.

I would also like to thank Dr. Dirac Twidwell for discussing the possibilities of a field-usable fire simulation with me. Your experience in fire modeling is invaluable. I would also like to thank Dr. Craig Allen for additional insights.

The user study would not have been possible without the guidance of Dr. Brittany Duncan, whose familiarity with human subjects studies and the process associated with them made my study a success. I would also like to thank her for her assistance with field tests.

The field tests would not have been possible without the help of the National Parks Service, the Department of the Interior, and the Federal Aviation Administration. I especially appreciate the efforts of the Loess Canyons Rangelands Alliance, Homestead National Monument, the Tri-County Prescribed Burn Association of Nebraska in accommodating us into their prescribed burns.

Finally, I would like to thank the volunteers who participated in my study, and Alisha Bevins for interviewing the participants.

GRANT INFORMATION

This work was supported in part by NSF-1638099, NSF-1217400, USDA-2013-67021-20947, and NE Game and Parks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Prescribed fires can reduce wildfire severity [6, 17, 11], control invasive species [14, 2, 34], and improve rangelands for livestock and grazing [21]. When prescribed fires are conducted, the perimeter of the area is burned first as a buffer, and then the interior of the area is ignited to speed up the burn. Burn crews typically use drip torches either carried by hand or mounted on ATVs to ignite the interior, as shown in Figure 1.1.

Figure 1.1: Interior ignition being conducted from an ATV. Photo taken at Loess Canyons prescribed fire in Spring 2016.

Figure 1.2: Helitorch

On difficult terrain, such as islands, plateaus, or ravines, it may be impossible, dangerous, or exhausting to ignite certain areas with a drip torch carried on foot or by ATV. There are several ignition tools that address this issue [36]. Flare guns and other incendiary launchers can ignite locations like these from a distance, but have a limited range (up to 90 meters). Aerial ignition tools such as the helitorch (Figure 1.2) and delayed aerial ignition sphere dispensers [20] can be mounted on airplanes or helicopters to ignite large areas, but the costs of running an aircraft are not economical for many landowners (it may cost $10,000 to contract a helitorch for a prescribed burn [28]). Prescribed burn crews need new tools for interior ignition that reduce risk, yet are low cost and easy to operate, to make them available to the majority of prescribed fire users.

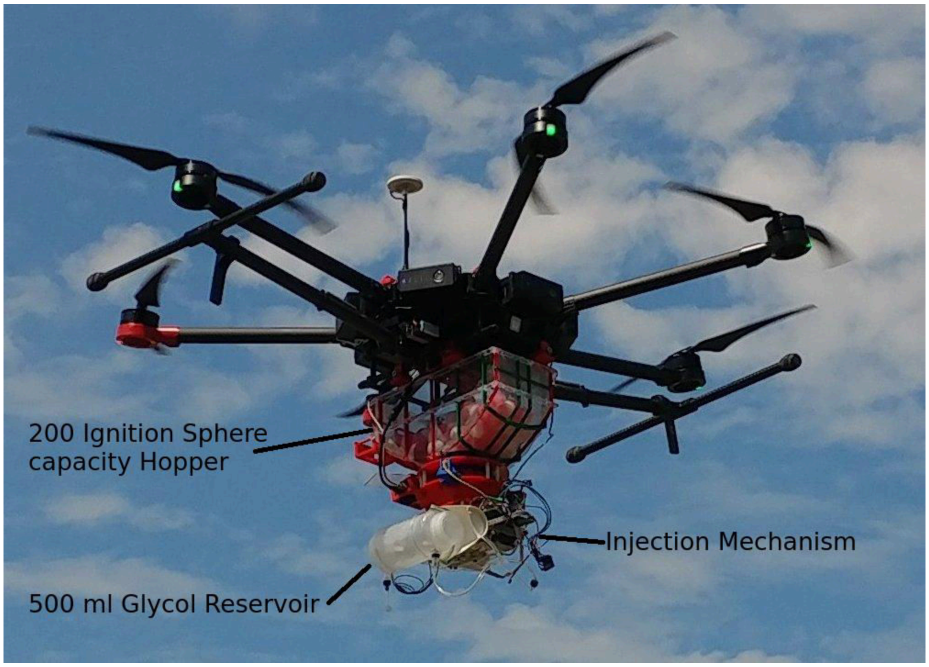


Figure 1.3: 2017 version of the UAS-Rx (3.0.1) in flight.

This work describes an unmanned aerial system for fire prescription (UAS-Rx), shown in Figure 1.3. The UAS-Rx transforms UASs from those that only remotely measure and monitor fires to a system that can actively manipulate the shape and trajectory of the fire to achieve the desired environmental management goals. The UAS-Rx ignites fires using the same commercially available delayed aerial ignition spheres already widely-used for ignition from manned aircraft. Figure 1.3 shows the hopper for carrying the ignition spheres, the reservoir containing the activating chemical, and the mechanism for injecting and dropping each ignition sphere. Figure 1.4 shows the results of an earlier (2015) prototype UAS-Rx igniting a prescription by dropping delayed aerial ignition spheres onto the invasive Cedar trees in the targeted area.

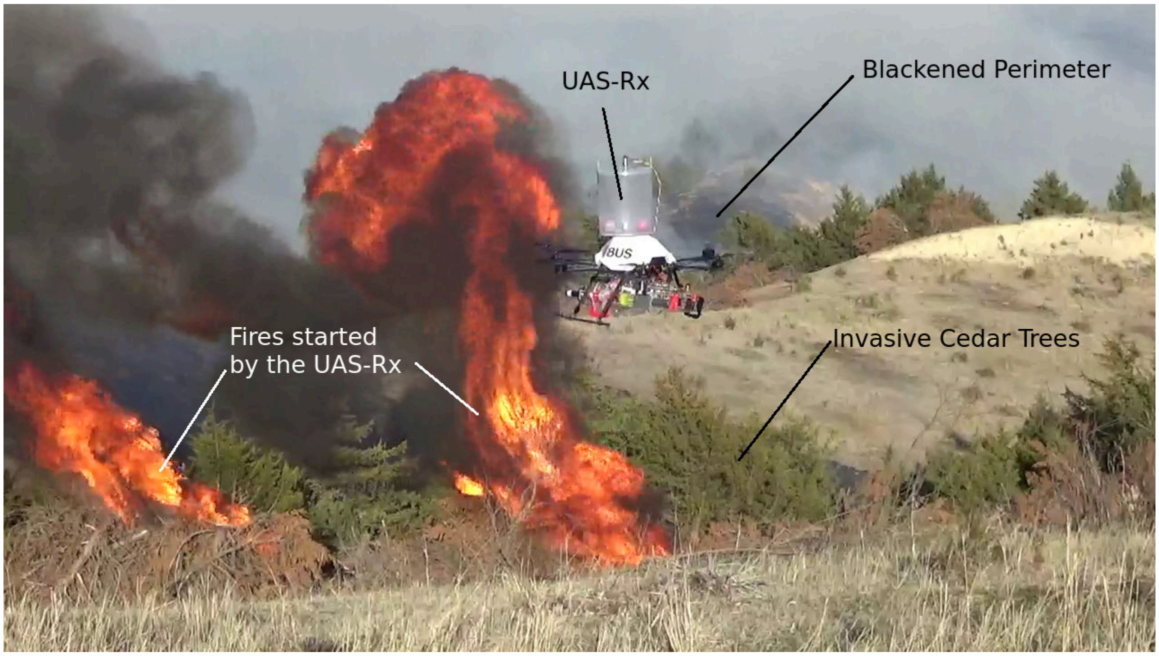


Figure 1.4: The 2015 prototype version of UAS-Rx (1.0.3) returning after starting a prescribed fire with the Loess Canyon Rangeland Alliance [3].

Our vision is that the UAS-Rx would be used at prescribed burns that cannot afford aerial ignition from a manned aircraft. The lightweight UAS could be carried by a firefighter to the burn site, and then be deployed to ignite terrain that is difficult to enter and ignite normally.

Another advantage of using a UAS for prescribed fires is that it offers an aerial platform for cameras and sensors, allowing the firefighters to maintain situation awareness. This work takes advantage of the UAS-Rx's sensing capabilities by developing an integrated fire simulation that can adjust to real-time observations of the fire. This work also proposes an algorithm to plan burn lines for the UAS-Rx to ignite. To the best of our knowledge, this is the first autonomous robotic system that has been designed for and used to start prescribed fires [10, 35].

The contributions of this work are:

- The integration of the UAS-Rx as a whole system and the design of the UAS-

Rx's ignition software,

- Evaluation of the UAS-Rx through lab tests and field tests at two prescribed burns,
- A light-weight fire simulation that can be corrected with real-time observations of the fire, and
- An algorithm that uses the fire simulation to automatically plan ignitions for the UAS-Rx.

Though these contributions are presented in the context of the UAS-Rx, we hope that the ideas in this thesis can inspire other researchers. Particularly, a correctable fire simulation could be useful in any scenario where an active prescribed or wild fire needs to be tracked. Additionally, providing an interface to the user so that they can see and correct the robot's model of the environment is a technique that has been and can be applied to any semi-autonomous robotic system, and the interface described in this work for correcting the fire simulation is a touchscreen implementation of that technique.

The chapters in this thesis are adapted from two conference papers published by the author and colleagues. Chapter 2 is adapted from *A micro-UAS to Start Prescribed Fires* [10] and covers the first two contributions and the design of the Dropper. Chapter 3 is adapted from *Fire-Aware Automated Aerial Ignitions* [9] (under review at the time this thesis was written) and covers the last two contributions.

The development of the UAS-Rx was an iterative process, and the design has gone through many revisions. Figure 1.5 shows the previous versions of the UAS-Rx, with Figure 1.3 showing the current version (3.0.1). Chapter 2 describes the design of the Dropper on Version 2.0.1, and the use of Versions 1.0.3 and 2.0.1 of the UAS-Rx at two

(a) V1.0.1

(b) V1.0.2

(c) V1.0.3

(d) V2.0.1

Figure 1.5: Previous versions of the UAS-Rx. (a) Version 1.0.1 with under-mounted hopper (b) Version 1.0.2 with over-mounted gravity fed hopper (c) Version 1.0.3 with agitated hopper (d) Version 2.0.1 with over-mounted chute and new dropper

prescribed fires. Chapter 3 describes features of the current version of the UAS-Rx. The design of the dropper has not changed since Version 2.0.1. For more information on the previous versions, see Chapter 5 of *Design, Testing, and Evaluation of Robotic Mechanisms and Systems for Environmental Monitoring and Interaction*[19].

1.1 Related Work

This is the first UAS to be used to ignite prescribed fires. However, UASs have previously been used in the fire domain for remote fire measurement and monitoring [4, 24, 25], including simulations on how to track fire and optimize flight paths in these conditions [12, 32].

Dropping the ignition spheres is similar to dropping wireless sensor nodes, which has been performed using autonomous helicopters [13, 5], and fixed-wing UAS's [26]. However, our dropping mechanism must also safely and reliably puncture and inject each sphere before dropping.

We also note that this is a first step towards fire simulations that can take human observations as input to support fire ignitions, but we are not alone in pushing for the incorporation of real-time observations into fire simulations. Recently Gollner et al. stated the need for such operational wildfire spread models that can take real observations [18]. Existing fire simulations like FARSITE [15] and FSPro [16] are intended to be run off-line, with intensive simulations of various fire scenarios to help the burn crew prepare before the fire.

From a robotics perspective, using human input is a common technique to improve the robot's model of the environment (e.g., [22], [33]). In order to do that, the environment model built by the robot needs to be conveyed to the human so the human can understand the decisions made by the robot and provide adjustments [30].

The UAS-Rx combines these ideas in order to effectively perform semi-autonomous aerial ignition in the fire domain, however, these ideas can be generalized into a framework for any semi-autonomous field robot, shown in Figure 1.6. The UAS-Rx's fire simulation fills the role of the model of the environment, and interacts

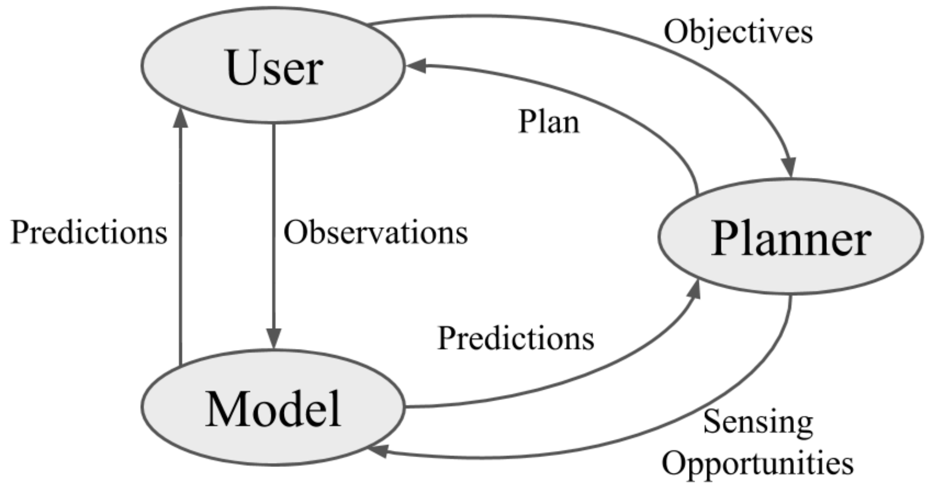


Figure 1.6: Inputs and outputs of a system in which the user interacts directly with the planner and model.

with the ignition planner and user. In Figure 1.6, the model is the robot's internal representation of the environment or system, and the planning algorithm uses this and objectives from the user to create a sequence of actions for the robot to take. The user then has the opportunity to accept, reject, or modify this plan before allowing the robot to execute it. The planner may balance completing objectives with collecting information about the environment, and thus creates sensing opportunities for the robot to collect information about uncertain parts of the environment, which can be used to improve the model. We recognize that the UAS-Rx can serve as a platform for fire monitoring, and since collecting fire information is crucial for updating the fire simulation, a future planner for the UAS-Rx may need to balance ignition with observing the fire, using algorithms related to the techniques presented by Casbeer [12] and Skeele [32].

The key part of the system in Figure 1.6 are user interfaces and methods for viewing and correcting the robot's model of the environment. The model provides predictions about the environment to the planner algorithm, but in order to convey the environment model to the user, a way to visualize or communicate those predictions in a human-consumable form is needed. This visualization can also provide an interface for the user to input their own knowledge and observations of the environment, and make corrections to errors in the model. Building, conveying, and updating a robot's environment model can be non-trivial endeavors as the model may not fully map to reality due to limitations of the robot's sensors, resources, and algorithms, and the human's participation, although valuable, may be challenging to obtain and incorporate cost-effectively. In this work we attack each of these challenges in the context of the fire environment with input from fire personnel.

Chapter 2

A micro-UAS to Assist Prescribed Fires

2.1 Requirements

For the UAS-Rx to be successful, the technical capabilities need to be contextualized in the fire-ignition domain. This context is defined by target areas covering hundreds to thousands of acres, teams of firefighters performing different roles and operating a variety of vehicles, all working under a burn plan and a set of regulations and common practices, and operating in specific ignition situations that make firefighters especially vulnerable. For example, at the Homestead National Monument prescribed fire described later in this chapter, 50 firefighters coordinated to ignite a 50 acre area. This burn had many regulations for the personnel to follow, as this was being conducted on a national monument. In contrast, the Loess Canyons prescribed fire involved around 60 firefighters, who ignited 2000 acres in a single day using faster ignition practices. This context and our early studies with fire ecologists, land managers, and firefighters defined an initial set of parameters that have influenced the design of the UAS-Rx:

- Must be small and light enough to be carried by a single firefighter on foot, or in their vehicle.
- Must be easily deployable and operable in a hostile environment (e.g. wind gusts, smoke, hot temperatures) and terrain (e.g. canyons, trees, gullies).
- Must not increase the potential for uncontrolled fires.
- Must align with large body of practices and regulations on how such fires must be conducted.

These requirements lead to the design of a UAS-Rx prototype built on a micro-UAS platform, that can navigate and drop a fire payload with enough precision to remain within specified regions, and that replicates an accepted form of fire-ignition delivery in a miniaturized and automated fashion. The next section covers key technical elements underlying these themes.

2.2 Technical Approach

This section describes the design of version 2.0.1 of the UAS-Rx, including an overview, the design of the dropping mechanism, the embedded microcontroller, and the user interface. Our design of the UAS-Rx has gone through several revisions that explored different sensing and payload tradeoffs (Figure 1.5). We present the prototype used in the Homestead National Monument field test in this section. The design of the dropping mechanism and embedded microcontroller have not significantly changed between this version and the current version of the UAS-Rx (3.0.1). However, the hexacopter, ignition sphere container, and interface have changed since this version and will be described in Chapter 3.

2.2.1 Design Overview

The prototype version 2.0.1 of the UAS-Rx is shown in Figure 2.1. It consists of three main parts: a hexacopter (commercially available, Ascending Technologies Firefly UAS), a chute that contains ignition spheres, and a "Dropper" attached underneath the hexacopter. The UAS-Rx is 39 cm tall, 65 cm wide, and has a mass of 1.9 kg at takeoff.

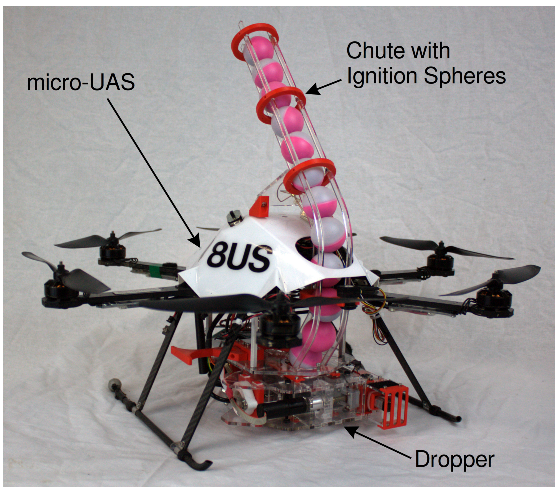


Figure 2.1: Unmanned Aerial System for Fire Prescription (UAS-Rx).

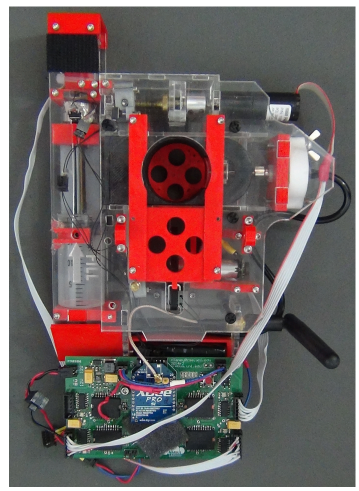


Figure 2.2: Dropper Top View.

The chute on the UAS-Rx carries 12 delayed aerial ignition spheres, which are used to start the fire. Ignition spheres are a commercially available product designed to be used for aerial ignition from helicopters. The brand used in this work is the Premo Fireball [23]. Each ignition sphere is a 32 mm diameter hollow plastic sphere containing 3 grams of Potassium Permanganate. When an ignition sphere is injected with 1 ml of common automotive antifreeze, the Ethylene Glycol in the antifreeze will start an exothermic chemical reaction with the Potassium Permanganate. The ignition sphere will burst into flame 20 to 60 seconds after injection, depending on the ambient temperature and other factors. Figure 2.3 shows the flame generated during the initial combustion of an ignition sphere. The use of ignition spheres that were already widely used by the fire community has significantly aided the acceptance of the UAS-Rx.

Figure 2.3: Ignition Sphere Igniting

The device for injecting and dropping the ignition spheres, the Dropper, (shown in Figure 2.2) is attached underneath the hexacopter by a manual quick-release mechanism. The ignition spheres are gravity-fed to the Dropper by a chute that wraps around the front of the hexacopter. These mechanisms are described in greater detail by Higgins [19]. The total mass of the ignition spheres and dropper is 782 grams. On the Firefly, this payload constrains the maximum flight time to 10-12 minutes. The system, however, is designed to be self-contained with its own battery, processing, and communication, so that it can be carried by larger multi-rotor or fixed wing UASs with correspondingly longer flight times, such as the DJI Matrice 600 used in Version 3.0.1 of the UAS-Rx.

Figure 2.4 shows the elements used to transmit information from and send control signals to Version 2.0.1 of the UAS-Rx. The operator controls the Hexacopter and

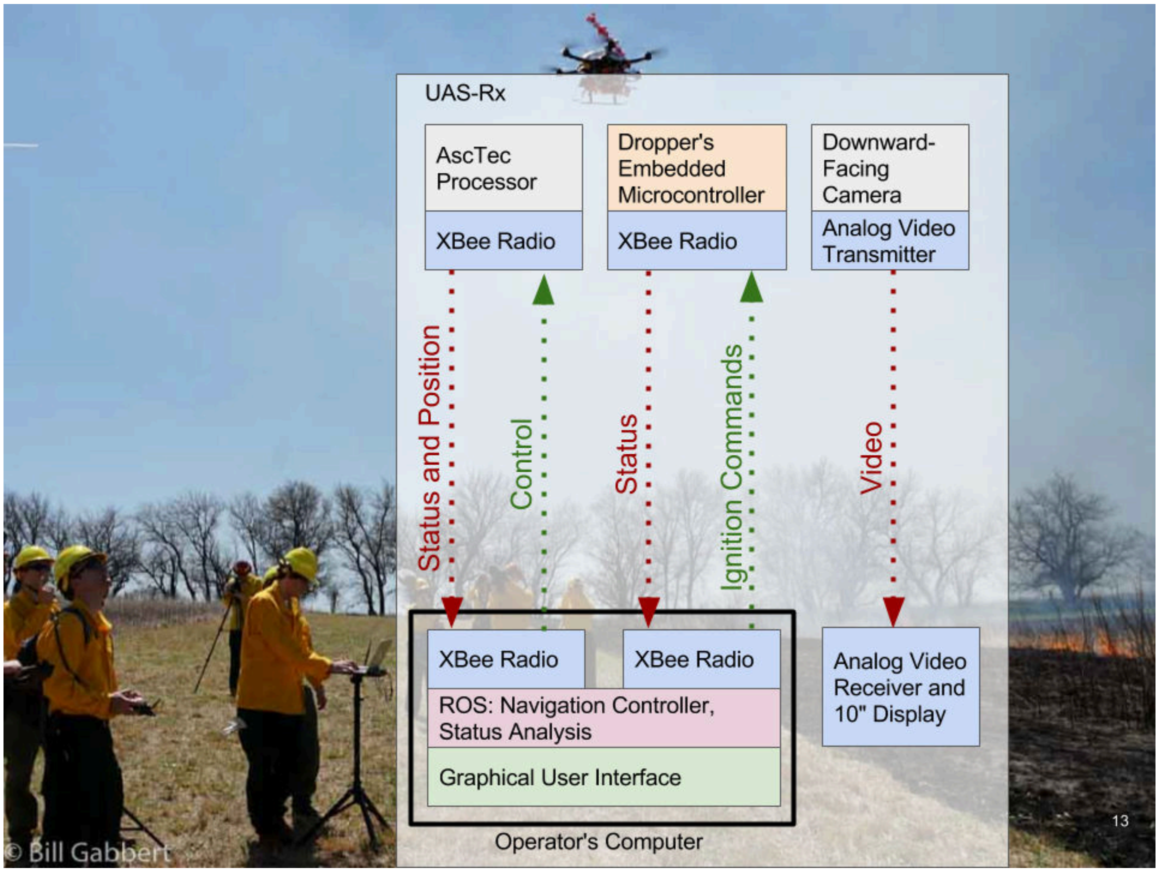


Figure 2.4: Communication and control elements of Version 2.0.1 of the UAS-Rx

Dropper using a laptop computer. The control software for the UAS-Rx is implemented using Robot Operating System (ROS) [27]. Status and GPS Position information of the UAS-Rx are transmitted over an XBee Radio link to the computer, which uses a PID controller to autonomously fly the UAS-Rx to waypoints set by the operator. Another XBee link communicates with the embedded microcontroller on the Dropper, and can be used to command it to inject and drop an ignition sphere. Video is streamed from a downward-facing camera mounted on the UAS-Rx to an analog video receiver next to the operator.

2.2.2 Dropper Mechanical Design

The dropper is responsible for loading, piercing, injecting, and releasing the ignition spheres, and accomplishes this using three motors. The structural components of the Dropper were rapidly prototyped from 3-D printed thermoplastics and laser cut acrylic. Figure 2.5 shows the loading and release system, a pair of sliding hatches controlled by a single motor.

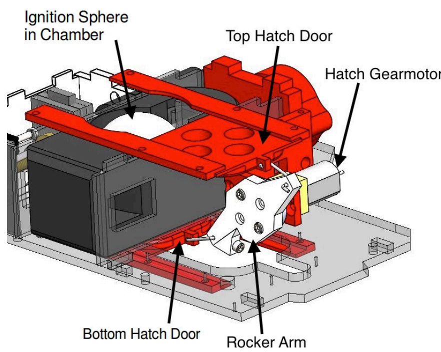


Figure 2.5: Loading and Releasing System.

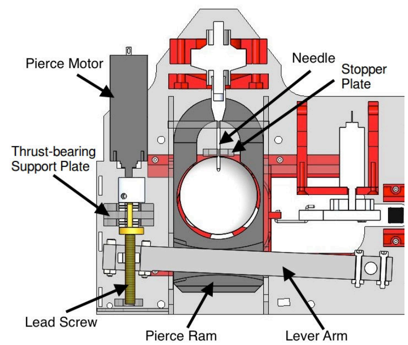


Figure 2.6: Piercing System.

Once an ignition sphere has fallen into the chamber, the pierce motor (see Figure 2.6) pulls on the lever arm and drives the ignition sphere onto a 16 Gauge stainless-steel needle. Puncturing the ignition sphere with the needle normally requires approximately 50 Newtons of force. However, the shell of the ignition sphere has ribs and a seam of thicker plastic that can require up to 100 N of force to pierce. The combination of the piercing motor, lead screw, and lever arm can produce an estimated piercing force of 130 N, assuming 80% loss caused by the lead screw and friction between moving components.

As the pierce ram pushes on the ignition sphere, the curved surface on the interior of the chamber centers the ignition sphere onto the needle. This ensures that the needle does not get deflected and bent by an oblique strike on the curvature of the ignition sphere.

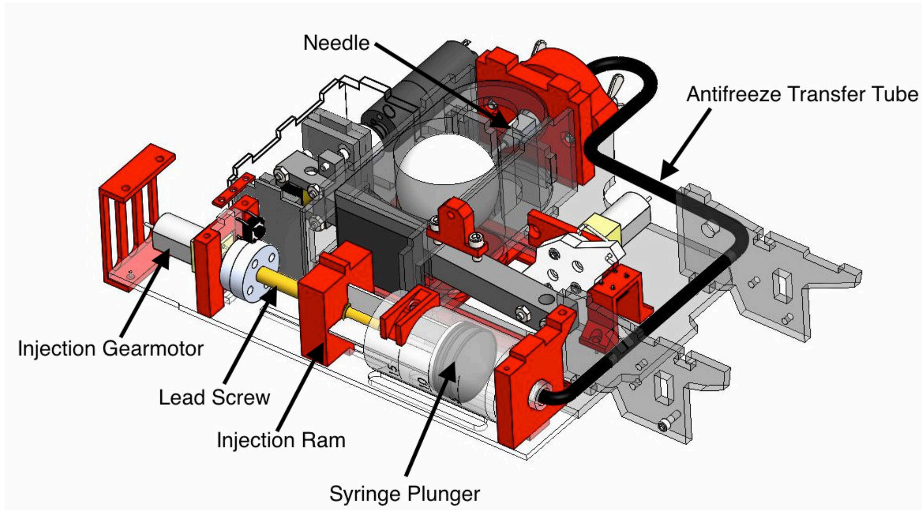


Figure 2.7: Injection System

Figure 2.7 shows the system that injects the ignition sphere with antifreeze after it has been pierced. Antifreeze is carried in the syringe, which gets compressed by the injection motor. After compression, the antifreeze travels through the antifreeze transfer tube and out the needle.

When the ignition sphere is pulled off the needle, there is 2mm of clearance between the needle tip and the sphere. This is more than enough to ensure that it will not remain stuck on the needle tip when it needs to be dropped, and account for any variability in the shape of the ignition sphere.

2.2.3 Dropper Embedded System Design

The embedded system was designed to reduce the risk of an ignition within the dropper. This is accomplished by closely monitoring the motors to detect any failures, taking precautions before injecting the ignition sphere, and making the sequence of operations required to inject and drop an ignition sphere an atomic operation from the user's perspective.

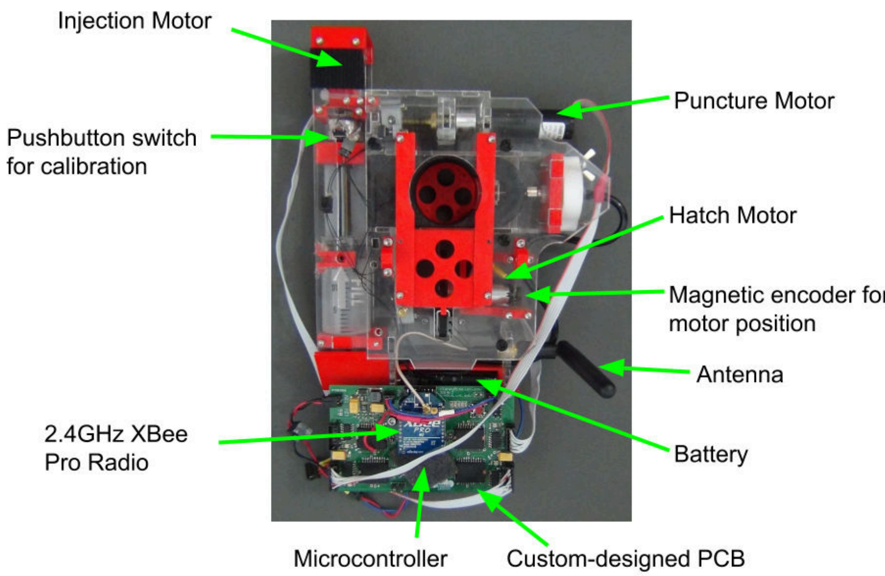


Figure 2.8: Dropper Electrical Components

Figure 2.8 identifies the major electrical components of the Dropper. The Dropper is controlled by an ATMega2560 microcontroller on a custom-designed printed circuit board. Each motor is controlled by a motor driver with built-in current sensing and over-current protection. Quadrature counter chips track the position of magnetic encoders on each motor. We placed pushbutton switches at the limit of each actuator's range of motion to calibrate the positions on startup, and to detect when the actuator reaches the limits of its motion. The processor in this prototype communicates to the ground station using a 2.4 GHz XBee radio module that has a range of 1 km.

While running a motor, the processor monitors the current draw and position in a 500Hz control loop, and follows the algorithm shown in Figure 2.9. The processor uses the counter to track the actuator's position, and stop it at the correct place. If the counter stops incrementing or decrementing while the motor is being powered, or if the motor is drawing a large amount of current, the motor is assumed to have stalled, and is stopped to prevent damage. As a fail-safe, each operation has a configurable timeout that limits how long the motor will run before the processor considers its

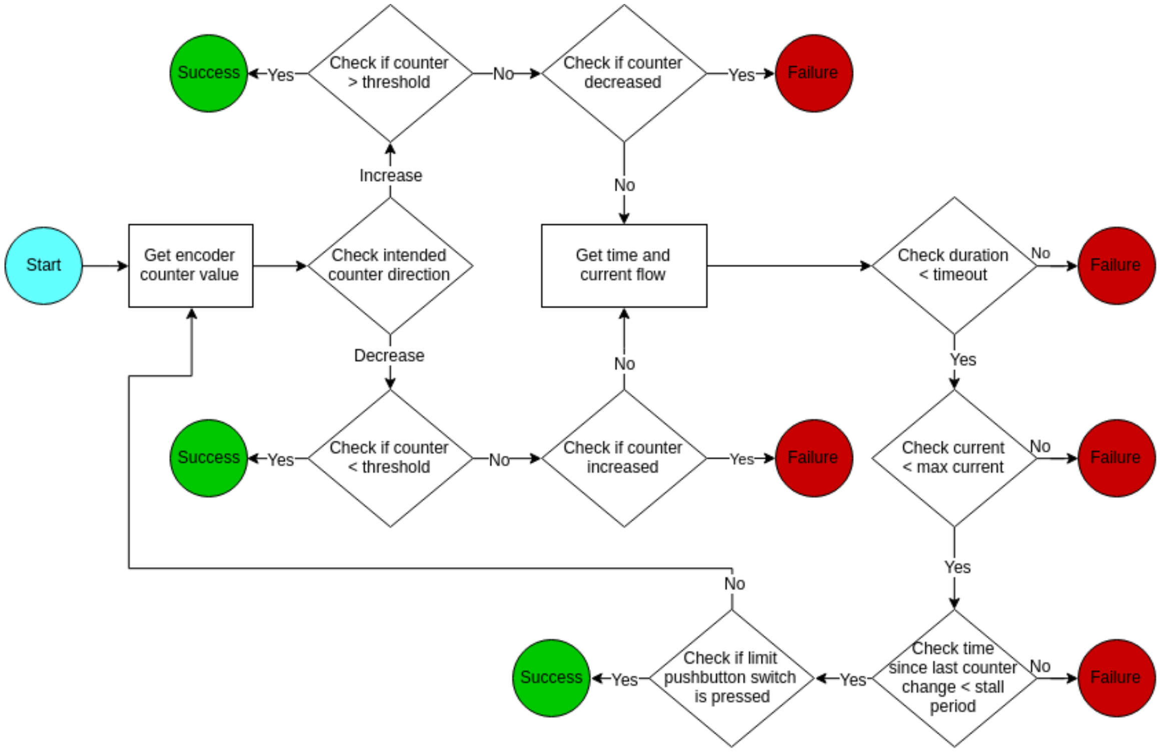


Figure 2.9: Procedure to safely actuate a motor by continuously checking termination conditions.

next action. Status messages are transmitted from the dropper automatically at a rate of 5 Hz, and inform the operator about what the dropper is trying to do, its state, and any failures that have occurred.

Figure 2.10 shows the details of the procedure that the embedded processor follows to inject and drop an ignition sphere. The rectangular operations that involve motor actuation use the safe motor actuation in Figure 2.9. A success or failure in Figure 2.9 determines the next step of the operation in Figure 2.10.

The worst case scenario is for an ignition sphere to be injected, but unable to be released. The procedure in Figure 2.10 helps reduce the probability that a mechanical failure will lead to this situation by only injecting if the bottom hatch was successfully opened, and if the piercing ram is functional. In the event that the piercing ram is unable to drive back after injection and drop the ignition sphere, the operator is

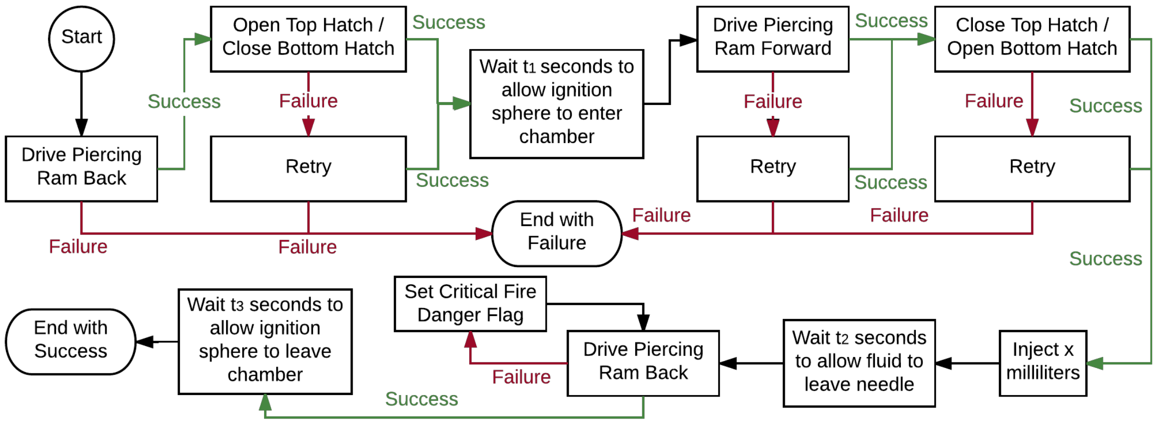


Figure 2.10: Procedure to inject and drop an ignition sphere. Wait times and the injection amount can be customized over the radio link, but default to 1 s and 1 ml.

alerted by the critical fire danger flag in the periodic status messages transmitted by the Dropper's processor. The dropper will automatically continue to attempt to remove the ignition sphere from the needle. Only in the event that none of these attempts succeed will the ignition sphere ignite inside the dropper.

The operator has limited control over the actuators in the dropper. This is to prevent unintentionally injecting an ignition sphere without dropping it. A single command starts the entire inject and drop process shown in Figure 2.10. Ending with a Failure or Success in Figure 2.10 will return the Dropper to an idle state, where it waits for the next command. Since the operator is receiving information about what failures have occurred and whether the dropper is successfully completing this procedure or not, the operator may decide to stop sending drop commands and abort the mission if the dropper is repeatedly failing this procedure. This operator action may be necessary in the event that the Dropper jams or a part breaks mid flight.

2.2.4 User Interface

Prescribed burns are highly dynamic, and changes in wind or the progress of the fire may require adjusting the burn plan. The operator needs a clear understanding of the UAS-Rx's situation in order to react to these changes. To facilitate this, our prototype displayed information on the operator's laptop computer (Figure 2.11), and relayed video to a separate radio video receiver. The computer renders a top-down view of the area centered on the UAS-Rx's takeoff point. The rendered view has icons for the UAS-Rx (red dot), the path it has recently traveled (yellow line), and the current waypoint (blue dot). In addition to this rendered view, the UAS-RX has a downward-facing video camera and analog video transmitter to allow the operator to see where the ignition spheres are landing.

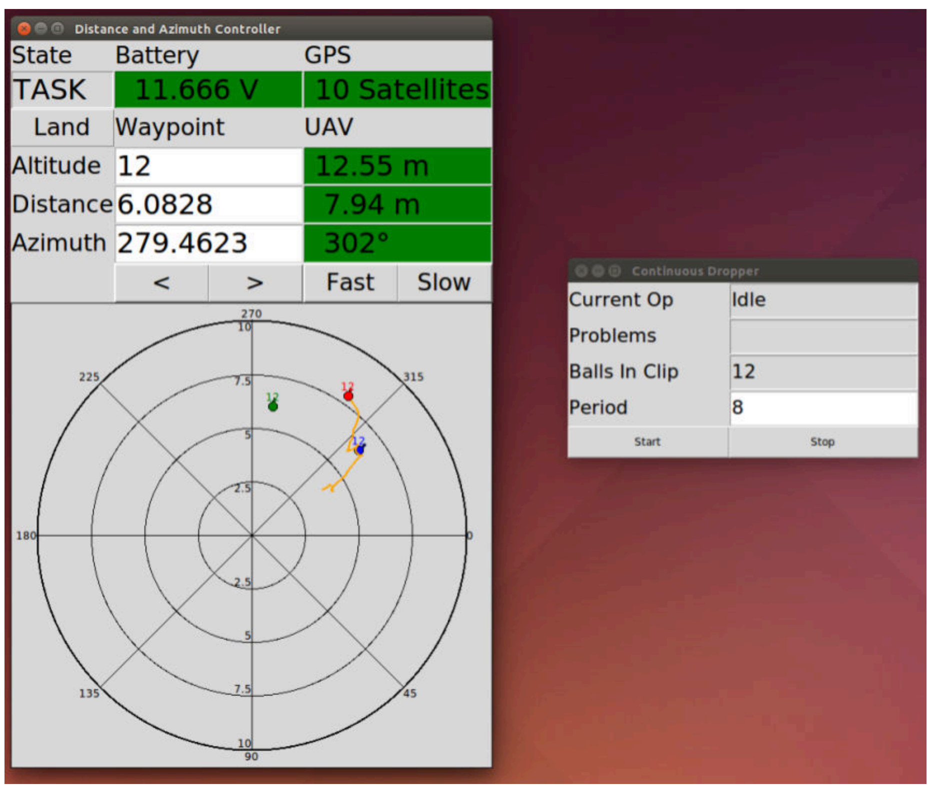


Figure 2.11: Graphical User Interface for Version 2.0.1 of the UAS-Rx, including a window for flying the UAS, and a window for periodically dropping ignition spheres.

The operator can move a cursor (green dot) around using the arrow keys to set a new goal location for the UAS-Rx to fly to. The speed of the UAS-Rx can be set to one of two pre-programmed speeds (4 m/s or 0.25 m/s). The dropper is controlled by the smaller window, which allows the operator to periodically drop ignition spheres. Adjusting the dropping period will result in different spacings between ignition spheres for a given UAS speed.

2.3 Experiments and Results

We tested the UAS-Rx both in-lab and at two actual prescribed burns. In-lab tests were conducted mainly to quantify the reliability of the dropper in a controlled setting. The purpose of the prescribed fire tests was to gain information about the kind of missions the UAS-Rx is expected to be able to complete, the fire environment, and to identify ways to further improve it for use at prescribed burns.

2.3.1 In-lab tests

The UAS-Rx was extensively tested in our lab and also in an indoor arena where we could test ignitions in a controlled environment. Encoder and motor failures were simulated in order to validate that the software can detect the failures and respond correctly. Communication tests showed that 96% of status messages are received when the UAS-Rx was 200 meters away.

A key portion of our tests evaluated the Dropper, which needs to be able to reliably and safely puncture, inject, and drop each ignition sphere. To do this, we built two Droppers and processed 60 ignition spheres through each, injecting water instead of glycol for safety. Each ignition sphere was weighed before and after injection to determine the amount of fluid injected. Additionally, the Dropper transmits information about each injection, such as the time it took to pierce the plastic, and the time it took to compress the syringe by 1 mL. During these tests, the needle never became dull, bent, or plugged with plastic, and no sphere became jammed in the system or had difficulty leaving the Dropper after injection.

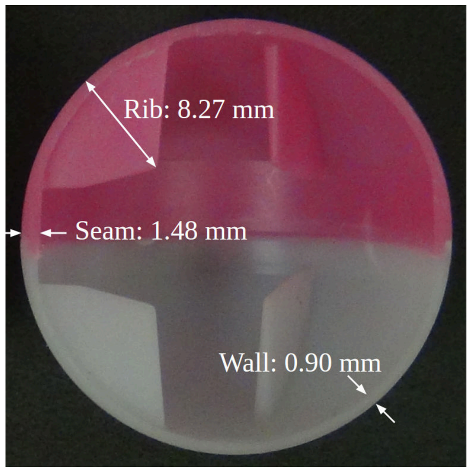


Figure 2.12: Cross section of an ignition sphere.

We noted that the performance of the Dropper heavily depended on where the needle pierced the ignition sphere, as the shell of the ignition sphere is not uniformly thick. Figure 2.12 shows the anatomy of an ignition sphere, consisting of two hemispheres of plastic welded together at the seam. Each half has 3 ribs for support. Piercing through the seam and ribs requires more force than the rest of the wall of the sphere, and the improperly punctured hole can restrict fluid flow into the sphere.

Figure 2.13 shows a histogram of the results of all 120 tests (60 with each dropper). The colored portion of the bars indicates what fraction of the occurrence pierced each location on the ignition sphere. Puncturing the wall always resulted in more than 0.5 mL of fluid being injected, while puncturing through a rib always resulted in 0.5 mL or less fluid being injected.

Figure 2.14 shows a scatter plot of each sphere's time to puncture and time to inject, and colors the points by the puncture location. The seam and rib provide greater

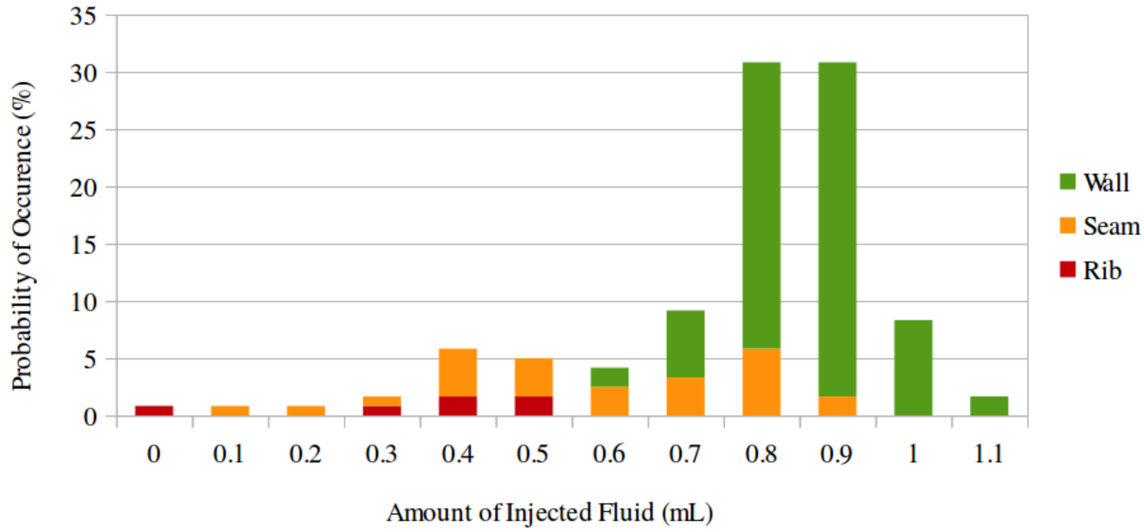


Figure 2.13: Histogram of puncture location and amount of fluid injected.

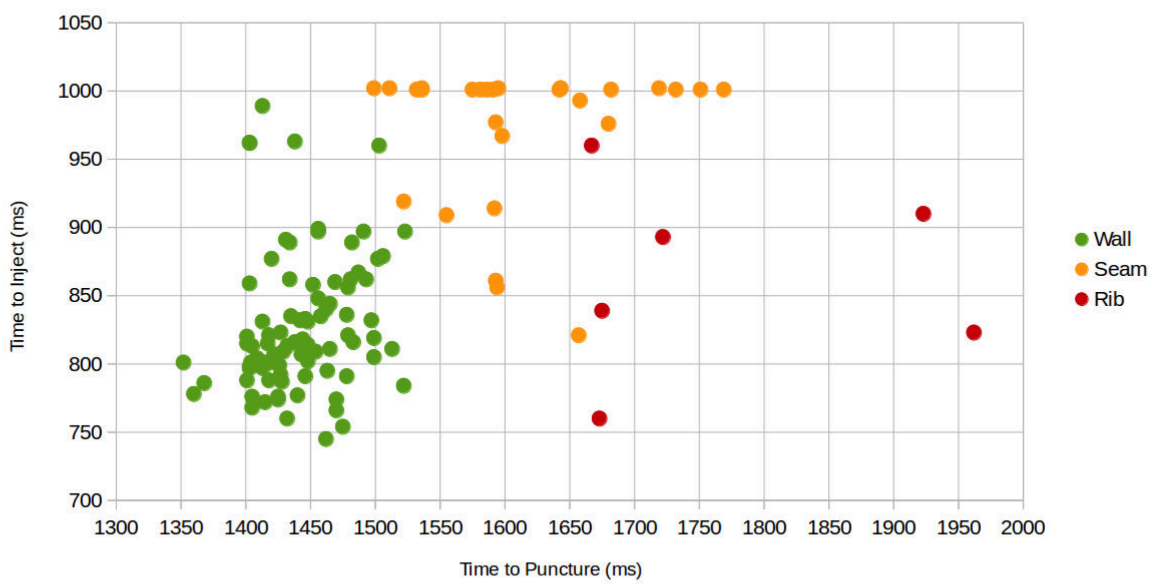


Figure 2.14: Puncture location for 120 injection tests

resistance to the needle during puncturing, resulting in a higher time to puncture. Additionally, puncturing at the seam and rib results in poor puncture holes that restrict fluid flow into the sphere, resulting in higher time to inject. These experiments were conducted using the same programmed safety measures that would be used in practice to protect the motors on the dropper from stalls. One of these is a 1 second time limit on the running of the injection motor, which is why none of the occurrences have an injection duration much longer than 1000 ms.

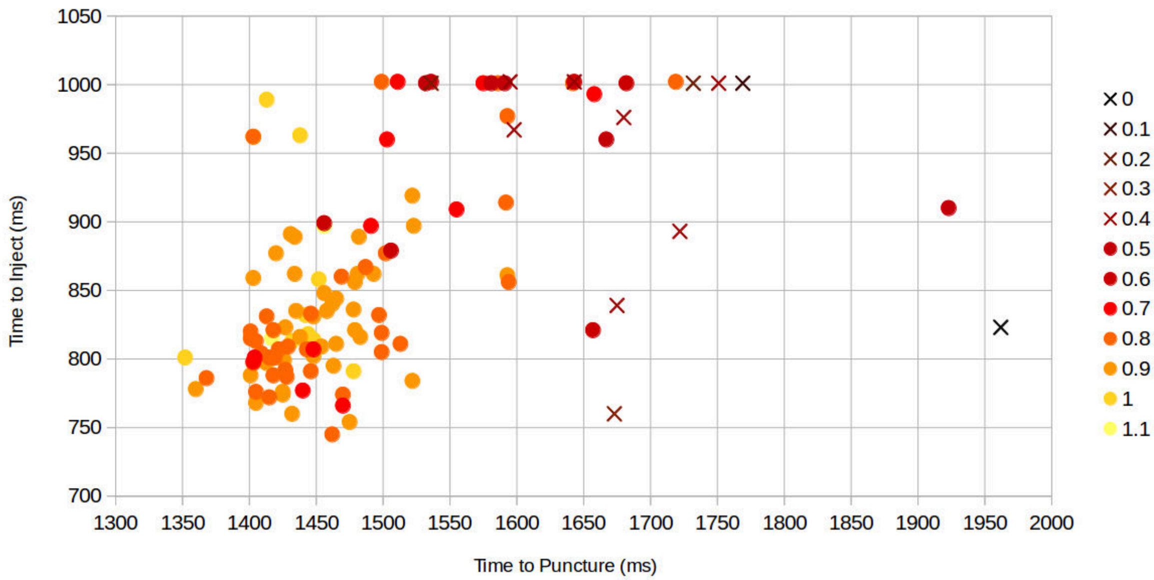


Figure 2.15: Injection amount for 120 injection tests

Figure 2.15 shows the same plot, except this time colored by injection amount. Injection amounts less than 0.5 mL are marked by X's, as these are unlikely to result in a successful ignition. Low fluid injections occurred at high times to puncture or inject.

2.3.2 Loess Canyon Rangeland Alliance Prescribed Burn

The first UAS-Rx prescribed burn was conducted with the Loess Canyon Rangeland Alliance [3] in south-western Nebraska. It required coordination with the fire council of the area (which includes the land owners) and the Federal Aviation Administration. Under the guidance of the burn boss, we targeted an area of approximately 40 acres (0.16 km^2), within a larger effort to ignite over 2000 acres (8 km^2), and involved about 60 fire-fighters for a full day. We performed 5 flights over 3 gullies that were overgrown with Eastern Red-Cedar (an invasive evergreen tree species).

Our ignition plan was to hover about 10 meters over the cedar trees and drop multiple ignition spheres in each spot to ensure ignition. However, we learned that due to the flammability of the cedar trees, a single ignition sphere was sufficient to ignite a large portion of the gully. The left side of Figure 2.16 shows the paths of the five flights we performed and the spots where the UAS-Rx dropped ignition spheres. Note that the UAS-Rx was able to ignite locations within or behind thickly vegetated terrain that a human would have a difficult time accessing (see flight paths 1 and 2, at the top). All five flights successfully ignited their targets. The delay on the ignition spheres ensured that the fire started after the UAS-Rx had left the area.

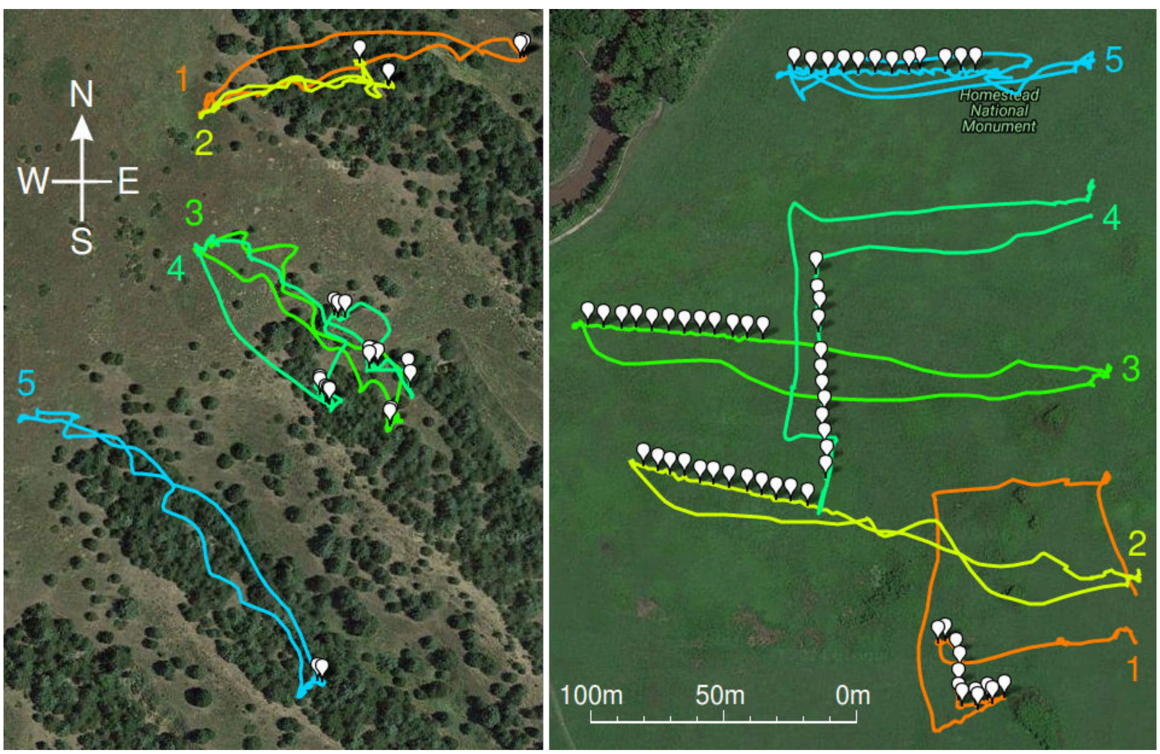


Figure 2.16: Flight paths and ignition sphere drop locations (white markers) at prescribed burn tests. Left: Loess Canyon Rangeland Alliance (LCRA), Right: Homestead National Monument (HNM). Both images are at the same scale. Map Data ©Google, Imagery ©DigitalGlobe, Map created at GPSVisualizer.com

This exploratory test was conducted with version 1.0.3 of the UAS-Rx that could hold 30 ignition spheres in an agitated hopper (see the cylindrical container on the UAS-Rx in Figure 1.4). Since a single ignition sphere can ignite a large area, we redesigned the UAS-Rx to use a gravity-fed chute in version 2.0.1, which holds fewer ignition spheres, but is lighter and provides a smoother ball flow. The dropper was redesigned to be able to apply more force, making it more reliable. In regards to the interface, we attached a downward-facing camera to the UAS-Rx so the operator can see if the UAS-Rx is above the target, and also see where the ignition spheres land.

2.3.3 Homestead National Monument Prescribed Burn

The prescribed burn at Homestead National Monument of America tested version 2.0.1 of the UAS-Rx. It required cooperation with professional fire-fighters and numerous government organizations (FAA, National Parks, Department of the Interior, and others), including needing special permission to fly a UAS at a national monument. This prescribed burn involved 22 firefighters, and burned 23 acres (0.09 km^2) in 2 hours. During this prescribed burn, firefighters with drip torches ignited the perimeter, while the UAS-Rx ignited the interior. Interior ignition is typically conducted by igniting a line of ground perpendicular to the wind. The downwind side of the line is quickly burned, and the fire runs out of fuel when it reaches the previously burned area. When that happens, another line is ignited. The UAS-Rx flights at this test sought to replicate this strategy.

The right side of Figure 2.16 depicts the Homestead National Monument burn area. The wind is blowing towards the South. Firefighters ignited a perimeter along the East, South, and West sides of the image. A typical flight proceeded as follows: we set up behind the East perimeter, launched the UAS-Rx to a height of about

Table 2.1: Prescribed Burn Flight Data

Flight	Flight Time	Round Trip Distance	Max Range	Battery Voltage before landing	# of Drops	Avg Dropping Altitude AGL
LCRA 1	4.62 min	270.79 m	122.82 m	10.784 V	4	16.38 m
LCRA 2	6.02 min	169.24 m	73.53 m	10.673 V	5	12.17 m
LCRA 3	4.52 min	257.31 m	100.76 m	10.821 V	2	14.66 m
LCRA 4	5.67 min	310.97 m	99.49 m	10.777 V	14	13.19 m
LCRA 5	4.47 min	346.90 m	151.46 m	10.764 V	2	20.42 m
HNM 1	5.67 min	373.73 m	96.06 m	10.830 V	12	11.05 m
HNM 2	5.53 min	429.34 m	195.56 m	10.535 V	12	17.49 m
HNM 3	4.73 min	420.40 m	200.86 m	10.946 V	12	20.39 m
HNM 4	4.88 min	466.42 m	157.37 m	10.988 V	12	17.23 m
HNM 5	6.32 min	456.07 m	116.60 m	10.691 V	12	16.11 m

15 m, and flew over the perimeter and 200 m into the interior. We then directed the UAS-Rx to fly back to us at a speed of 0.5 meters per second while dropping one ignition sphere every 8 seconds (one every 4 meters). After it had dropped all 12 ignition spheres, we directed it to return to us and land. The total flight lasts approximately 5 minutes, giving us over 5 minutes of reserve flight time. The right side of Figure 2.16 shows the flight paths of the 5 tests conducted at Homestead National Monument. Table 2.1 lists information about each of the 10 prescribed burn test flights.

The average dropping altitude was between 11 and 21 meters above the ground. This height was high enough to prevent the line of sight from being blocked by terrain or vegetation, and provided at least 7 meters of clearance over trees, bushes, and fire. Flying any higher would only increase the distance the ignition spheres could be carried by the wind as they fall. We have not yet extensively characterized how much the falling ignition spheres are affected by wind and the momentum of the vehicle, and in the future we would like to factor these effects into the UAS-Rx's software for more accurate placement of ignitions spheres. However, at these prescribed fires, dropping from directly above the desired ignition location was sufficient to ignite the cedars or the ignition line.

The longest flight was HNM 5, which lasted 6.32 minutes. For this flight, we were sufficiently far enough ahead of the fire line that we had time to fly back over the locations we dropped ignition spheres and collect footage with the downward-facing camera mounted on the UAS-Rx. Figure 2.17 shows several frames of this footage.

Figure 2.17: Video frames from a flyover of the ignition spheres dropped during the fifth flight at Homestead National Monument. Arrows point to locations ignition spheres were dropped.

Of the 12 ignition spheres that were dropped as part of flight HNM 5, only the tenth did not ignite. This ignition sphere took 15% more time to puncture than normal, indicating that the needle struck a thick spot on the shell of the ignition sphere, such as the seam or a rib, which may have obstructed flow of antifreeze into the ignition sphere. This ignition success rate closely corresponds to the 90% ignition rate found by the in-lab tests. After examining the logs during the other 4 Homestead flights, we inferred that 6 of the 48 ignition spheres were unlikely to ignite, based on the time it took to puncture and inject each sphere.

Despite the fact that some ignitions spheres failed to ignite, we did not discover any unburnt patches of land after the fire, as the fire from each ignition sphere was able to spread to cover the gap. Notice in Figure 2.17 that the fire from ignition spheres 1 and 2 have joined together. It is probable that the ignition spheres could be spaced further apart than the 4 meters we programmed and still yield a connected line of fire. This would allow the current prototype of the UAS-Rx to ignite longer fire lines. In Chapter 3 we describe how an integrated fire simulation can be used to preview the effects of a planned line of ignition spheres. This should help users choose a spacing of ignition spheres for their burn.

In addition to the downward-facing camera, we also attached a temperature sensor to the UAS-Rx. However, it didn't measure any abnormally high temperatures. It measured an average temperature of 24 C while the UAS-Rx was on the ground, and 17 C while the UAS-Rx was flying 15 meters in the air.

The average preparation time between flights at the Homestead National Monument was 5 minutes, which we wanted to reduce further. The latest design (3.0.1) took this into consideration by allowing ignition spheres and antifreeze to be quickly replenished without lifting up the UAS-Rx.

During these tests, we observed that the fire fighters' attention is heavily demanded by observing how the fire is progressing, and communicating over their hand-held radios. Manually directing the UAS-Rx requires the operator's continual focus, therefore more extensive autonomous flight planning would be beneficial. For example, the fire-fighter could draw the perimeter of the area that needs to be burned, and the UAS-Rx could autonomously plan the ignition lines and drop locations, take off, and complete the mission. This type of input drove the next iteration of the prototype.

2.4 Conclusion

Fire-fighters need new tools for interior ignition that are safe and cost-effective. This chapter described the design and evaluation of a prototype (version 2.0.1) unmanned aerial system to start prescribed fires from a distance (UAS-Rx). This unmanned aerial system was designed to safely and reliably puncture, inject, and drop ignition spheres, a commercial product designed for aerial ignition from manned aircraft. The UAS-Rx's mechanical and system design detect and help prevent failures, and reduce the severity of their consequences. The UAS-Rx has demonstrated reliability, with a 90% ignition rate and no mechanical or system failures occurring in hundreds of test injections, and it has demonstrated effectiveness, by successfully igniting the interior areas at two prescribed fires. The prescribed burn tests gave valuable insight into ways to improve the usability of the UAS-Rx, such as adding a downward-facing camera, reducing preparation time, and increasing autonomy.

This work demonstrates a great potential of unmanned aerial systems as an ignition tool. The mechanical design of the dropper can be further refined to be stronger, more light-weight, and easier to resupply. Although the UAS-Rx prototype presented in this chapter has a limited flight time and ignition sphere capacity, the modularity of our Dropper allows us to easily continue our work on a larger UAS for version 3.0.1. Furthermore, we can make the UAS-Rx capable of autonomously planning and flying missions with these scaled-up capabilities, which is the focus of the next chapter. These improvements should make the UAS-Rx a valuable tool for conducting prescribed burns safely and easily in the future.

Chapter 3

Fire-Aware Automated Aerial Ignitions

3.1 Introduction

Despite its potential, the UAS-Rx lacked the fire-awareness to operate autonomously and efficiently. Being aware of the location, direction, and general evolution of the fire is crucial not just to optimize the effect of the ignitions, but also to keep the vehicle and personnel safe. The UAS-Rx must go beyond planning trajectories to visit a set of waypoints and meet three key requirements. First, it must avoid visiting hot dangerous areas. Second, it must be able to drop ignition spheres in specific locations to assist in managing the fire direction and intensity. Third, it must leverage the knowledge and expertise of the burn crew to quickly adapt to the fire environment that can rapidly change.

Conceptually, the solution seems deceptively simple: integrate a fire simulation with a path planner. However, existing fire simulations like FARSITE [15] and FSPro [16] are intended to be run off-line, with intensive simulations of various fire scenarios to help the burn crew prepare before the fire. Yet, fires can be unpredictable in nature, especially when there are different types of vegetation, terrains, changes in wind speed and direction, etc. Since these fire simulators work in batch, they do not allow for quick user adjustments as the fire progresses.

We need a fire simulation system that can be run during the fire and can quickly adapt by leveraging the sensing capabilities on the UAS-Rx and user input. We want to take advantage of the fire crew's continuous fire assessment to correct the simulation's predictions as the environment affects the simulation. In addition, we want the crew to be involved in approving the generated ignition plans, and resetting the plan objectives as other non-modeled factors are considered, such as the location of the crew. To assist the operator's decision-making, the system must be able to leverage the simulation to show how the fire will behave after executing the ignition plan. Last, although the ignition planner itself does not need to be sophisticated, it must incorporate constraints to enforce the fire perimeter and reduce exposure to hot areas, and it must drop ignition spheres at the right location and intervals to implement pre-defined fire patterns.

To meet our objectives, we designed and implemented an approach that includes a specialized fire simulation and a planner that builds on it, both amenable to input from the system's sensors and the fire crew. The light-weight fire simulation is unique in that it can provide quick estimates of fire evolution and can be corrected by sensor and user input to counter the unpredictability of the environment. The ignition line planner is novel in that it generates a set of ignition sphere drop points and a path to reach them, using the fire simulation to avoid hot areas while dropping ignition spheres to perform, for example, the grid ignition technique shown in Figure 3.1. This is a common technique where the burn crew ignites a grid of spot fires inside the burn area, and the spacing and timing of the spots is used to regulate the

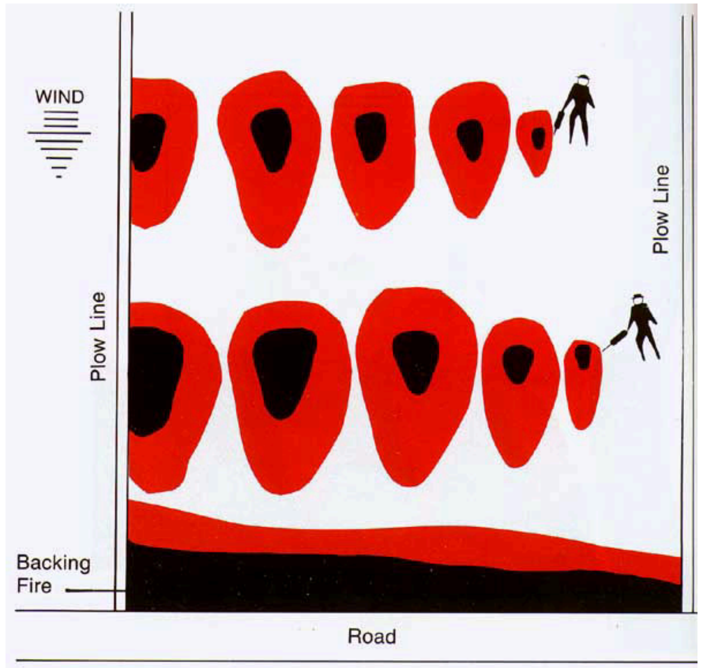


Figure 3.1: Grid Ignition Technique [37]

fire intensity. Our UAS-RX simply replaces the interior ignition personnel carrying the drip-torches, removing them from close proximity to the fire, and allowing the technique to be executed over difficult terrain with greater precision.

The sections in this chapter cover:

- An overview of version 3.0.1 of the UAS-Rx
- A light-weight fire simulation that can be corrected with real-time observations of the fire,
- An algorithm that uses the fire simulation to automatically plan ignitions for the UAS-Rx, and
- A preliminary study on the accuracy and usability of our fire simulation using fire observations input by a human.

3.2 System Overview

The previous chapter describes how our light-weight mechanism for puncturing, injecting, and dropping ignition spheres functions, and demonstrates how it can be used by a micro unmanned aerial system to ignite prescribed fires [10]. Our prototype UAS-Rx (version 2.0.1) could only carry 12 ignition spheres and fly for 10-12 minutes, which severely limited the missions that could be accomplished with it.

We have since increased the capabilities of the UAS-Rx in version 3.0.1 (Figure 1.3) by mounting our custom ignition sphere dropping mechanism on a DJI Matrice 600 hexacopter. The DJI Matrice 600 is 1.7 meters wide, 0.76 meters high, weighs 9.6 kg, and is powered by 6 129.96 Wh batteries. This allows us to carry up to 200 ignition spheres in a newly designed hopper and 500 ml of antifreeze in a new reservoir, carry additional sensors, and fly for up to 30 minutes on a single set of batteries. The total weight of these additions is 3.1 kg. The increased mission capabilities of this UAS-Rx drove the need for a semi-autonomous system.

Figure 3.2 shows the elements involved in controlling version 3.0.1 of the UAS-Rx. The UAS-Rx is controlled by a dual-joystick RC transmitter that also mounts an Android tablet. This tablet runs a custom Android application we created to control the UAS and the injection mechanism. This application has many of the features expected of a UAS-flying app, such as live video, avionics displays, satellite imagery, and waypoints, in addition to controls for dropping ignition spheres [8]. For further details on the app, see Appendix A.

Instead of using a separate radio link for controlling the Dropper, the Dropper communicates with the Android application over the radio connection between the remote controller and the Matrice 600. DJI provides an interface for this in the Android SDK, and a protocol for communicating over the UART port on the Dropper side.

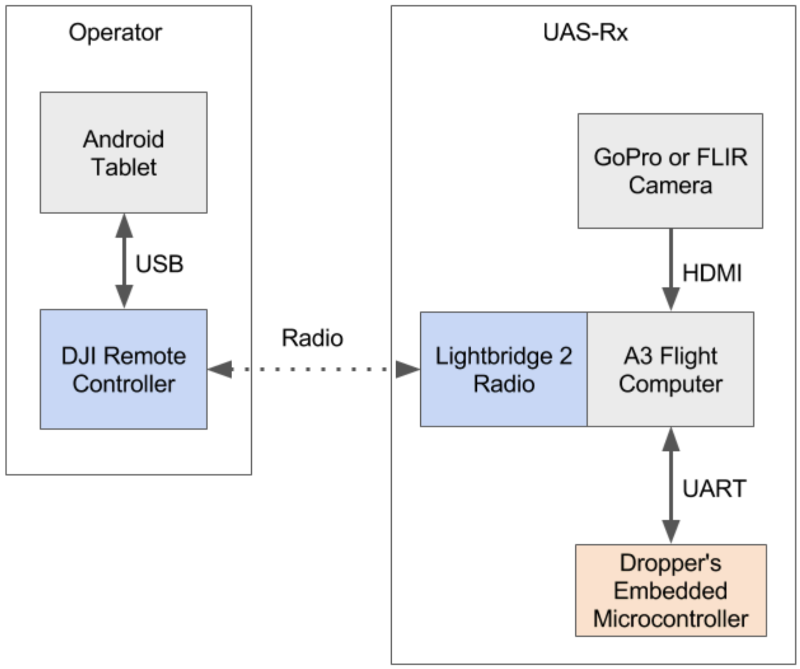


Figure 3.2: Communication and control components of version 3.0.1 of the UAS-Rx.

This UART port was previously used to communicate with the XBee Radio. However, we also wanted to continue to be able to use the XBee radio for easy debugging and development, and to keep the Dropper from requiring the use of a DJI vehicle. To accomplish this, the Dropper wraps each message in our own protocol's header and footer before wrapping it in DJI's protocol's header and footer before transmitting it, as shown in Figure 3.3. Messages from the Android App are automatically wrapped in DJI's protocol before being transmitted to the Dropper on the UART port.

The UAS-Rx protocol ignores everything except well-formed packets. Therefore, the Dropper doesn't even need to be able parse the DJI Protocol, as parsing for the UAS-Rx protocol will strip the DJI Header and Footer. Similarly, if XBee radios are used, the receiver only needs to parse the UAS-Rx protocol in order to strip the DJI Header and Footer off of the Dropper's transmissions. The UAS-Rx protocol's packet structure is very simple, consisting of only a start byte, a sequence number to detect

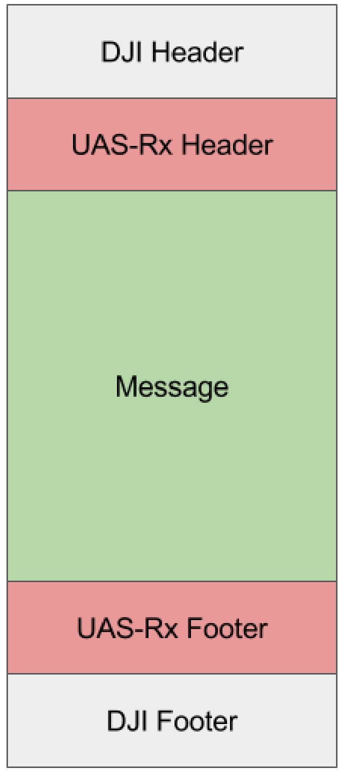


Figure 3.3: Protocol structure of a packet transmitted or received by the Dropper on the UART port.

duplicate transmissions, a data length, and a cyclic redundancy check.

The fire simulation and ignition line planning algorithms later described also run in this app and take advantage of the touchscreen interface, shown in Figure 3.4. The fire simulation darkens the areas of the map that have been burned, and the user can use the touch screen to make corrections to the simulation. The ignition line planning algorithm will display the planned line to the user so that they can decide whether to accept it or reconfigure it. The simulation can also render what the fire will look like in the future, which can help the user preview the effects of the planned ignition line before they decide to execute it.

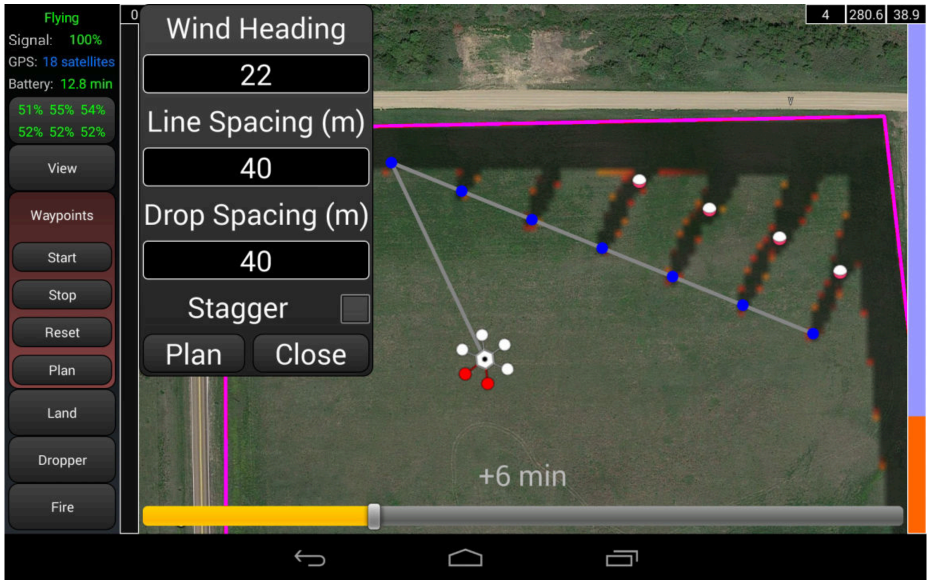


Figure 3.4: UAS-Rx Android application interface for planning ignition lines. The UAS-Rx is represented as a red and white hexacopter on the satellite map. A popup menu over the map has inputs for the ignition planner. The grey line and blue dots are the planned path and ignition locations. At the bottom is a yellow slider to project the fire simulation into the future. The slider is currently set at 6 minutes in the future, and the fire simulation shows how the planned ignitions would connect to the previous ignitions.

3.3 Correctable Fire Simulation

This fire simulation needed to be simple enough to be computed quickly and usable in the field on a tablet computer, therefore we made several assumptions to simplify the fire modeling problem. Long-term accuracy is already challenged by changing conditions, so we opted instead for a system that could leverage input from sensors and the user to make corrections to the simulation.

The fire simulation is composed of two main parts, shown in Figure 3.5. The first part is a fire simulation that simulates how fire spreads outward from point and line ignitions, and can compute at what time a point will ignite. The second part uses observations of the fire front's location at various times, and computes the error of the simulation at that location. These errors get interpolated to estimate the error at any particular point, and are used to improve the prediction of the fire simulation.

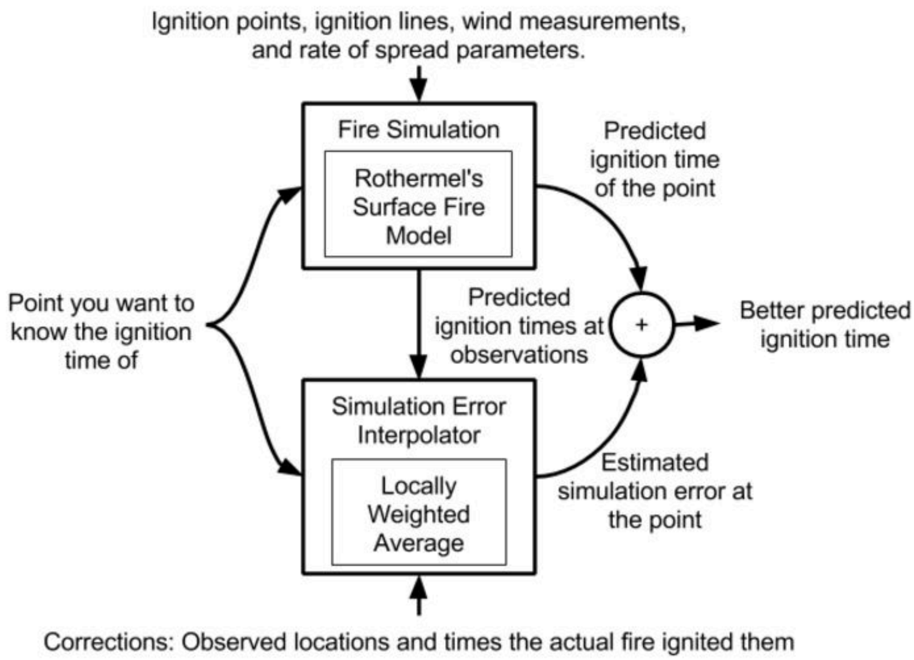


Figure 3.5: Dataflow diagram of the correctable fire simulation, showing the inputs and outputs of the modules.

3.3.1 Fire Spread Model

The fire spread model defines how fire spreads from an ignition source. To make planning easier, we wanted the model to:

- Describe how fire spreads outward from point ignitions caused by dropped ignition spheres and from lines ignited by the perimeter burn crew.
- Be a function that maps points on the ground to times that the fire front will reach that location.
- Be simple while accounting for changes in wind speed and direction, as wind can change several times over the duration of the burn and has a significant impact on how fire spreads.

The fire model we use defines how fire spreads from a point ignition, such as when an ignition sphere dropped by the UAS-Rx ignites on the ground. For fire spreading from other ignition sources, such as the lines ignited by the perimeter burn crew, we assume the fire spreads as though there were a point ignition at every point along the ignition source.

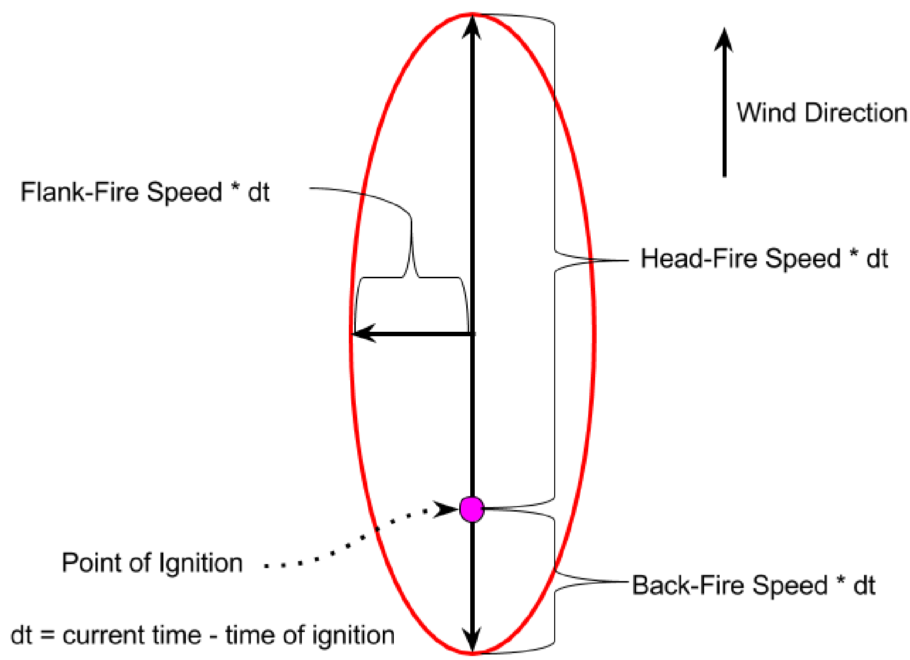


Figure 3.6: Fire spread from a point ignition

The simulation models the fire spread from a point ignition using an ellipse template [1], shown in Figure 3.6. The rate of fire spread is based on Rothermel's surface fire model [29], which provides equations that relate wind speed, slope, fuel moisture, and other factors to the head fire's rate of spread across the ground. The non-wind parameters are estimated and configured before the burn, and are assumed to be the same for every point on the terrain. Figure 3.8 shows an example curve for fire rate of spread as a function of wind speed. The back fire rate of spread is assumed to be equal to the head fire rate of spread at 0 wind [1]. Flank fire rate of spread is approximately equal to back fire rate of spread at low wind conditions [1], [7]. Because prescribed fires are typically conducted at low wind speeds, we also make the assumption that the flank fire rate of spread is equal to the back fire rate of spread for this simulation.

Once the fire's rates of spread have been computed, the equation for the ellipse in Figure 3.6 can be inverted to yield a function that calculates the time the ellipse will spread to a point on the ground. This is very convenient for planning paths and ignitions, as the simulation can quickly compute when the fire front will reach any location. For more details, see Appendix B.

3.3.2 Corrections

Corrections to the fire simulation are inputted as a location coordinate and a time that the actual fire front was observed to have reached that location. This kind of information could be extracted automatically by finding the fire front in the video from a thermal camera. However, our interface also allows the operator to manually make these corrections by touching a point on the map displayed on the touchscreen. This input format is mainly used to mark the present position of the fire front, but these coordinates and ignition times could also be inputted retroactively. This format can also be used to mark areas as unburned by inputting an ignition time after the present time, or to blacken areas by inputting an earlier time.

The corrections made by the human may not be perfectly accurate, however, it will still improve the accuracy of the fire simulation. The primary usage of this simulation is to plan ignition lines. However, the ignition line planner does not require an extremely accurate simulation in order to plan its ignition lines, and includes a safety margin to keep the UAS-Rx away from the fire even in the event of errors in the simulation caused by the user misplacing a correction. Additionally, in case the user makes a large mistake, we provide methods to erase or undo their corrections.

Whenever a correction is made, the fire simulation is used to compute the predicted time that point will ignite. The difference between this prediction and the actual time that point ignited is saved as the error. The correction point, actual ignition time, and simulation error measurement are saved in a list of corrections.

When the simulation is queried for an ignition time, the errors are interpolated using a locally weighted average to estimate the error of the simulation at the query point. The errors are weighted by a squared exponential function of the distance to the query point. Figure 3.7 shows how this interpolated error is added onto the predicted ignition time to correct the prediction.

3.4 Ignition Line Planning

The ignition line planner determines the next line of ignition sphere drop locations, and is initiated when the user presses the plan button in Figure 3.4. It leverages the fire model to predict where the fire will be to ensure that the UAS-Rx is never too close to a fire front and incorporates environmental conditions such as wind. It also incorporates feedback from users to ensure that the plan is appropriate given the current conditions and personnel positions. The approach is based on the grid ignition technique shown in Figure 3.1. A line of spot ignitions is placed orthogonal to the wind and offset from the backfire. The spacing between the lines and ignition spots can be configured to regulate the intensity of the fire and time to complete the burn [37]. The menu in Figure 3.4 has three main controls for the user to regulate the fire intensity and duration of the burn: the wind heading (wh), the line spacing (ls), and the drop spacing (ds).

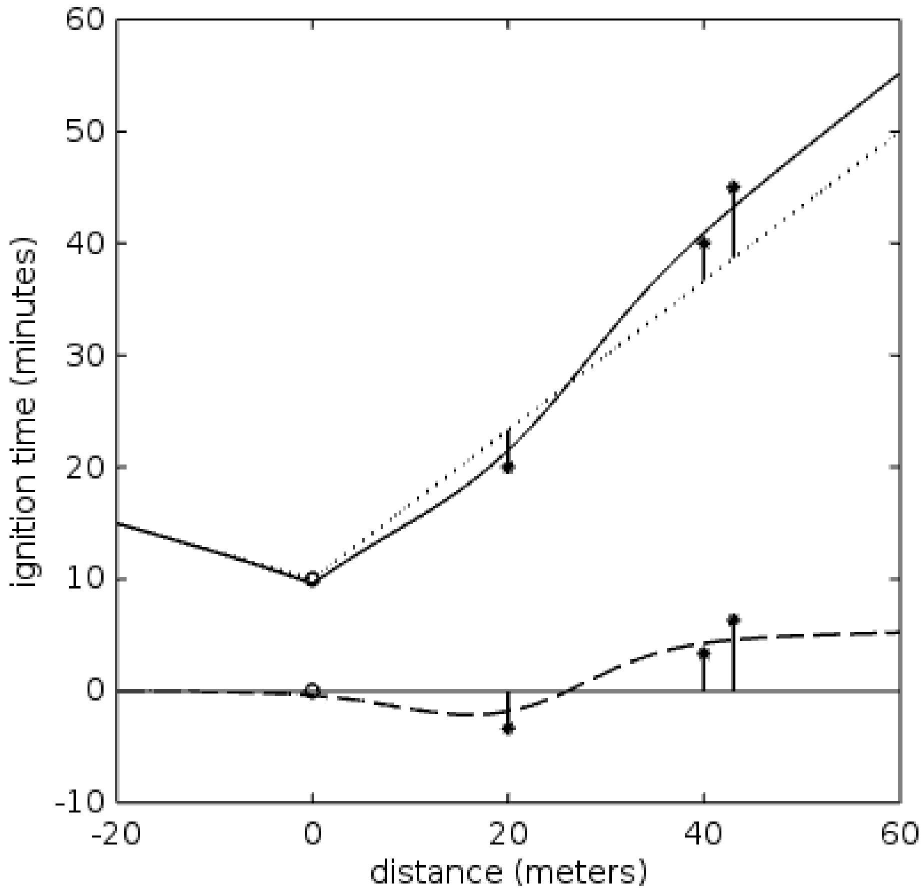


Figure 3.7: Example of how correction errors adjust the estimated ignition time of a point ignition at time 10 minutes. The dotted line shows the predicted ignition time spreading outward from the ignition location. Black dots are observations of the actual ignition time, which have an error from the predicted ignition time. These errors are interpolated using a locally weighted average, and plotted as the dashed curve on the horizontal axis. The solid curve is the sum of the dotted and dashed plots, and shows the corrected prediction passing through the observed ignition times.

Algorithm 1 shows the pseudocode of the ignition planner. Each call to the algorithm plans only the next line of ignitions (*plan*), represented as a list of waypoints to fly to and a set of locations to trigger ignition sphere drops. Each ignition line burns off the downwind portion of the unburnt area, so the UAS-Rx uses a polygon (*area*) to track the remaining area to burn between calls to the algorithm. This polygon initially starts as the control perimeter. In order to align the planned fires with the fires started by the drops of the previous line, a list of previous drop points on the perimeter of the unburnt area (*prev_drps*) is also managed by the algorithm. Additionally, the planner takes the fire simulation (*fsim*) and location of the UAS-Rx (*uas*) as input.

Algorithm 1: Plan Next Ignitions

```
Input  : Unburnt Area area, Previous Drop Points prev_drps, Wind Heading
         wh, Line Spacing ls, Drop Spacing ds, Fire Simulation fsim, Uas
         Location uas
Output: Waypoints and Drop Points plan
1  il ← findNextIgnitionLine(area, wh, ls);
2  if il is outside area then
3  |  return No Plan;
4  end
5  dap ← findDropAlignmentPoint(il, prev_drps);
6  pot_drps ← generatePotentialDropPoints(il, dap, ds);
7  Remove points from pot_drps that are not inside area;
8  ip ← planSafePath(uas, pot_drps, fsim);
9  if ip.numSafeDropPoints >0 then
10 |  return ip
11 else
12 |  area ← il.cutOff(area, prev_drps);
13 |  Goto 1;
14 end
```

Line 1 of the algorithm finds the vertex of the unburnt area polygon that is most downwind. From there, it moves upwind a distance equal to the line spacing to find the potential line the ignition spheres will be dropped along, orthogonal to the wind. At Lines 2 and 3, if the potential ignition line is outside of the control perimeter, then the prescribed fire is done, and no more lines can be planned.

Line 5 finds a point on the potential ignition line, *dap*, to align the drops with. This point is selected so that it is directly upwind of a previous drop location that is close to the center. This way the fire from this drop location will meet up with the fire from the previous drop location. If the drop spacing or orientation of the planned line differs from the previous line, then the fires started at the other drop locations will not exactly line up, but the total alignment error is minimized by ensuring a central drop is aligned.

Once the alignment point is found, a drop location is placed at the alignment point and others are placed along the line every drop spacing, ds. If $area$ is concave, this may result in placing potential drop points outside of the unburnt area. Line 7 ensures these are removed.

Line 8 plans a safe path from the UAS's current location to each of the potential ignition points, ip. The planner plans two paths, one starting from the leftmost potential ignition point and the other from the right, and ultimately picks the faster plan. For each drop location, the planner considers how to get there from the previous location. It first considers flying directly to the destination, and uses the fire simulation to check whether that path is safe. Safety is determined by sampling the fire simulation at points along the path and checking whether the ignition time of any point is less than the time that point would be reached (plus a safety margin). If the path is not safe, it instead considers ascending to a predetermined safe altitude and then flying and descending to the drop location. If the drop location is not safe when it would be reached, the planner skips that destination, and instead tries to reach the next destination from the current location.

Line 9 checks whether there were any safe drop locations in the planned line. If there were, then it returns the plan. If not, perhaps because the fire has already encroached to that point, Line 12 cuts that section off of the unburnt area and tries planning again until it finds a plan or returns no plan because the burn is complete.

After the UAS-Rx drops the last ignition sphere along a planned line, the area and previous drop points downwind of the completed line is trimmed from the unburnt area. The previous drop locations in the trimmed area are also removed because the algorithm is only interested in aligning with the drop locations along the perimeter of the unburnt area.

3.5 Preliminary User Study

We ran a preliminary user study to estimate how the accuracy of the fire simulation can be improved with human corrections, and to obtain feedback from users. This is just our first step in evaluating the correctable fire simulation, and we will use the information we gained to refine our system for future studies with members of the prescribed fire community. Future studies will evaluate the fire simulation on complex terrain and the acceptability of the plans generated by the ignition planner.

This study imitated how the fire-simulation would be used during UAS-Rx operation over a prescribed fire. We used the UAS-Rx to capture aerial video recordings of two prescribed fires in Eastern Nebraska. One prescribed fire was used for training the participants, and the other was used for the experiment. The participants watched portions of these recordings while correcting a simulated fire on a tablet to match the fire in the recording. We then assessed whether these corrections improved the simulation's accuracy.

3.5.1 Scenario

The prescribed fire we recorded aerial footage of for use in our experiment is typical of a prescribed fire in Eastern Nebraska. The burn was conducted on April 23, 2017 and covered a roughly square 125000 m^2 (30 acre) area of restored prairie. The terrain is mostly flat, with approximately a 10 m difference between the lowest and highest points. The grasses in the southern portion of the area were greener and wetter than the north, and caused the fire to spread slowly in this area.

The wind blew from the south and west with average wind speeds around 3.5 m/s and gusts up to 6.8 m/s. The perimeter burn crew ignited backfires along the north and east sides of the area as fire breaks, and then completed the burn by igniting headfires along the west and south sides. The UAS-Rx only observed the fire, it did not perform any ignitions, and there was no interior ignition crew. The burn took approximately two hours to complete.

The camera used to record the fire has a fish-eye lens that is able to see more than half of the area when the UAS-Rx flies 100 meters above the ground. Most of the video used in the study was taken at or around this altitude. The UAS-Rx patrolled the area to record the whole fire. The prescribed fire used for practice burned 72000 m^2 (17 acres) of grassland, and was recorded with the same vehicle and camera.

The simulated fire in this scenario propagates fire from lines along the perimeter. For the study, these lines were programmed to be automatically inputted into the simulation along where the perimeter burn crew ignited. In practice, this data would also have to be inputted during the burn by drawing a line segment over the ground the perimeter burn crew has ignited. For the study we made this automatic, as this would require periodically tracking the perimeter burn crew over the whole two hour duration of the burn, and we wanted to reduce the amount of video the participants would need to watch to just 15 minutes.

The simulated fire is blocked by a polygon representing the control perimeter. The wind conditions used in the study come from the National Weather Service's wind forecast taken a few hours before the burn. The wind was forecasted to blow from the south-southwest with wind speeds increasing from 2.5 to 4.5 m/s over the duration of the burn. The actual wind deviated from the forecasted wind, and had a significant effect on the evolution of the fire. A large portion of the area the western and eastern perimeter fires covered was when the wind blew from the west and east respectively.

The fire rate of spread parameters were estimated after the burn from the video recordings by measuring fire spreads under wind velocities measured with an anemometer at the site. The curve for the head fire rate of spread as a function of wind speed is shown in Figure 3.8. In practice, these parameters would have to be estimated before the fire, which could be done by measuring or estimating the fuel moisture content and then selecting a model from [31] based on the terrain type.

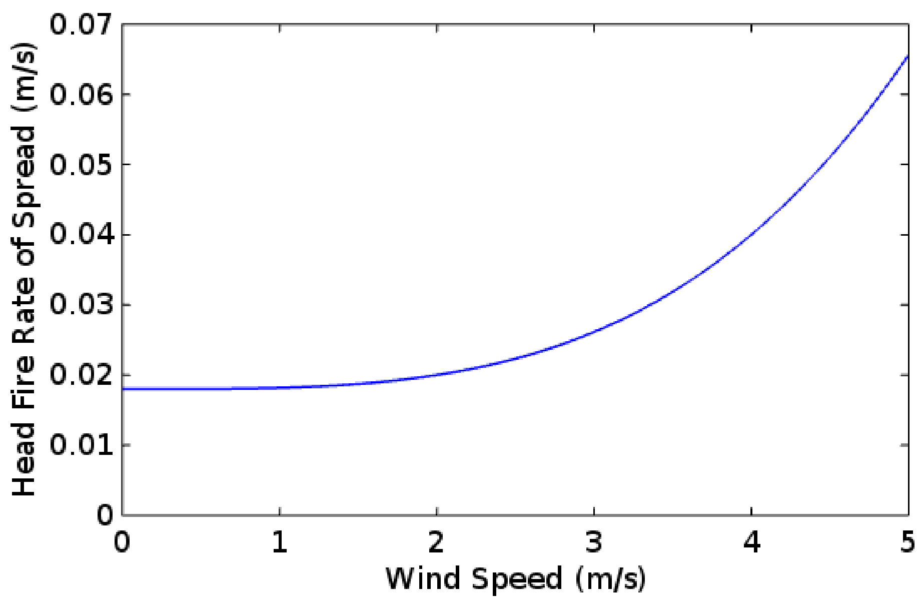


Figure 3.8: Head fire rate of spread curve used by the simulation in the study.

3.5.2 Participants

This study was conducted with five participants who are graduate students of Agronomy, Horticulture, and Applied Ecology. They all had been to prescribed fires, and four have been igniters at prescribed fires. The age of the participants ranges from 25 to 27, and each participant uses touchscreen devices and Google Maps at least weekly. The map on the interface uses Google Maps imagery and the default touch gestures for moving the map. The participants were contacted by email through a professor in their department, and they were given $15 in compensation for their time.

3.5.3 Experiment Procedure

3.5.3.1 Setup

The study was conducted with the participant in a quiet conference room and proctored by a researcher. A computer monitor was positioned on the table in front of where the participant sits, and was used to display the recorded video. The fire simulation was run on an Asus Nexus 7 tablet, which the participant used while sitting in front of the monitor.

When the participant arrived, they were greeted and asked to read and sign an informed consent form that informed the participant of their rights and briefly described the study. Next, the participant was asked to read a document that described how the correctable fire simulation will help the UAS-Rx perform ignitions at prescribed fires. After this the participant was asked to fill out a questionnaire about their prescribed fire experience and familiarity with touchscreen devices and Google Maps. Appendix C contains a copy of each document used in the study.

3.5.3.2 Training and Practice

For training, the participant watched a five minute long video that demonstrates how to use the fire simulation's interface on the touchscreen tablet. The video is a screencapture of the tablet running the fire simulation in the practice scenario with a narrator demonstrating and describing each function of the interface. Touches on the screen were represented by white circles so that the viewer can see the point of contact. Figure 3.10 shows a screenshot of the interface that the participants used.

Figure 3.9: A frame from the video the participants watched during the experiment.

Figure 3.10: The interface the participants used to adjust the fire simulation. The tools for adjusting the fire simulation are on the right side. The black pen tool marks areas as burned, the orange pen tool marks areas as currently burning, the green pen tool marks areas as unburned, and the eraser tool erases marks

Participants had five tools to interact with the fire simulation. The black, orange, and green pen tools are used to make corrections to the simulation by telling the simulation that the touched location ignited in the past, present, or future respectively. A mark appears at the touched location to remind the user of the correction, and the eraser tool is used to erase corrections by clicking on them. The undo button undoes the last correction or erasure. The training video also demonstrates that small features cannot be created with the simulation, as dense corrections get averaged out, and instructs the viewer that it's more important to adjust the main fire front to the correct position.

After the video, the participant was handed the tablet with the practice fire scenario running. The proctor asked the participant to try each operation demonstrated in the video, which included touch gestures for moving the map as well as usage of the tools.

Next the proctor led the participant through a practice run similar to how the experiment would be conducted. The tablet was reset to the practice fire scenario, and then was given to the participant while a synchronized video recording was started on the monitor. They were then asked to use the interface to correct the simulated fire to match the fire in the recording. The video ran for 3 minutes, but the participant was allowed to finish making corrections after that time.

3.5.3.3 Experiment

After the practice session, each participant was asked to read another document that explains how the experiment is structured, and describes and shows the burn plan that was distributed to the burn crew before the prescribed fire. Before the experiment, the participant also watches a one minute video recorded by the UAS-Rx taking off and flying over the burn area of the prescribed fire. This gives the participant an opportunity to become familiarized with the terrain and match landmarks to the map on the burn plan.

For the experiment, five three-minute segments were selected from the recorded video, and the tablet was programmed to be able to start the simulation from the beginning of each of these segments. The participants would view each segment in chronological order and use the tablet to make corrections to the simulated fire. The corrections made in each segment would persist on to the next. Each correction, erasure, and undo was logged. Figure 3.9 shows a frame from the video the participants watch, and Figure 3.10 shows an example screenshot of the tablet interface at that frame.

After the experiment, the proctor left the room and another person entered to interview the participant so that the participant wouldn't feel pressured against providing criticism in front of a researcher on the project. The participants were asked questions about how well they thought they did, how difficult it was, if it felt like they were making a lot of corrections, and something they liked about the interface and something they would like to change.

3.5.4 Results

To assess the accuracy of the fire simulation with and without user corrections, we first needed to obtain the ground-truth of where the real fire front was located at any given time. When possible, we directly used the video recordings to determine the real fire location. When the camera only provided partial views of the fire, we used a triangular mesh to interpolate the fire front location for the areas that were not readily visible. This interpolation is shown in Figure 3.11.

Given the ground-truth fire front location, we then computed the average distance

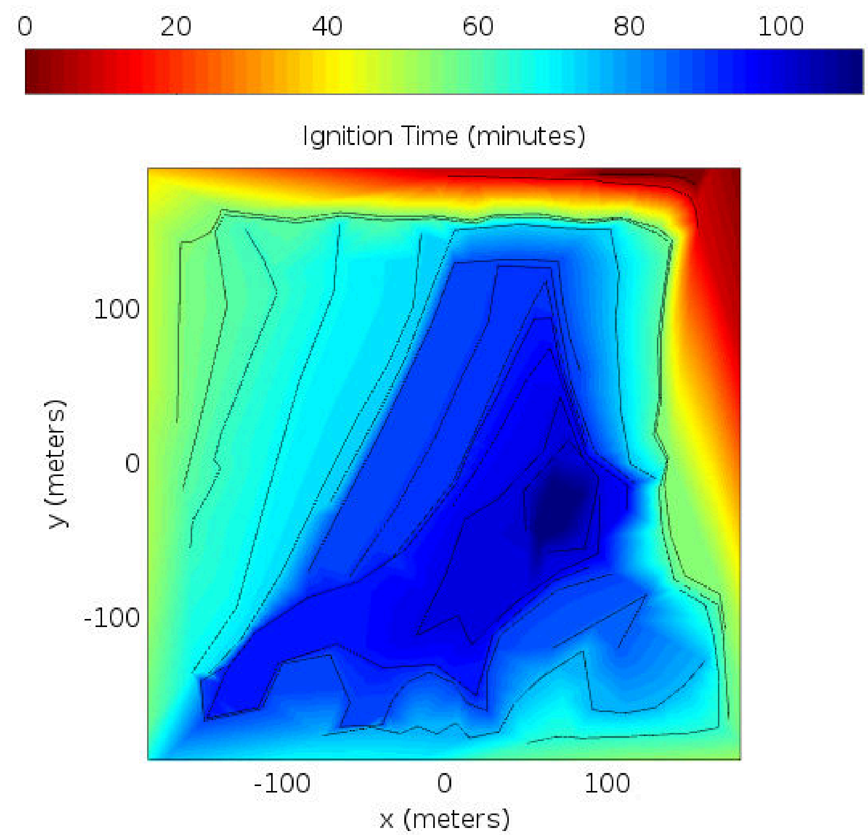


Figure 3.11: Ignition time map of the actual fire. Black lines show portions of the front sampled from the video at various times.

to the closest point on the simulated fire front, using a one meter grid, at every minute for the duration of the prescribed fire. Figure 3.12 shows an example of the fronts and their comparison using the interface from the experiment, with the simulated fire (red line) providing a close approximation to the real file (blue line).

Figure 3.13 shows the error distance over time. The thick black line in this graph shows the error of the simulation when no corrections are made, with a rapid descend early on in the fire and some period of error reduction as the simulation catches up to the fire front progress. The dashed lines show the error of the simulation with each participant's corrections. Over the period between the start of the first video segment and the end of the last video segment, the simulation with no corrections had

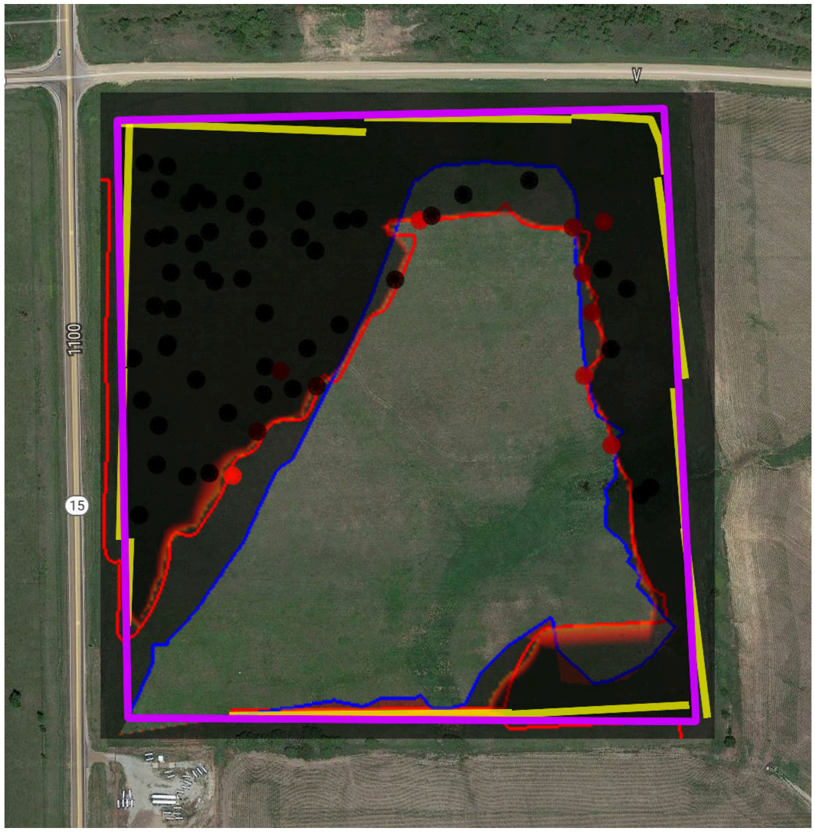


Figure 3.12: Example comparing the simulated fire (red front) with a participant's corrections to the actual fire (blue front) at 78 minutes into the burn.

an average fire front error of 24.1 meters, while the average simulation with participant correction front error was 15.3 meters (sample standard deviation: 2.1 meters). The results of this experiment are statistically significant, as the average participant error is 4.4 standard deviations from the error of the simulation with no corrections. The corrections yield on average a 34% reduction in error over the simulation without corrections, for minimally trained users operating an interface prototype. For some context, the 15.3 meter average error is 4.5% of the width of the 340 meter wide burn area.

The survey responses, summarized in Table 3.1, also indicate that the participants did not feel like their corrected simulation was a poor representation of the actual

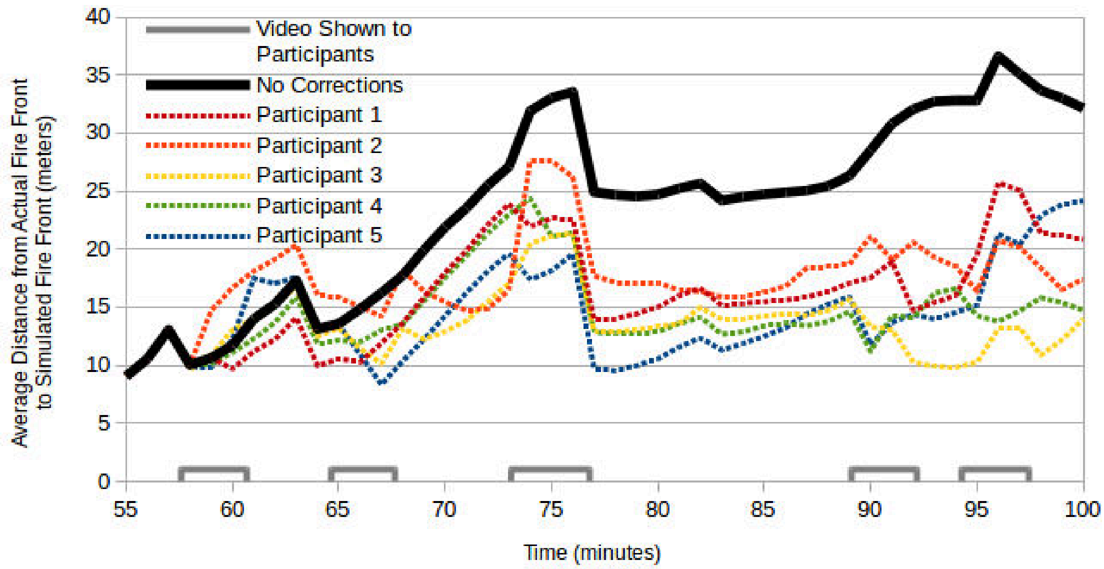


Figure 3.13: Error between the actual fire and the simulated fire with each participant's corrections, and with no corrections. The intervals at the bottom show the video segments shown to the participants during the experiment.

fire. That said, two participants mentioned that the blotchiness and smoothness of the simulation was difficult to work with, and we speculate that may be caused by the chosen resolution. One participant noted that it was difficult to represent patchy areas of the fire with the simulation and another noted that the monotony of the grassland made it difficult to match the video to the map, and it was easier to match the shape of the fire than the exact position. Features like roads, creeks, and telephone poles helped, but there were not many in parts of the video. Another participant stated that the fish-eye lens distortion was confusing, but seeing the UAS-Rx on the map helped figure out the location. We will leverage this feedback in our future work.

The participants made from 77 to 175 corrections in total, but the number of corrections did not have a strong correlation with the final accuracy of the model. The effort in providing those corrections, however, weighted on the three participants with the most corrections, who reported that they felt they were making lots of them.

Table 3.1: Summary of Participant Responses

Question	Participant Responses		
How well did you think that your fire simulation with the corrections matched the fire you saw in the video?	2 Good	3 Neutral	0 Bad
How hard or easy was it to correct the simulation to match the simulated fire to the fire in the video?	1 Easy	2 Neutral	2 Hard
Did it feel like you were making a lot of corrections?	1 Few	1 Neutral	3 Lots

Three participants also suggested a pen-like tool to draw a continuous line or curve to mark where the fire front is instead of placing individual dots. This is again something we will consider in future work.

Chapter 4

Future Work

This thesis presented an unmanned aerial system for prescribed fires (UAS-Rx). Chapter 1 defines the problem space and contribution. Chapter 2 introduces using unmanned aerial systems as an ignition tool, and demonstrates aerial ignition at two prescribed fires. Chapter 3 extends the capabilities of the UAS-Rx to autonomous ignition over a dynamic fire through the use of a correctable fire simulation. This fire simulation is evaluated by a preliminary user study and demonstrates that real-time user corrections can greatly improve the accuracy of the simulation.

As future work, we plan to address the participant's feedback by developing a more advanced cell-based fire simulation that can incorporate terrain elevation, fuel maps, and crown fire spread models. With a cell-based simulation, the corrections could be made by erasing or painting fire into the cells. However advancements will need to be made to make the cell-based simulation perform well on large areas.

The prescribed fire used in this experiment is favorable for the assumptions made in our current fire simulation. There are no trees, so Rothermel's surface fire spread model is applicable, and the terrain is mostly flat, so the slope's effect on fire spread is negligible. We plan to collect data from a prescribed fire at a forested area with many ravines to see if a correctable fire simulation can still be useful under more complex conditions.

This preliminary study sampled 5 participants from within the university. We plan to reach out to prescribed fire crews in and around Nebraska, to conduct a larger study with more people who are likely to use this technology in the future. This study will also be used to get feedback on the ignition lines generated by the planner.

The fire and fuel parameters for this fire simulation have to be calculated by tools external to the system. To make the system more readily usable, we plan to create a built-in interface for selecting fire simulation parameters by selecting the appropriate model from [31], and inputting readily available weather information such as the drought index, forecasted temperature, and humidity.

The fire observations in the UAS-Rx currently come from the human operator. We plan to also have these observations be automatically made by parsing video from a thermal camera mounted on the UAS-Rx. This will help with making corrections to the fire simulation in the absence of notable landmarks for the human operator to reference.

Wind direction has a very significant effect on the rate of spread of a fire, and the updrafts caused by the fire can make the wind blow from different directions at different parts of the fire. However, the wind direction can be estimated by the direction the smoke is blowing off the fire front. As another point of future research, we would like to develop vision processing algorithms to estimate wind direction and speed from smoke, so that this information could be fed back into the fire simulation for better fire predictions.

This work has a large potential for swarm applications. Many burn crews have multiple interior ignition personnel igniting parallel lines in order to hasten the burn.

The ignition line planner presented in this work could be easily extended to multiple UAS-Rx by planning several sequential lines and having each UAS-Rx ignite one. Each UAS-Rx can use its sensors to contribute to a shared fire simulation for a greater coverage of the fire.

The UAS-Rx could also be used to communicate with firefighters on the ground using aerial motions and gestures. For example, if the UAS-Rx detects a spot fire outside the burn perimeter, it could direct burn personnel to its location.

The UAS-Rx has a huge potential for assisting prescribed fires not just by making them easier, safer, and cheaper to ignite, but also through sensing, fire simulation, and providing information to the users. We believe that exploring these possibilities will allow the UAS-Rx to exceed the expectations of burn crews and result in an ignition tool that revolutionizes prescribed burning.

Appendix A

Android Application Interface

A.1 Introduction

In order to increase the ignition sphere capacity and flight time of the UAS-Rx, we transitioned from the Ascending Technologies Firefly that we were familiar with using to a DJI Matrice 600. DJI provides a Mobile Source Development Kit (SDK) for creating an Android application that can control one of their unmanned aerial systems. We decided to create our own application, as this would allow us to communicate with the Dropper through the Matrice 600's radio link instead of a separate XBee link, and give us a highly customizable framework to prototype new technologies to use with the UAS-Rx, such as the fire simulation and ignition line planner. By the end, this fully-functional app was composed of 67 java classes and 17,500 lines of java code, not counting the DJI SDK. This appendix describes the main functions and interfaces of this application.

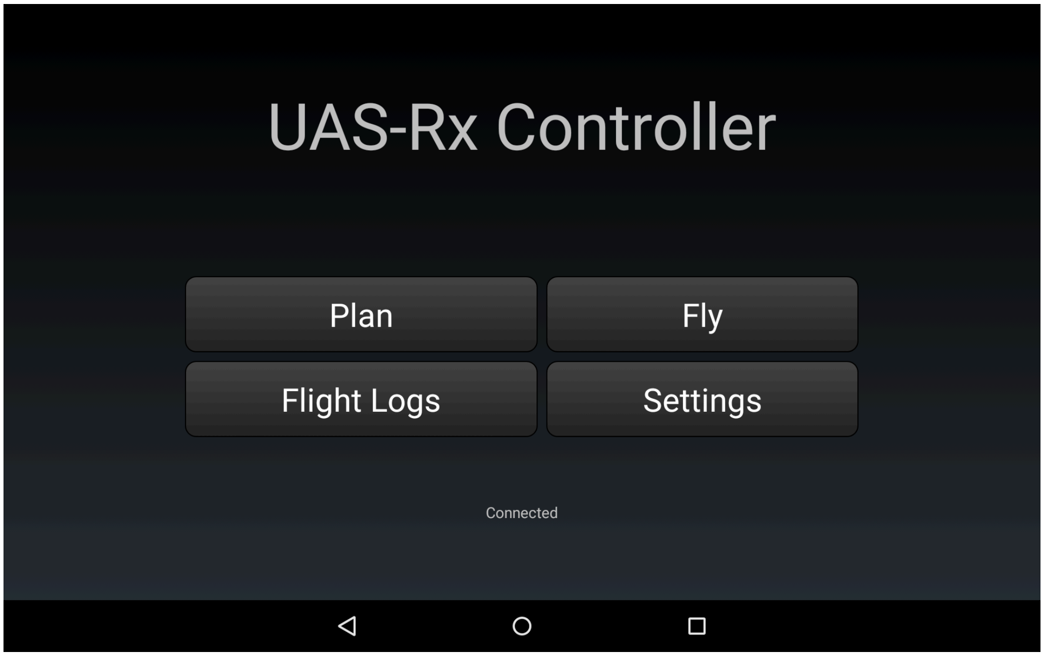


Figure A.1: Main menu of UAS-Rx app.

A.2 Main Menu

Figure A.1 shows the initial screen of the application. There are four main buttons to take you to different activities within the application, as well as a text string that displays the current connectivity status to the UAS-Rx. The DJI SDK needs to register itself before its functions can be used. The first registration requires an internet connection to DJI's servers, after which it downloads a local key that can be used offline for registration. Once successfully registered, the application will attempt to connect to the controller via a USB cable, and then connect to the UAS via the controller's radio link. If the aircraft or controller become disconnected, the application will periodically attempt reconnection until the connection is reestablished.

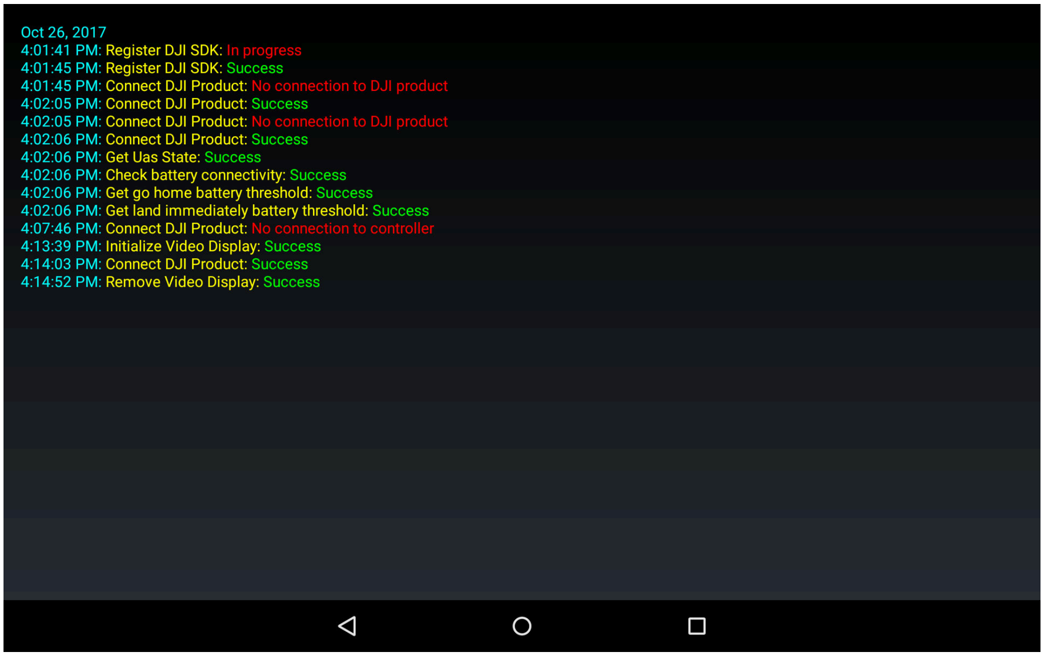


Figure A.2: Software log.

A.3 Software reliability and error handling

Figure A.2 shows a hidden debug screen displaying a log of the state of various software operations that the application has attempted. The DJI Mobile SDK is continuously under development and being improved, but some functionalities are still quite complex. Many operations in the DJI Mobile SDK require the user to define a multitude of functions for the SDK to callback when an event occurs, and many functions can possibly return errors that need to be handled, resulting in a multitude of possible software states.

For example, let's consider the simple task of sending the UAS to a set of waypoints. To do this you need to define callback functions for the upload progress (e.g. 30% uploaded), the start of the execution of the waypoint mission, and the end. You must then load your waypoints into the mission manager, which may return an error. After that, you must tell the mission manager to upload the mission to the UAS, which may also return an error. Additionally, you must define a callback for when the mission is done uploading, which may be called in case of failure. Once the mission is done uploading, you must then start the mission, which may return an error. Again, you must define a callback for when the mission is done being started, which may also be called in case of failure. Furthermore, there is no guarantee these callbacks will be called.

Enforcing order out of this madness was essential to creating a functional UAS-Rx. To do this, each intuitive operation is encapsulated within an Action class. Running the action will execute all of the necessary DJI SDK calls to perform that action, and in a safe manner. The Action class ensures that the action cannot be re-executed while it is already in progress, and cannot be executed if a prerequisite Action has not succeeded (such as if you are not connected to the UAS). Furthermore, the Action class wraps everything inside try catch blocks to prevent the application from crashing in the event of an unhandled exception. The status of the action is saved to a human-readable string, which is automatically logged, as shown in Figure A.2. These actions are contained inside a class that wraps the entire DJI SDK into a simpler abstracted interface. Some of these actions also implemented workarounds for bugs we discovered in the SDK until they were fixed in later versions of the SDK. We also contributed to the development of the Mobile SDK by carefully documenting bugs and reporting them to the developers.

A.4 Burn Planning

In order to use some of the more intelligent features of the UAS-Rx, such as the fire simulation and the ignition line planner, the UAS-Rx requires some knowledge about the burn plan which can be inputted before the actual prescribed burn. The plan

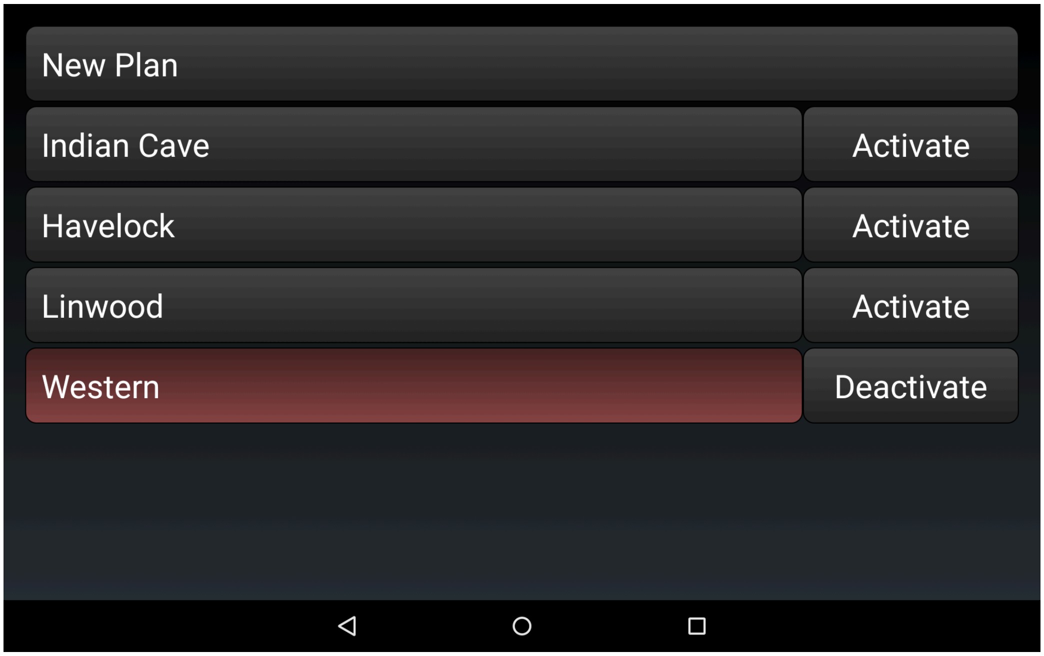


Figure A.3: Burn plan selection screen.

button in the main menu (Figure A.1) brings you to the screen shown in Figure A.3, where you can create a new plan, edit an existing plan, or activate a plan when you are actually going to burn that area.

Figure A.4 shows the activity to edit a plan. Burn crews typically distribute a map of the burn area and important features. This can be loaded into the app as an image and positioned on the map, so that the operator can reference the map in relation to the UAV and fire simulation during flight. The app also allows the operator to draw the perimeter of the burn. This defines the area that the UAS-Rx is allowed to drop ignition spheres in, and the ignition planner will attempt to burn this entire area. Below this, the user can input the expected wind, which will be used by the fire simulation if there isn't real-time measurements. The user can also input the expected head fire rate of spread for 0, 2, and 4 meter per second winds. These three data points define the function curve to compute the head fire rate of spread from any wind speed. These three numbers have to be estimated before the burn based on

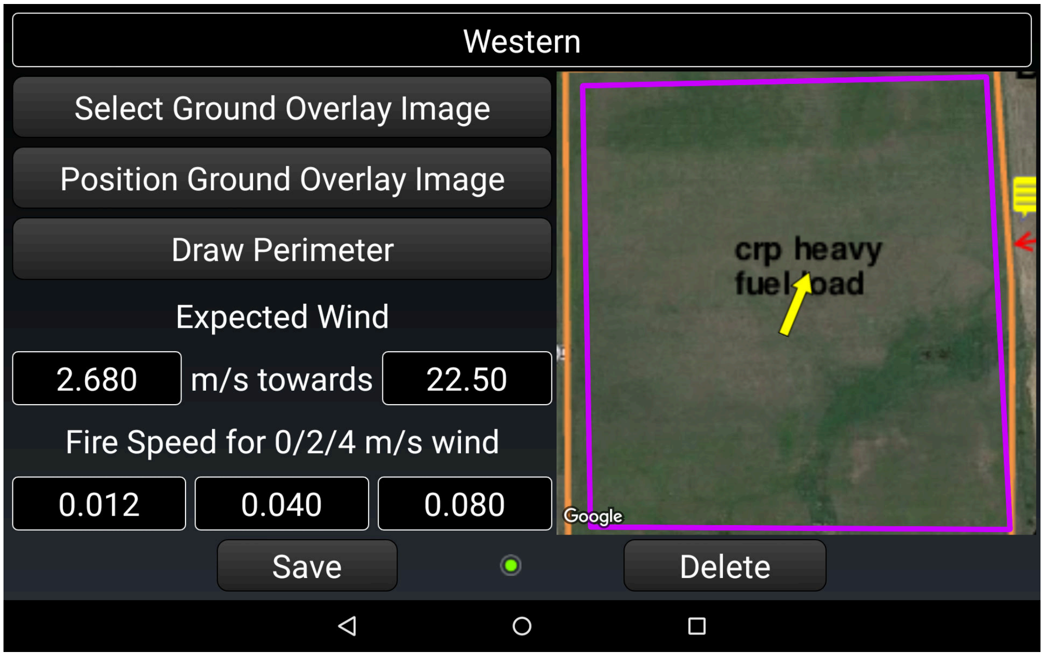


Figure A.4: Burn plan editing screen

the type of fuel and its moisture content, but are then used by the fire simulation to propagate the fire front.

A.5 Flight controls

The main activity for flying the UAS is shown in Figure A.5. This activity has many features necessary for useful for flying any UAS. The majority of the screen shows a Google Maps satellite map of the area. This map can be zoomed, rotated, or scrolled using the common touch gestures. On the map is an icon of two red circles and four white circles connected to a central hub. This icon represents the UAS-Rx, which is a hexacopter. The icon moves around on the map to represent the current latitude and longitude of the UAS, and also pitches, rolls, and yaws to indicate the attitude of the UAS. If the user clicks on the UAS icon, the map will scroll and rotate to keep the UAS centered and facing upward. Clicking again decouples the map from the

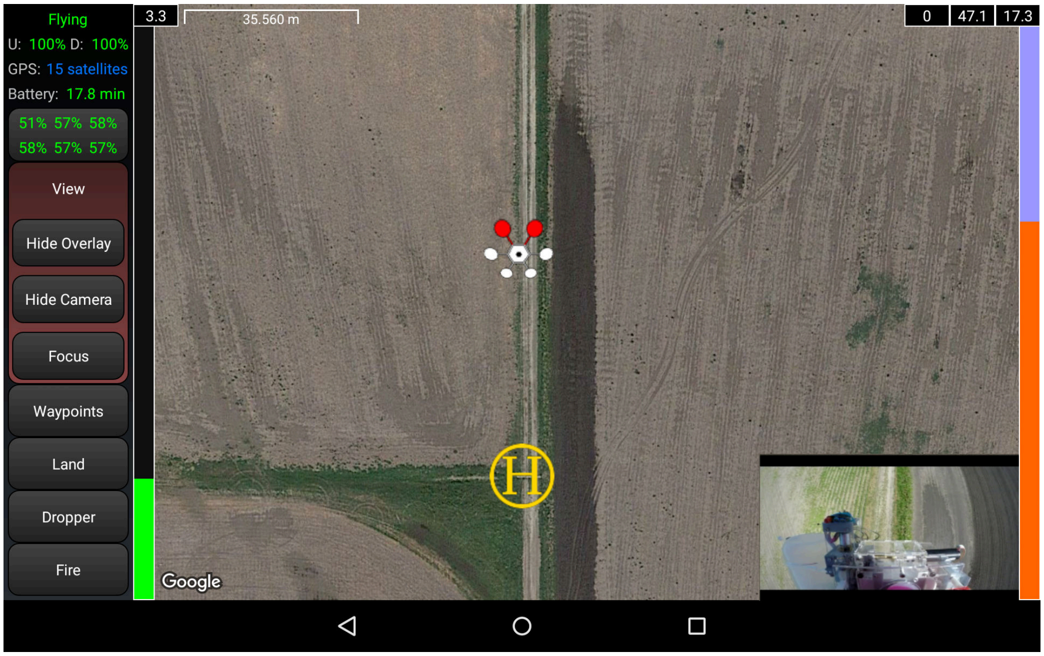


Figure A.5: Flight screen with no active burn plan

UAS. The yellow H is the home point of the UAS, which is the point the UAS took off from, and will return to if it loses connection. In the bottom right corner of the map is a live camera feed from a GoPro Hero 4 mounted on the UAS. Touching this will expand the view to take up the whole screen.

At the top of the map is a white scale bar indicating how long that distance is on the map. The app is currently set to display all units in metric, so distances are shown as meters, and velocities are shown as meters per second. Framing the map are two bars indicating the velocity and altitude of the UAS. The left bar fills with green from the bottom as the UAS travels faster. The number directly above the bar is the velocity. The right bar fills with blue from the top as the UAS ascends. The orange part represents the ground (or fire), so the more blue, the more distance from the ground. Above that are three numbers, from left to right, indicating number of ignition spheres dropped, horizontal distance to home, and altitude.

In the top left corner of the screen are more status displays. At the top is the operational status of the UAS, which is currently "Flying". In this state, the remote control's joysticks control the drone's motion and yaw. When the drone is on the ground, the state could be either "Motors Off", "Motors On", or "IMU Preheating". There are also several states in which the operator has limited control: "Auto Takeoff", "Auto Landing", "Waypoints", and "Going Home". Beneath the status are uplink and downlink radio signal strength, then GPS signal strength. Next is the estimated flight time remaining, calculated by the average current draw from the batteries during flight. Beneath that are the power percentages for all 6 batteries. Clicking this will allow you to select one of the batteries, and then show you detailed information about the selected battery, shown in Figure A.6.

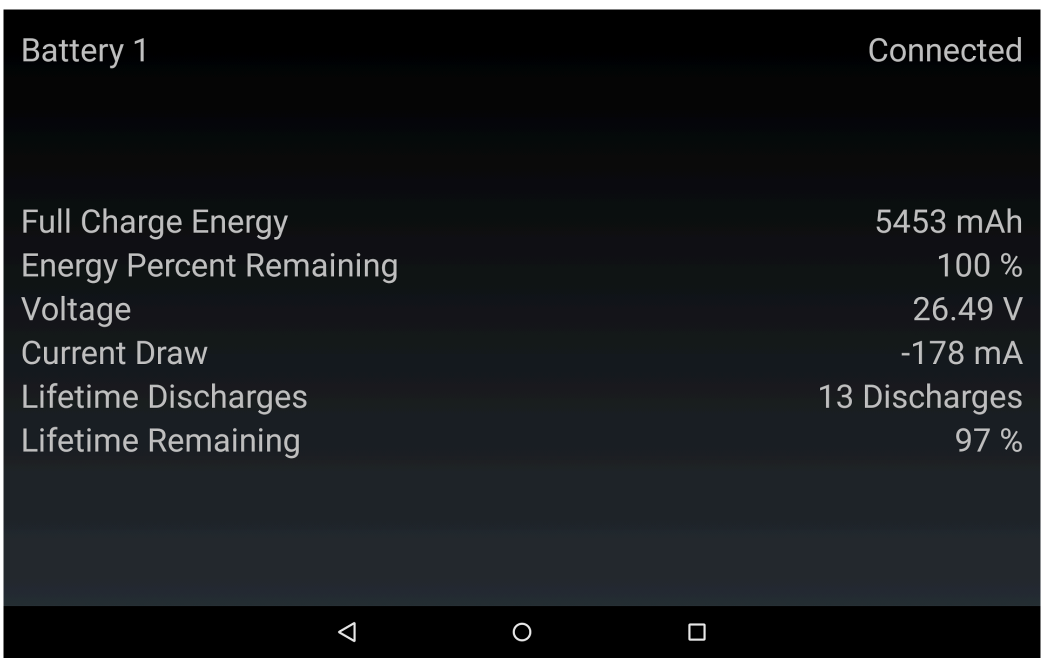


Figure A.6: Battery details screen.

Below the battery status are 5 tabs containing more controls: View, Waypoints, Land/Takeoff, Dropper, and Fire. Clicking one of these tabs will open up its relevant controls, and hide the controls of the other tabs. The View tab is currently active in Figure A.5, and has buttons to show or hide the burn plan's overlay image, show or hide the camera feed, and focus the map on the UAS.

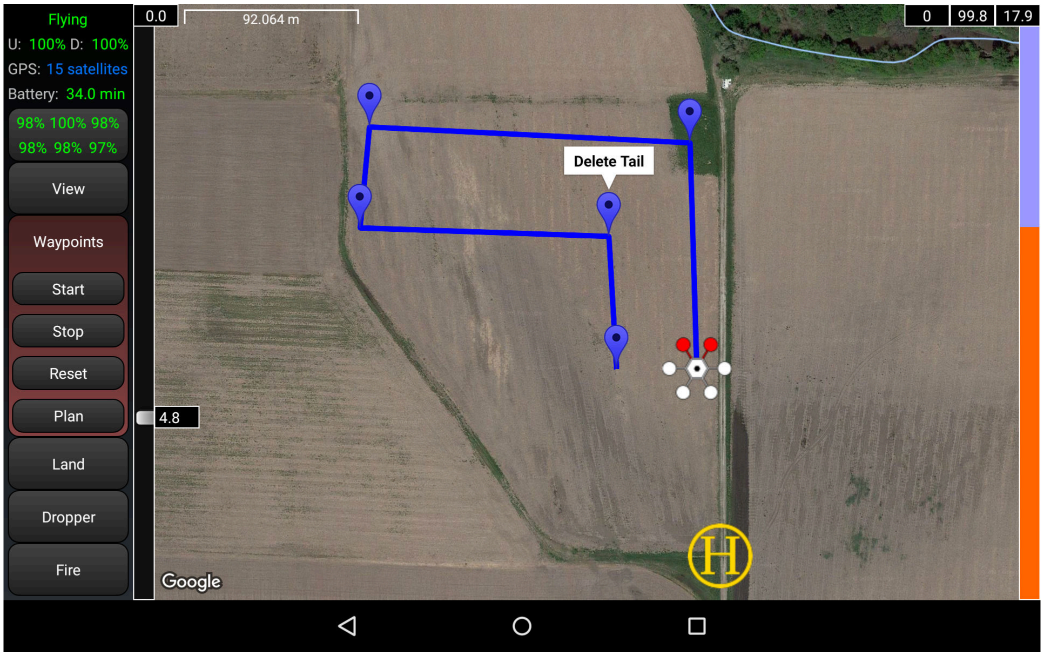


Figure A.7: Configuring waypoints for autonomous flight.

Beneath the View tab is the Waypoints tab, which allows the operator to send the UAS-Rx to GPS coordinates autonomously. Waypoints can be created by touching locations on the map, and blue markers will appear there. If marker is clicked, a box saying "Delete Tail" appears over the marker. Clicking the box will delete that marker and all the following markers on the path. Clicking and holding on a marker will pick it up so it can be dragged around. The cruising speed of the UAS is set by a slider along the velocity bar. The UAS will maintain a constant altitude during flight. Once the user is satisfied with the waypoint mission, the user presses the Start button to upload the waypoint mission to the UAS and have it begin autonomously flying along the waypoints. Figure A.8 shows the UAS flying along waypoints.

Once the UAS starts the waypoint mission, the waypoint path turns pink, and the UAS status changes to Waypoints. This signals to the user that the remote controller's joysticks have modified functionality. Yaw can still be controlled, but

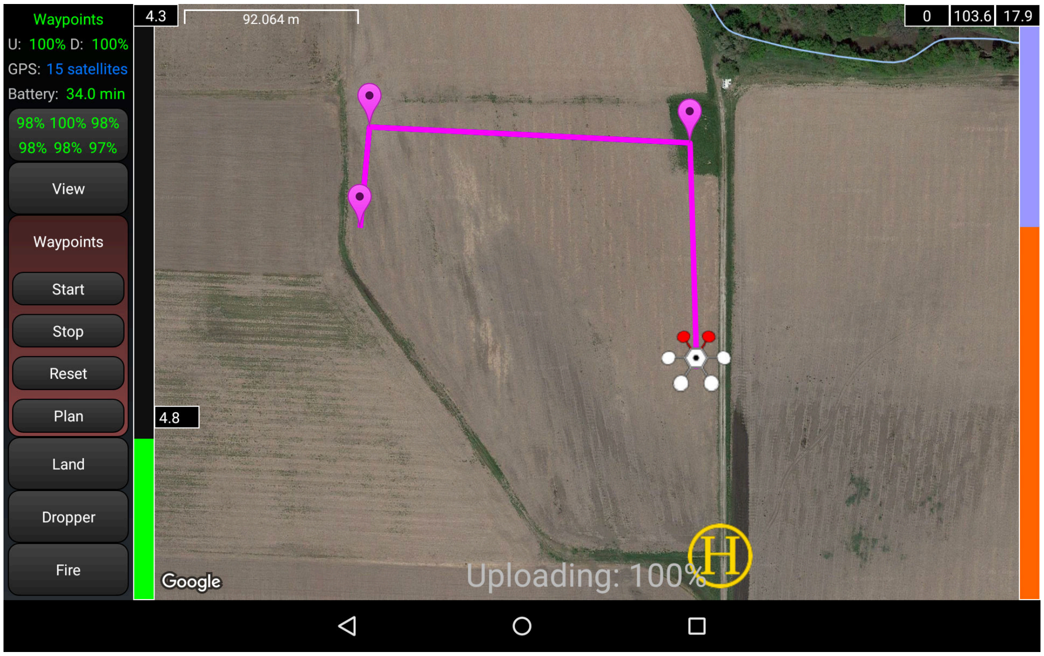


Figure A.8: Autonomously flying along waypoints.

altitude, pitch, and roll cannot. Pitch on the controller now controls the velocity of the vehicle, with neutral input being the configured cruise speed. The UAS will automatically slow down and stop at each waypoint. When the UAS gets close to a waypoint, that waypoint marker disappears. Autonomous flight can be stopped by pressing the Stop button in the Waypoints tab, which will put the UAS back in the Flying state. Unreached markers will remain on the map and turn blue so the waypoint mission can be easily resumed by pressing start again. The reset button stops the current waypoint mission and removes all markers. The plan button will be discussed later.

The Land Tab has buttons for starting and stopping an auto landing. If the UAS is on the ground, then the Land Tab changes into the Takeoff tab, which has buttons for starting and stopping the auto takeoff.

The functionalities described up to this point can all be used to fly any DJI Matrice 600. The UAS-Rx carries the Dropper and integrates the fire simulation and ignition planner, requiring additional controls. To use these additional features, a burn plan must be created and activated (Figures A.4 and Figures A.3). When a plan is active, the burn perimeter is drawn on the map as a purple polygon, as in Figure A.9. This figure also shows the actual burn plan used at a prescribed fire here overlaid on the satellite imagery.

Figure A.9: Burn plan overlaid on map, and burn perimeter.

A.6 Dropper controls

The Dropper tab shown in Figure A.9 allows manual control of the ignition sphere dropper. There are two safeties in place on the Dropper to prevent accidental ignitions. First, ignition spheres cannot be dropped outside the burn perimeter defined by the purple polygon. Second, the user must press the Arm button to arm the Dropper and intentionally allow ignition spheres to be dropped. While the Dropper is disarmed and the UAS-Rx is on the ground, the Reload and Advance buttons are shown in the Dropper Tab. These control the syringe on the dropper and can be used to pump antifreeze through all of the tubes. It is necessary to do this on the ground before taking off to ensure that antifreeze will be injected into the ignition sphere instead of air that was in the tubes.

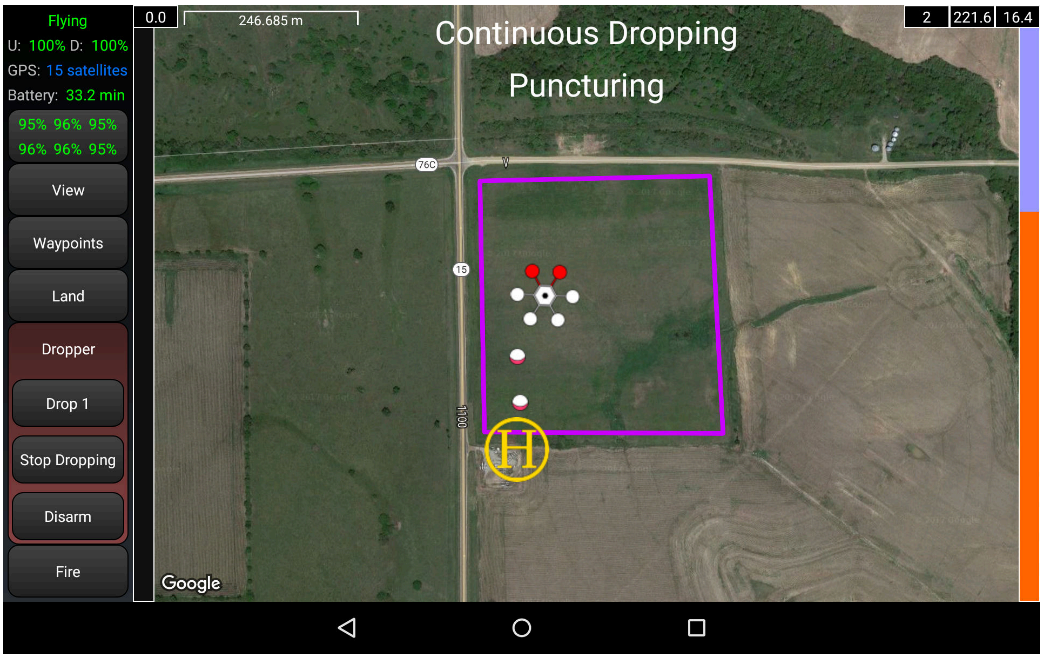


Figure A.10: Manually dropping ignition spheres.

When the Dropper is armed and the UAS-Rx is inside the burn perimeter, the Reload and Advance buttons are replaced by the Drop 1 and Start/Stop Dropping buttons. Pressing the Drop 1 button will inject and drop 1 ignition sphere. Pressing the Start Dropping button will cause the UAS-Rx to periodically drop ignition spheres (the period can be configured in a different menu). Regular ignition lines can be created by combining this with the autonomous waypoint following. Simply place waypoints at the start and end of the line you want to burn, start the waypoint mission, and when the UAS-Rx reaches the first waypoint, start continuously dropping. Adjusting the speed of the UAS-Rx and period of the dropping can adjust the spacing between ignitions.

While the UAS-Rx is continuously dropping, "Continuous Dropping" is displayed at the top of the map. Pressing the Stop Dropping button, disarming the Dropper, or leaving the perimeter all stop continuous dropping. The Dropper is continuously transmitting status messages to the application, and its current operation is displayed below "Continuous Dropping". When an ignition sphere drops, a white and pink circle appears on the map at that location. In the worst case, if the Dropper injects an ignition sphere and is unable to drop it, a red "Fire Danger" is displayed on the app.

A.7 Fire Simulation controls

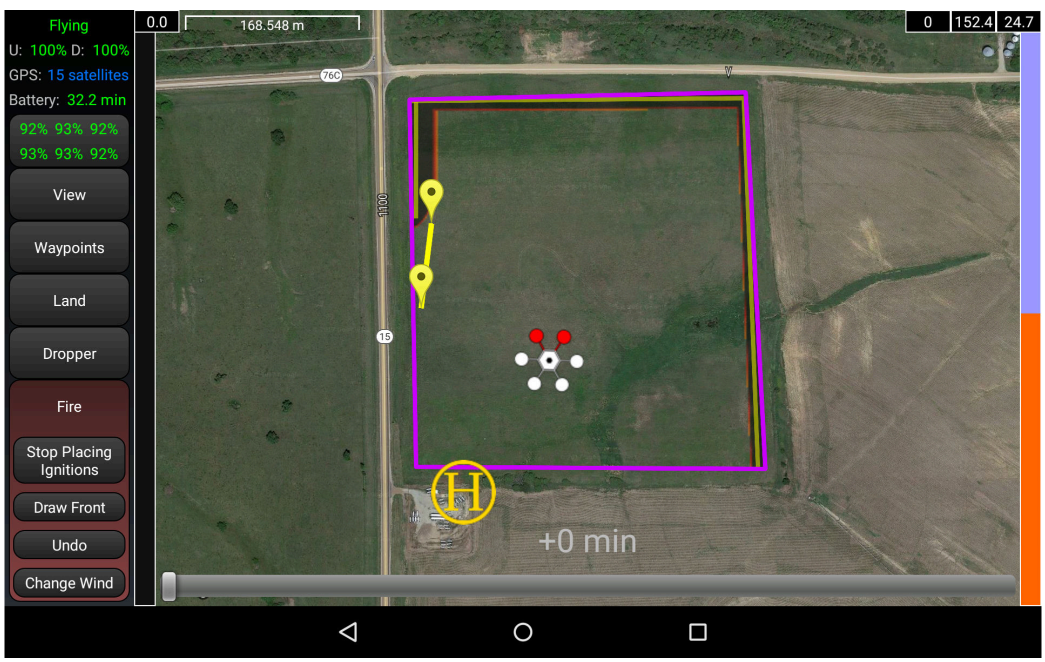


Figure A.11: Marking locations the perimeter burn crew has ignited.

The Fire tab has controls for the fire simulation. The first button in this tab is the Place Ignitions button, which allows the user to mark lines on the map that are being ignited by the perimeter burn crew (see Figure A.11). The controls for this are identical to the controls for placing waypoints. Once the user is done placing ignitions, the fire simulation will begin simulating fire spreading outward from those points and lines. The fire simulation darkens the burnt areas of the map, and colors the fire front reddish orange. Each ignition sphere that the UAS-Rx drops is automatically incorporated into the fire simulation as a point ignition.

Figure A.12: Adjusting the fire front.

The draw front button allows the user to adjust the simulated fire front. Touching locations on the map will place an orange dot there, and the fire simulation will adjust so that the fire front passes through that point, as shown in Figure A.12. The user can use this to correct the simulation to match the actual fire that is being captured in the UAS-Rx's camera for better ignition planning.

The undo button undoes the last correction or ignition line placed by the user. The change wind button allows the user to update the wind speed and direction. The fire simulation will continue simulating the fire from the currently burned area under the new wind conditions.

The underlying fire simulation is continuous, but for rendering a large area, it is evaluated at discrete intervals. Making a change to the fire simulation by placing a new ignition, making a correction to the front, or changing the wind requires recomputing the ignition time of each discrete point. These computations are done asynchronously in parallel by multiple threads, and at increasing resolutions over time. For lower resolutions, the ignition times are interpolated to give the appearance of a higher resolution until the higher resolution computations are done. This allows the rendered fire to instantaneously react to the user's input, without sacrificing detail.

A.8 Ignition Line Planning

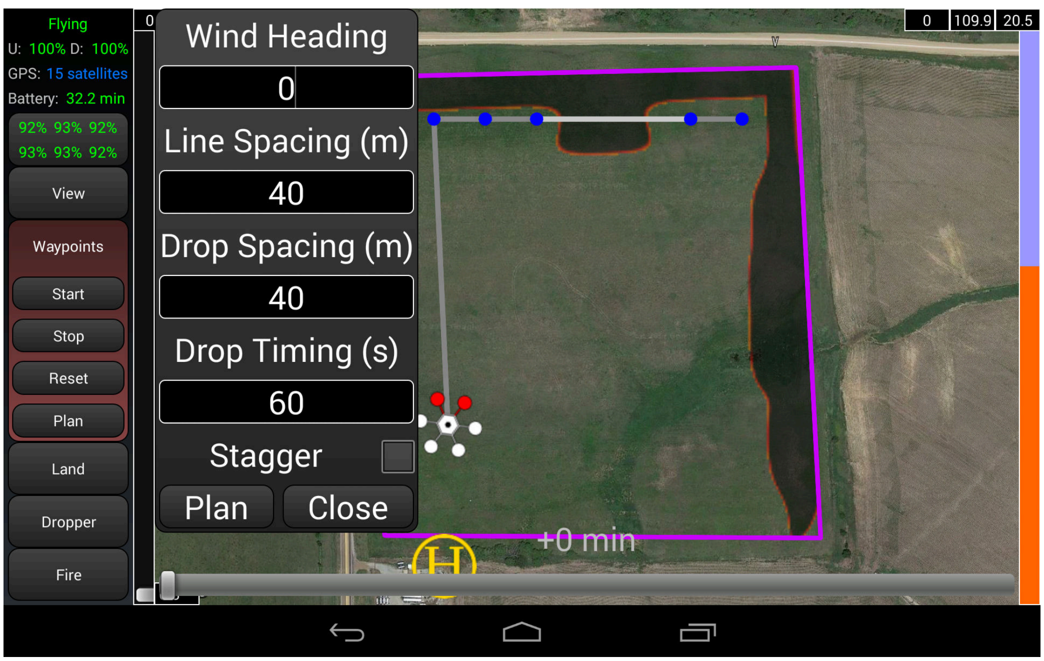


Figure A.13: Planning an ignition line.

Returning to the waypoints tab, the plan button opens a menu for automatically planing an ignition line, shown in Figure A.13. This allows the user to configure the orientation of the line, the distance between lines, the distance between ignition sphere drops, the timing between ignition sphere drops, and a checkbox to stagger the drops with the previous line. Pressing the Plan button generates the blue dots shown in Figure A.13, which represent spots to drop ignition spheres. The gray segments will be flown at the current altitude, however the white segment will be flown at a higher altitude to avoid the fire.

When the plan menu is closed, the user can place additional waypoints for the UAS to fly to after the planned ignition. Pressing the Start button will upload the waypoint mission and make the UAS-Rx fly along the planned line, automatically dropping an ignition sphere at each point. Figure A.14 shows the UAS-Rx flying and dropping ignition spheres at the planned locations. Each ignition line builds off of the previous line. The next line can be planned while the UAS-Rx is executing one line (as shown in Figure A.14), or after.

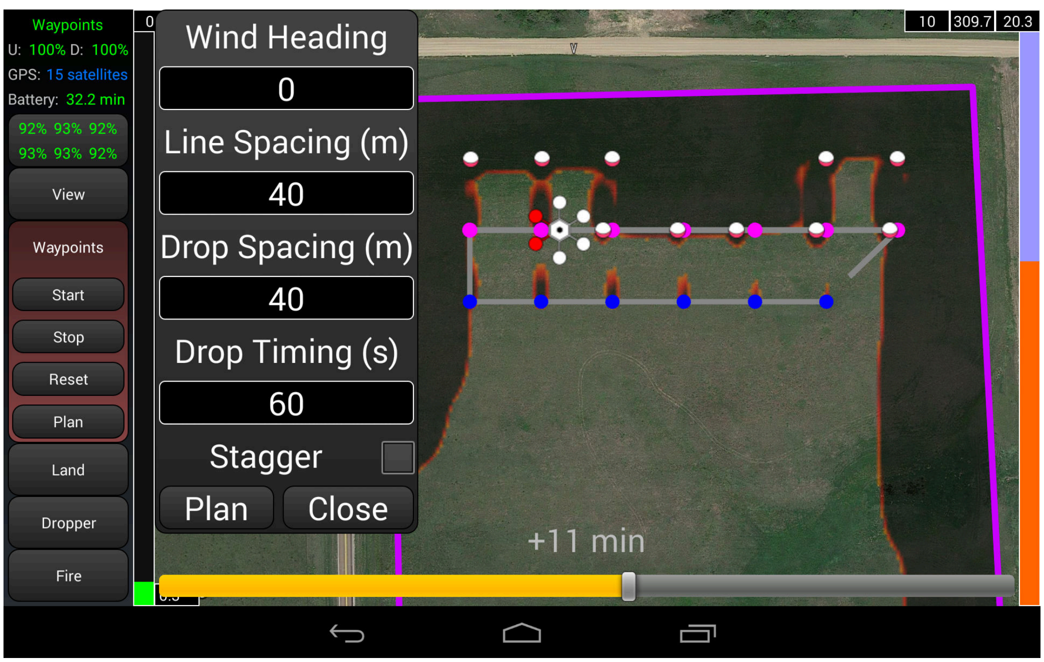


Figure A.14: Planning an ignition line off a previous ignition line. The fire simulation time slider is set to view the fire 11 minutes in the future, and shows the predicted effects of the planned ignition line.

The yellow and white slider at the bottom of the screen in Figures A.11 and A.14

can be used to see what the fire looks like in the future. When planning an ignition line, the fire simulation also incorporates the planned ignitions into its simulation, which allows the user to see the predicted effect of the planned line.

A.9 Conclusion

This application contains many unique features for conducting aerial ignition at a prescribed fire with a UAS. The fire simulation and ignition line planner are integrated into the main user interface for flying the UAS, which allows for them to be used during flight over a fire. The features shown here were rapidly developed as prototypes and proof-of-concept, and the layout and design of the graphical user interface can definitely be improved to further enable aerial ignition.

Appendix B

Fire Simulation Implementation

B.1 Fire spread from a point ignition

The fire simulation models fire spreading from a point ignition using an ellipse template, as shown in Figure 3.6. The following algorithms show how this ellipse template was implemented. Algorithm 2 details how to compute the shape of the ellipse at any point in time after ignition. This equation is inverted in Algorithm 3, which computes the time the ellipse will spread to any point in space. Both of these algorithms assume a constant wind speed and direction.

Algorithm 2: Fire spread from a point ignition as a perimeter

Input : Point Ignition Location (x_i, y_i), Point Ignition Time t_i, Wind Direction w_d, Back Fire Velocity v_b, Head Fire Velocity v_h, Time of Interest t

Output: Perimeter Ellipse (x_{center},y_{center},$orientation$,r_{major},r_{minor})

$t_d \leftarrow t - t_i$;

$x_{center} \leftarrow x_i + cos(w_d)(v_h - v_b)t_d/2$;

$y_{center} \leftarrow y_i + sin(w_d)(v_h - v_b)t_d/2$;

$r_{major} \leftarrow ((v_h - v_b)/2 + v_b)t_d$;

$r_{major} \leftarrow v_b t_d$;

return $(x_{center}, y_{center}, w_d, r_{major}, r_{minor})$;

Algorithm 3: Fire spread from a point ignition as ignition times

Input : Point Ignition Location (x_i, y_i), Point Ignition Time t_i, Wind Direction w_d, Back Fire Velocity v_b, Head Fire Velocity v_h, Point of Interest (x, y)

Output: Ignition time of (x,y): t

/* Translate and rotate the coordinates to be relative to the ignition */

$x_t \leftarrow x - x_i$;

$y_t \leftarrow y - y_i$;

$x_r \leftarrow x_t * cos(w_d) + y_t * sin(w_d)$;

$y_r \leftarrow x_t * sin(w_d) - y_t * cos(w_d)$;

/* Compute how quickly the major and minor radii grow */

$r_{minor} \leftarrow v_b$;

$r_{major} \leftarrow (v_h - v_b)/2 + v_b$;

/* Compute the coefficients of a quadratic equation */

$a \leftarrow r_{minor}^2 - 2r_{major}r_{minor}$;

$b \leftarrow -2x_r(r_{major} - r_{minor})$;

$c \leftarrow x_r^2 + y_r^2 r_{major}^2 / r_{minor}^2)$;

/* Use quadratic formula to compute ignition time */

$determinant \leftarrow b^2 - 4ac$;

if $determinant < 0$ **then**

  /* Determinant can be less than zero due to floating point errors */

  $determinant \leftarrow 0$;

end

$t \leftarrow (-b - sqrt(determinant))/(2a) + t_i$;

return t;

B.2 Line Ignitions

The point ignition simulations are useful for modelling how fire spreads from the ignition spheres the UAS-Rx drops, but can't easily model how the fire spreads from the continuous fire lines ignited along the perimeter of the burn area. In addition to ignition points, the fire simulation also allows zig-zagging chains of line segments to be inputted. Each vertex of the zig zag is treated as a point ignition, and the fire spreads outward from the line segments so that the fire front is parallel to the original line segment, and tangent to the elliptical point ignitions at the endpoints. An example of this is shown is shown in Figure B.1.

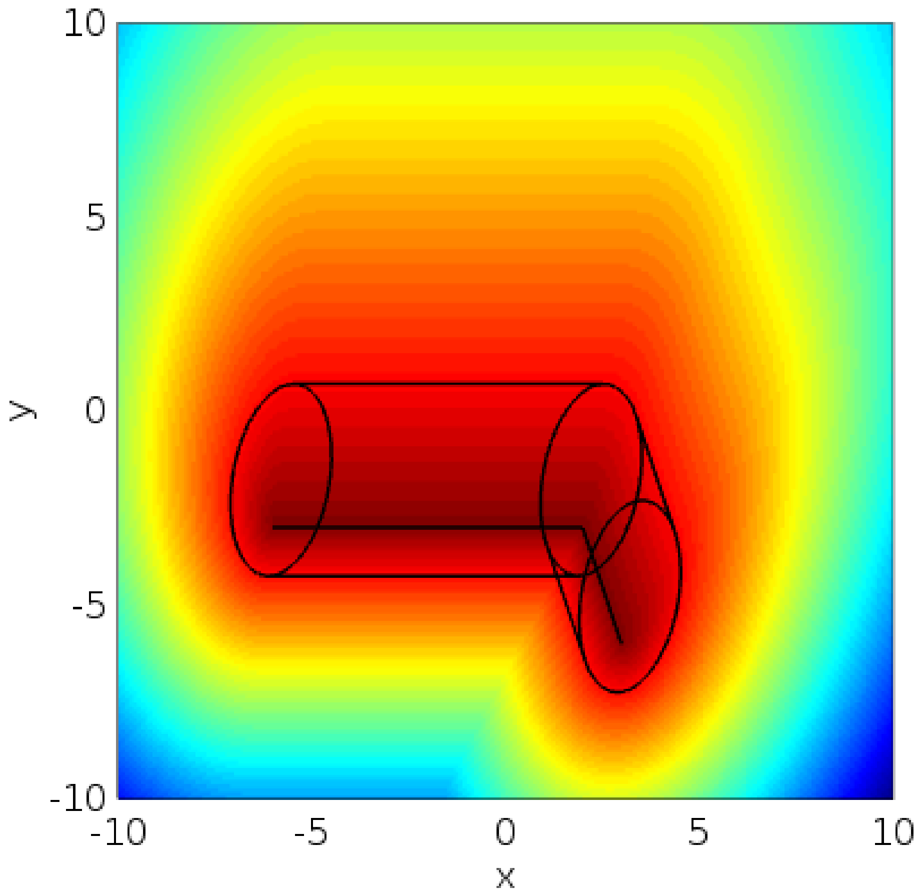


Figure B.1: Example ignition time of a fire started along two line segments.

B.3 Changes in Wind

At some time during the prescribed fire, the wind may change. This change could be measured by an anemometer and wind vane at the prescribed fire, inferred from the UAS's flight characteristics, or manually inputted. When the wind changes, the fire simulation saves the ellipses the point ignitions created, and continues simulating how fire would spread from these ellipses under the new wind conditions. We assume that fire spreads as if each point on the ellipse was a point ignition under the new wind conditions. Figure B.2 shows an example point ignition at the origin that spreads under two different winds.

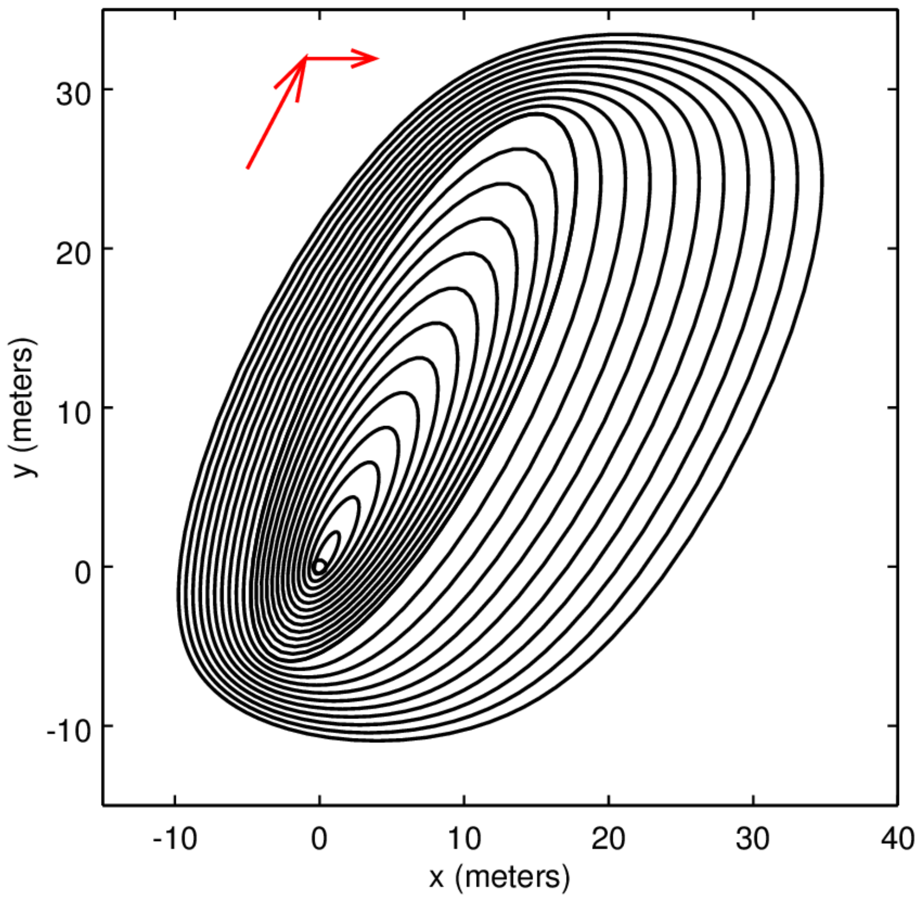


Figure B.2: Fire front at discrete time steps. Arrows indicate wind direction.

The dashed ellipse in Figure B.3 is the shape of the fire front at the time of the wind update. Though the level curves for this fire can be computed, the function cannot be inverted to tell the exact time of ignition for any point. Instead, the shape of the fire front is approximated by positioning 3 point ignitions in the center and extremities of the dashed ellipse, and by extruding the dashed ellipse into and against the wind. This gives a close approximation of the ignition time for the fire after a change in wind. For points inside the initial ellipse, the previous ignition pattern

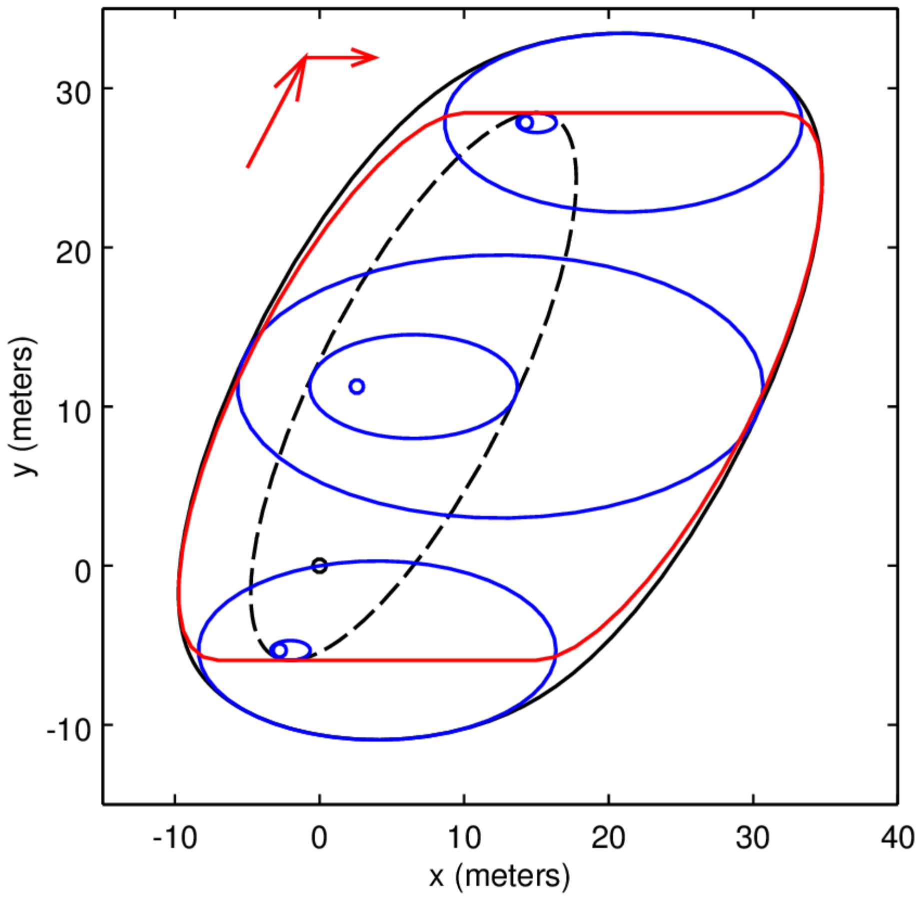


Figure B.3: Approximating the ignition time with 3 point ignitions (blue) and an extrusion (red). The dashed ellipse shows the fire front at the time the wind changed.

before the wind change is used to compute the ignition time.

The next time the wind changes, the fire front's shape would become even more complicated. Instead, we simplify the shape with an ellipse, as shown in Figure B.4, and continue simulation from that ellipse. This approximation is least valid when wind speeds are high and the wind direction changes almost 90 degrees. However as wind speed increases, the variability in its direction decreases [1], so this case should be rare during prescribed fires.

The linear fire fronts propagated from line segment fires are not as complicated. The length of the linear part of the front does not change, only the direction it moves.

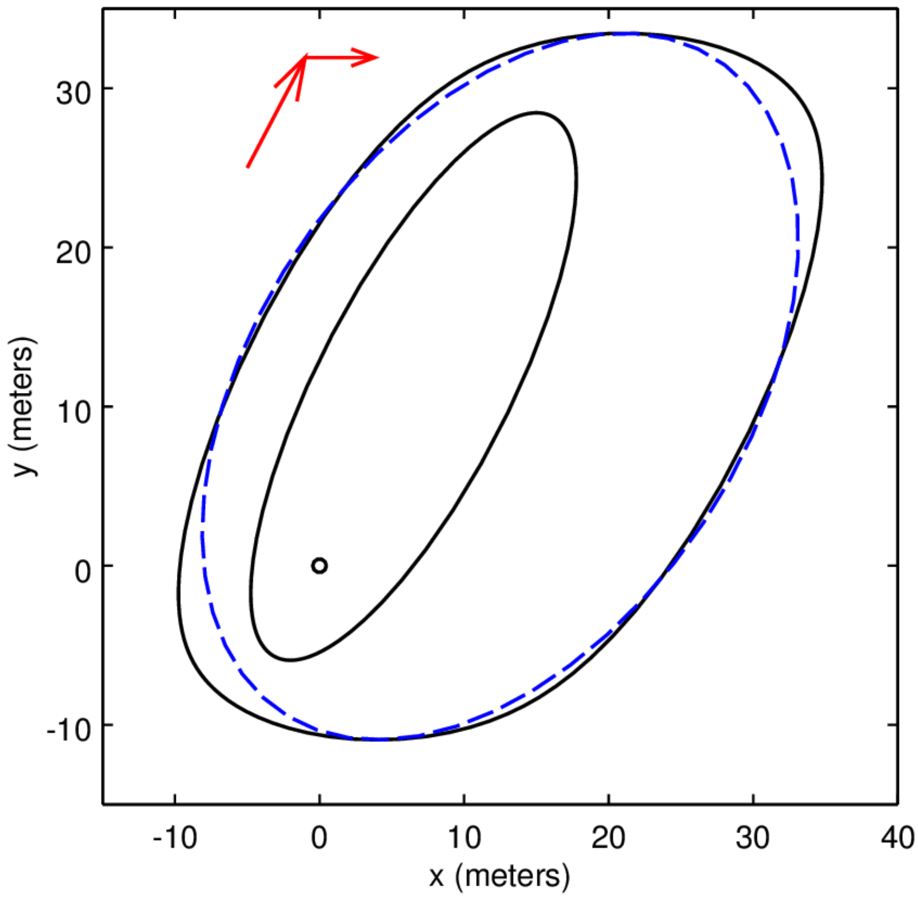


Figure B.4: Approximating the fire front as an ellipse during the next wind update

This can result in a small discontinuity due to the approximations used to update the elliptical fires around the endpoints of the line segment.

Appendix C

User Study Materials

C.1 Experiment Script

Hello, my name is Evan Beachly. Thank you for agreeing to participate in this study.

Please read and sign this consent form. Let me know if you have any questions about it.

Give participant consent form and pen.

Start recording on video camera after the consent form is signed. Close door.

I'm trying to keep this experiment as controlled as possible, so I'm not going to do much talking during this study, but feel free to ask me any questions. Please read this paper that describes the purpose of this study.

Give participant study purpose document

Next we want to get some information about you. Please fill out this questionnaire.

Give participant questionnaire

Please watch this instruction video. After the video you will have an opportunity to practice this.

Play instruction video

Next we'll do a practice session so you can try out some of these things.

Give participant tablet with practice scenario running. Do not play any video.

- Try moving the map around by pressing and dragging.
- Try rotating the map by placing 2 fingers on the touchscreen and rotating.
- Try resetting the rotation by clicking the compass in the top left.
- Try zooming in by touching the map in two locations and moving them further apart.
- Try zooming out by touching the map in two locations and moving them closer together.
- Adjust the map so that the UAV is centered and the red circles point upward, and the tablet is at a good zoom level.
- Try clicking the orange pen button to select the orange pen tool. Now click next to one of the fire fronts inside the perimeter to move the fire to that point. Try doing this a couple more times along the front.
- Try clicking the black pen button to select the black pen tool. Now click on some unburnt area to make it blackened.
- Try clicking the green pen button to select the green pen tool. Now click on some burnt area to make it unburnt.
- Try clicking the eraser button to select the eraser tool. Now click on some previous pen marks to erase them.
- Use the undo button to undo all of your actions.

- Deselecting your currently selected tool, and click on the fire to select that fire front.
- Deselect that fire front by clicking on the fire again, or clicking on some unburnt area.

Mute computer. Take tablet from participant. Back out of practice session on tablet. Start playing aerial video recording and restart the practice session on the tablet at the same time. Give tablet back to participant.

Now try matching this fire front to the one in the video.

Receive tablet from participant when they are done

Please read this document that describes the experiment procedure. Let me know if you have any questions.

Prepare first experiment session on tablet, and the corresponding aerial video recording

Now we'll conduct the experiment.

Prepare to play the first aerial video recording. Prepare the first session on the tablet. Start both at the same time and give to the participant. Wait until participant is done. Back out and repeat for the other four recordings.

Okay now my assistant will ask you some questions

Leave room and send in Alisha. Alisha asks the participant each interview question while they are recorded. Return once Alisha is done.

Please sign this receipt to receive your compensation.

Give participant receipt, pen, and 15 dollars

University of Nebraska Lincoln
Computer Science and Engineering Department

Participant Informed Consent Form

IRB# 17390

Title: NRI: Enabling UAS Fire Ignitions in Complex Firefighting Contexts

Purpose:
This research project will aim to evaluate a fire simulation that can be improved with human feedback. You must be 19 years of age or older to participate. You are invited to participate in this study because you are familiar with prescribed fires.

Procedures:
Before the experiment you will be asked to complete a questionnaire about yourself and your experience with prescribed fire, touchscreen devices, and unmanned aerial systems. The experiment will involve using a touchscreen tablet to adjust a fire simulation to match an actual prescribed fire videotaped by an unmanned aerial system. First you will be instructed on how to use the touchscreen, then you will be given a simple practice task. During the experiment you will watch 4-5 minute long videos of the prescribed fire, and will use the touchscreen to adjust the fire simulation. After the experiment you will be interviewed about your thoughts on the system. You will be videotaped during the experiment and interview. The procedures will last for about 1 hour, and will be conducted at/in room 211.1 of the Schorr Center at UNL.

Benefits:
There are no direct benefits to you as a research participant; however, the benefits to science and/or society may include intelligent unmanned aerial systems that can assist with prescribed fires for rangeland management.

Risks and/or Discomforts:
There are no known risks or discomforts associated with this research.

Confidentiality:
Any information obtained during this study which could identify you will be kept strictly confidential. Personally identifiable information will be stored in a locked cabinet in the investigator's office, and destroyed after 7 years. Video recordings will be transcribed and then destroyed within 1 month. Non-personally identifiable data will be stored electronically in private folders on secure UNL servers. These records will only be seen by the research team, and this data will be uploaded and accessed only using encrypted connections.

The information obtained in this study may be published in scientific journals or presented at scientific meetings but the data will be reported as group or summarized data. No identifiers linking you to this study will be published.

Compensation:
You will receive a $15 Amazon gift card for participating in this project.

Opportunity to Ask Questions:
You may ask any questions concerning this research and have those questions answered before agreeing to participate in or during the study. Or you may contact the investigator(s) at the phone numbers below. Please contact the University of Nebraska-Lincoln Institutional Review Board at (402) 472-6965 to voice concerns about the research or if you have any questions about your rights as a research participant.

Freedom to Withdraw:
Participation in this study is voluntary. You can refuse to participate or withdraw at any time without harming your relationship with the researchers or the University of Nebraska-Lincoln, or in any other way receive a penalty or loss of benefits to which you are otherwise entitled.

Consent, Right to Receive a Copy:
You are voluntarily making a decision whether or not to participate in this research study. Your signature certifies that you have decided to participate having read and understood the information presented. You will be given a copy of this consent form to keep.

Participant Feedback Survey:
The University of Nebraska-Lincoln wants to know about your research experience. This 14 question, multiple-choice survey is anonymous. This survey should be completed after your participation in this research. Please complete this optional online survey at: http://bit.ly/UNLresearchfeedback.

Participant Name:

____
(Name of Participant: Please print)

Participant Signature:

______________________________________ ___________________________
Signature of Research Participant Date

Name and Phone number of investigator(s)

Brittany Duncan, Ph.D, Principal Investigator	bduncan@cse.unl.edu	(402) 472-5073
Evan Beachly, Secondary Investigator	ebeachly@cse.unl.edu	(531) 333-0247

Study Purpose

Unmanned Aerial System for Prescribed Fires (UAS-Rx)

We've created an unmanned aerial system (UAS), also known as a drone, that can actually start fires at prescribed burns. This can be used by burn crews to help them ignite fire lines or ignite difficult-to-reach areas. However, we want to make our system more intelligent, so that it can stay safe and avoid fire, figure out what areas need to be ignited and which are already ignited, and alert personnel to danger. To do this we simulate the fire spreading over the terrain, but the simulation isn't perfect. The real fire might come across some terrain or vegetation that lets it spread faster or slower than predicted.

Fortunately, this unmanned aerial system is able to carry video cameras and other sensors that can observe and measure the fire. These observations can be used to correct the simulation so that it better matches the real fire. The simulation doesn't need to be extremely accurate, as its main use is to guide the UAS away from the fire, and find large unburnt regions to start fire lines in.

In this study, you'll be watching some aerial video we collected with our UAS at an actual prescribed fire, and then making corrections to our simulation of that fire. It's possible for the UAS to process the video autonomously, but it's important that we also allow the human operators to make these corrections in case the UAS misses something. This study is evaluating how accurate our simulation is with human input, and how easy to use our interface for doing that is.

Pre Questionnaire

Age: __________

Gender: (Check One)

☐ Male ☐ Female ☐ Other/Undisclosed

Are you right-handed or left-handed? (Check One)

☐ Right ☐ Left ☐ Ambidextrous

Occupation: ____

Education Level: (Check One)

☐ Some High School
☐ High School
☐ Some College
☐ College
☐ Some Graduate School
☐ Graduate School

Major: ____

Approximately how many prescribed fires have you been to? _________

What role(s) have you had at prescribed fires?

Approximately how large of an area was the largest prescribed fire you have been to?

Have you ever used a fire modelling/simulation program before? (Check One)

☐ Yes ☐ No

If yes, please list what programs you've used:

Approximately how frequently do you fly unmanned aerial systems (UAS) / quadcopters / drones / RC aircraft? (Check One)

☐ Daily	☐ Weekly	☐ Yearly
☐ Every Few Days	☐ Monthly	☐ Never

Approximately how frequently do you use a touchscreen device? (Check One)

☐ Daily	☐ Weekly	☐ Yearly
☐ Every Few Days	☐ Monthly	☐ Never

Approximately how frequently do you play video games? (Check One)

☐ Daily	☐ Weekly	☐ Yearly
☐ Every Few Days	☐ Monthly	☐ Never

Approximately how frequently do you use Google Maps? (Check One)

☐ Daily	☐ Weekly	☐ Yearly
☐ Every Few Days	☐ Monthly	☐ Never

Approximately how frequently do you use other online map programs? (Check One)

☐ Daily	☐ Weekly	☐ Yearly
☐ Every Few Days	☐ Monthly	☐ Never

If you use other map programs, please list them here:

Rate how much you agree with the following statements:
Observing prescribed fires is a good use for unmanned aerial systems (Check One)

☐ Strongly Disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Fighting fires is a good use for unmanned aerial systems (Check One)

☐ Strongly Disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Igniting prescribed fires is a good use for unmanned aerial systems (Check One)

☐ Strongly Disagree ☐ Disagree ☐ Neutral ☐ Agree ☐ Strongly Agree

Please use this space if you would like to give a more detailed explanation for any of your answers in this questionnaire:

C.5 Instruction Video Transcript

In this video I will show you how to use the interface for this fire simulator. During the experiment, you will be watching footage of a prescribed fire recorded by an unmanned aerial robot, and also using a touchscreen tablet. This video shows a screencapture of the touchscreen tablet.

When I click on a location on the touchscreen, a white circle will appear at that location, so you can see what I'm doing.

The main display on the screen shows satellite imagery from google maps. You can interact with this map using common touch gestures.

If I touch the map with 1 finger and drag, I can scroll the map around.

If I touch the map with 2 fingers and bring them closer together, I zoom out.

If I touch the map with 2 fingers and bring them further apart, I zoom in.

If I touch the map with 2 fingers and rotate them, I rotate the map.

If the map is not oriented with north up, a red and white compass appears in the top left of the screen. If I click on the compass, I reset the orientation to north.

Next I'll go over some of the other icons and buttons displayed on the screen.

In the top left is a scale bar that tells you how long that white bar is in feet.

Beneath that is a yellow arrow indicating the direction that the wind is blowing.

On the map is a purple polygon. This is the control perimeter established by the burn crew. The burn crew will keep the fire from spreading outside this perimeter.

Near the perimeter are some yellow lines. These are places that the perimeter burn crew has ignited. Fire will spread outward from these yellow lines, and it spreads fastest in the direction of the wind.

The blackend part of the map show locations that the simulated fire has already burned. The reddish-orange locations are the fire front, and are currently on fire.

Moving around on the screen is a white and red icon for the unmanned aerial robot that is video recording this fire. The two red circles indicate the front of the unmanned aerial robot, so you may want to rotate the map so that these are pointed upward, to align the map with the video it is recording.

On the right side of the screen are 4 buttons for different tools to adjust the fire simulation. To select a tool, just click on it. You can only have one tool selected at a time. To deselect a tool, click on it again.

You will most commonly use the orange pen tool. When this tool is selected, if you touch a point inside the burn perimeter, an orange circle will appear, and the fire simulation will adjust itself so that the fire front passes through that point. I can use this tool to move the fire front futher out, or further back.

If I touch a point far away from the fire, the simulation might not believe that that point could have ignited already, so it shows a disconnected blob of burnt area. If you want to tell the simulation that yes, this is actually where the fire front is now, you can use the black pen to mark blackened areas. If I put a black spot in between the two fires, it connects them.

The last pen is the green pen, which is used to indicate locations that are unburned.

In review, the black pen is used for marking places that are supposed to be black and burnt. The orange pen is used for marking places that are currently on fire. The green pen is used for marking places that are still green and haven't been burned yet.

Below the pens, is the eraser tool, which you can use to erase pen marks you've made. Select the eraser tool, then click on the pen marks you want to erase.

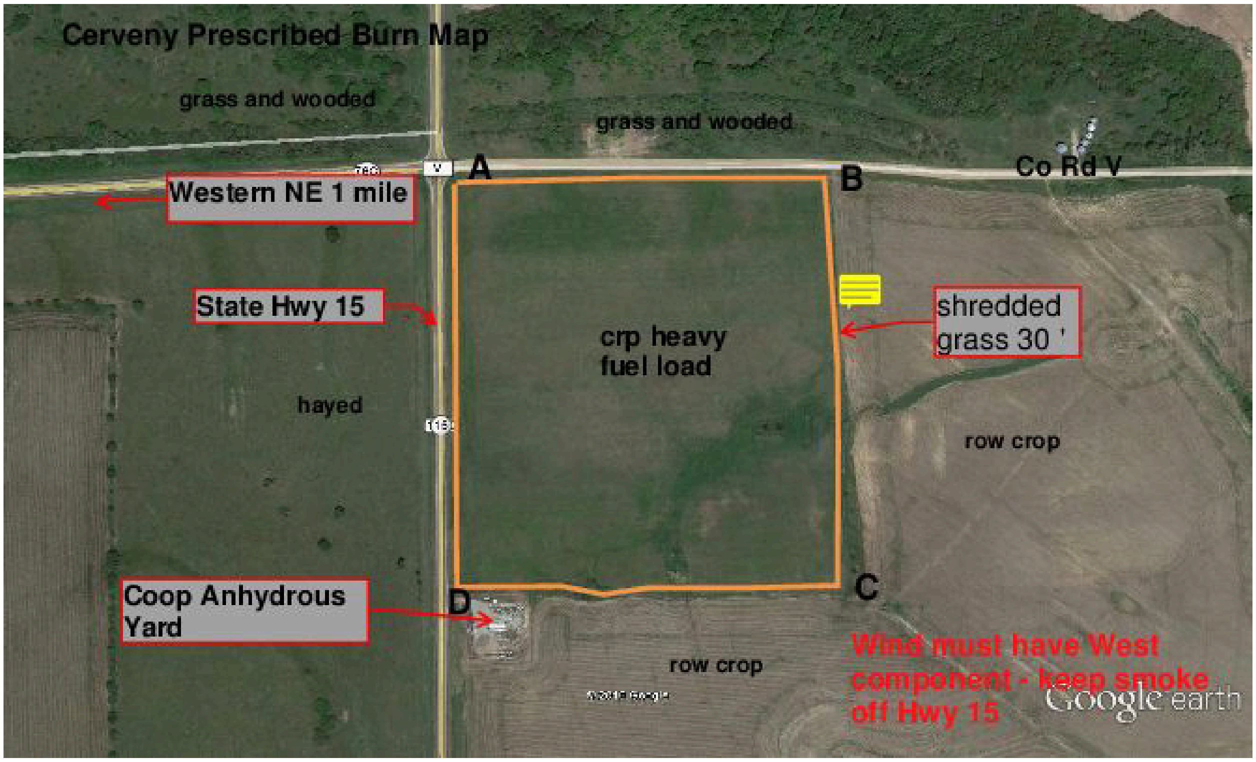
In the bottom right of the screen, there is also an undo button that can undo any pen marks or erases you make.

There are a few issues to be aware of. If you click approximately the same location twice rapidly, the map will zoom in on that location. This can interfere with you if you are trying to make several pen marks rapidly.

Finally, in certain cases the fire simulation may pull fire from a front you did not intend. If you deselect your currently selected tool, you can then click on a fire to select a particular front. Clicking off the fire will deselect it. When you have a front selected, further pen markings you make will only be applied to that front.

The number of marks you need to make is up to you, but there is a limit to how finely you can shape the fire front. If the points are too close together, they will average out. Remember, the fire simulation is going to be used to help the unmanned aerial system figure out where to start fire lines, so it doesn't need to be extremely accurate. It's more important to keep the main fire fronts in approximately the correct location.

Experiment Procedure



Burn Map

During the experiment, you will match the simulated fire to a different prescribed fire than what you saw during the practice. The above image shows the burn plan distributed to the burn crew before the fire. The top of the image is aligned with north, and the wind is forecasted to blow towards the northeast. The orange boundary marks the area to be burned (approximately 40 acres). The plan is to ignite the north-east corner (B), and have two crews starting fires along the north and east sides and work towards corners A and C. Then the crews will finish by igniting fires along the west and south sides and meet at corner D.

First, you will watch a short video taken by the UAS (unmanned aerial system) at the beginning of the fire in order to familiarize yourself with what the terrain looks like. In this video, the UAS will take off from the Coop Anhydrous Yard and fly towards corner B. Next, you will watch 5 3-minute long video clips taken by the UAS. During the video, use the touchscreen tablet to adjust the simulated fire to cover the areas that have already been burnt by the real fire, and the areas that the UAS needs to burn. The video clips are staggered throughout the duration of the burn. Adjustments you make during the video clips will carry over into the later video clips.

If you have difficulty using the interface or getting the simulated fire to match the real fire, don't give up, use the undo button or eraser to remove problematic adjustments. There will be an opportunity at the end to give feedback on issues you had so that we can improve the interface.

Post Interview Questions

How well did you think the fire simulation with your corrections matched the fire in the video?

How hard or easy was it to correct the simulation to match the simulated fire to the fire in the video? What made it feel hard or easy?

Did it feel like you were making a lot of corrections?

How easy or difficult was it to visually match what you're seeing in the video to a location on the tablet's map. What made it feel hard or easy?

What was something you liked about the interface?

If you could change something about the interface, what would it be?

Is there anything else you would like to say?

Bibliography

[1] ME Alexander et al. Estimating the length-to-breadth ratio of elliptical forest fire patterns. In *Proceedings of the eighth conference on fire and forest meteorology*, volume 29, pages 85–04, 1985.

[2] Elizabeth A Allen, Jeanne C Chambers, and Robert S Nowak. Effects of a spring prescribed burn on the soil seed bank in sagebrush steppe exhibiting pinyon-juniper expansion. *Western North American Naturalist*, 68(3):265–277, 2008.

[3] Loess Canyons Rangeland Alliance. Loess canyons rangeland alliance, 2016.

[4] VG Ambrosia, S Wegener, T Zajkowski, DV Sullivan, S Buechel, F Enomoto, B Lobitz, S Johan, J Brass, and E Hinkley. The ikhana unmanned airborne system (uas) western states fire imaging missions: From concept to reality (2006–2010). *Geocarto International*, 26(2):85–101, 2011.

[5] D. Anthony, E. Basha, J. Ostdiek, J.-P. Ore, and C. Detweiler. Surface classification for sensor deployment from uav landings. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[6] MJ Baeza, M De Luıs, J Raventós, and A Escarré. Factors influencing fire behaviour in shrublands of different stand ages and the implications for using pre-

scribed burning to reduce wildfire risk. *Journal of Environmental Management*, 65(2):199–208, 2002.

[7] Richard J Barney, Nonan V Noste, and Ralph A Wilson. *Rates of spread of wildfire in Alaskan fuels*, volume 311. Dept. of Agriculture, Forest Service, Pacific Nothwest Forest and Range Experiment Station, 1978.

[8] Evan Beachly, Carrick Detweiler, Sebastian Elbaum, Brittany Duncan, and Dirac Twidwell. Uas-rx interface for mission planning, fire tracking, fire ignition, and real-time updating. In *International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2017.

[9] Evan Beachly, Carrick Detweiler, Sebastian Elbaum, Brittany Duncan, Dirac Twidwell, and Craig Allen. Fire-aware automated aerial ignitions. 2018.

[10] Evan Beachly, James Higgins, Christian Laney, Sebastian Elbaum, Carrick Detweiler, Craig Allen, and Dirac Twidwell. A micro-uas to start prescribed fires. In *International Symposium on Experimental Robotics*, pages 12–24. Springer, 2016.

[11] Matthias M Boer, Rohan J Sadler, Roy S Wittkuhn, Lachlan McCaw, and Pauline F Grierson. Long-term impacts of prescribed burning on regional extent and incidence of wildfiresevidence from 50 years of active fire management in sw australian forests. *Forest Ecology and Management*, 259(1):132–142, 2009.

[12] D. W Casbeer, R. W Beard, T. W McLain, Sai-Ming Li, and R. K Mehra. Forest fire monitoring with multiple small UAVs. In *American Control Conference, 2005. Proceedings of the 2005*, volume 5, pages 3530–3535, June 2005.

[13] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 4, pages 3602–3608 Vol.4, April 2004.

[14] Joseph M DiTomaso, Matthew L Brooks, Edith B Allen, Ralph Minnich, Peter M Rice, and Guy B Kyser. Control of invasive weeds with prescribed burning 1. *Weed technology*, 20(2):535–548, 2006.

[15] Mark A Finney. Farsite: Fire area simulator-model development and evaluation. 2004.

[16] Mark A Finney, Isaac C Grenfell, Charles W McHugh, Robert C Seli, Diane Trethewey, Richard D Stratton, and Stuart Brittain. A method for ensemble wildland fire simulation. *Environmental Modeling & Assessment*, 16(2):153–167, 2011.

[17] Mark A Finney, Charles W McHugh, and Isaac C Grenfell. Stand-and landscape-level effects of prescribed burning on two arizona wildfires. *Canadian Journal of Forest Research*, 35(7):1714–1722, 2005.

[18] Michael Gollner, Arnaud Trouve, Ilkay Altintas, Jessica Block, Raymond de Callafon, Craig Clements, Anna Cortes, Evan Ellicott, Jean Baptiste Filippi, Mark Finney, et al. Towards data-driven operational wildfire spread modeling: A report of the nsf-funded wifire workshop. Technical report, 2015.

[19] James Higgins. Design, testing, and evaluation of robotic mechanisms and systems for environmental monitoring and interaction. Master's thesis, Department of Materials and Mechanical Engineering, University of Nebraska-Lincoln, 2016.

[20] A. Hodgson and N. P. Cheney. Aerial ignition for backburning. *Australian Forestry*, 33(4):268–274, 1969.

[21] Jon E. Keeley. Fire management impacts on invasive plants in the western united states. *Conservation Biology*, 20(2):375–384, 2006.

[22] Stefan Kohlbrecher, Alberto Romay, Alexander Stumpf, Anant Gupta, Oskar Von Stryk, Felipe Bacim, Doug A Bowman, Alex Goins, Ravi Balasubramanian, and David C Conner. Human-robot teaming for rescue missions: Team vigir's approach to the 2013 darpa robotics challenge trials. *Journal of Field Robotics*, 32(3):352–377, 2015.

[23] SEI Industries LTD. Premo fireballs, 2017. [Online; accessed 5-November-2017].

[24] Luis Merino, Fernando Caballero, J Ramiro Martínez-de Dios, Iván Maza, and Aníbal Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent & Robotic Systems*, 65(1-4):533–548, 2012.

[25] Luis Merino, José Ramiro Martínez-de Dios, and Aníbal Ollero. *Handbook of Unmanned Aerial Vehicles*, chapter Cooperative Unmanned Aerial Systems for Fire Detection, Monitoring, and Extinguishing, pages 2693–2722. Springer Netherlands, Dordrecht, 2015.

[26] Kristofer S Pister. Tracking vehicles with a uav-delivered sensor network, 2001.

[27] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.

[28] G Allen Rasmussen, Guy R McPherson, and Henry A Wright. Economic comparison of aerial and ground ignition for rangeland prescribed fires. *Journal of Range Management*, pages 413–415, 1988.

[29] Richard C Rothermel et al. A mathematical model for predicting fire spread in wildland fuels. 1972.

[30] Jean Scholtz. Theory and evaluation of human robot interactions. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003.

[31] Joe H Scott and Robert E Burgan. Standard fire behavior fuel models: a comprehensive set for use with rothermel's surface fire spread model. *The Bark Beetles, Fuels, and Fire Bibliography*, page 66, 2005.

[32] Ryan C. Skeele and Geoffrey A. Hollinger. Aerial Vehicle Path Planning for Monitoring Wildfire Frontiers. In David S. Wettergreen and Timothy D. Barfoot, editors, *Field and Service Robotics*, Springer Tracts in Advanced Robotics, pages 455–467. Springer International Publishing, 2016.

[33] Thane Somers and Geoffrey A. Hollinger. Human–robot planning and learning for marine data collection. *Autonomous Robots*, 40(7):1123–1137, Oct 2016.

[34] Jim F Stritzke and Terrence G Bidwell. Eastern redcedar and its control. *Weeds Today*, 15(3):7–8, 1984.

[35] Dirac Twidwell, Craig R Allen, Carrick Detweiler, James Higgins, Christian Laney, and Sebastian Elbaum. Smokey comes of age: unmanned aerial systems for fire management. *Frontiers in Ecology and the Environment*, 14(6):333–339, 2016.

[36] Dale Wade. Ignition devices for prescribed burning, March 2013.

[37] DD Wade and JD Lunsford. A guide for prescribed fire in southern forests. technical publication r8-tp 11. Technical report, United States Department of Agriculture, Forest Service Southern Region, 1989.