April 2011

SensorFly: Controlled-mobile Sensing Platform for Indoor Emergency Response Applications

SensorFly: Controlled-mobile Sensing Platform for Indoor Emergency Response Applications

Aveek Purohit, Zheng Sun, Frank Mokaya and Pei Zhang
Department of Electrical and Computer Engineering
Carnegie Mellon University
{aveek.purohit, zheng.sun, frank.mokaya, pei.zhang}@sv.cmu.edu

ABSTRACT

Indoor emergency response situations, such as urban fire, are characterized by dangerous constantly-changing operating environments with little access to situational information for first responders. In-situ information about the conditions, such as the extent and evolution of an indoor fire, can augment rescue efforts and reduce risk to emergency personnel. Static sensor networks that are pre-deployed or manually deployed have been proposed, but are less practical due to need for large infrastructure, lack of adaptivity and limited coverage. Controlled-mobility in sensor networks, i.e. the capability of nodes to move as per network needs can provide the desired autonomy to overcome these limitations.

In this paper, we present SensorFly, a controlled-mobile aerial sensor network platform for indoor emergency response application. The miniature, low-cost sensor platform has capabilities to self deploy, achieve 3-D sensing, and adapt to node and network disruptions in harsh environments. We describe hardware design trade-offs, the software architecture, and the implementation that enables limited-capability nodes to collectively achieve application goals. Through the indoor fire monitoring application scenario we validate that the platform can achieve coverage and sensing accuracy that matches or exceeds static sensor networks and provide higher adaptability and autonomy.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communications*; I.2 [**Artificial Intelligence**]: Robotics—*Autonomous vehicles, Sensors*

General Terms

Design, Experimentation, Measurement

Keywords

Mobile Sensor Networks, Aerial Networks, Hardware Design

1. INTRODUCTION

Indoor emergency response scenarios such as urban fire, earthquakes, gas leaks or hostage situations are characterized by dangerous and constantly changing operating environments for first responders. Rescue personnel have little prior information about the scene as well as adverse conditions such as smoke or structural collapse that may impede the planning and co-ordination efforts.

Deployments of wireless sensor networks have been proposed to improve the effectiveness of rescuers. Sensor networks can provide valuable fine-grained real-time environmental information. For example, the temperature profile within a building structure on fire can be used to predict the fire's propagation through building. Such information could potentially enable rescuers to anticipate the evolution of an emergency and adopt effective strategies to minimize injury, loss of life and damage to property. The very low-cost of sensor nodes has been envisioned to enable practical large-scale deployments in such circumstances and provide enough redundancy to survive these harsh environments.

Previously proposed sensor networks have largely been static in nature [16, 19, 35]. The sensor nodes either must be pre-installed as part of an infrastructure, or be deployed manually by first responders. As a result, these static node approaches require the following improvements for widespread adoption and utility:

- **Less infrastructure and maintenance**. In most approaches, a pre-installed sensing infrastructure is assumed to be in place. The cost of universally creating (placing of nodes) and maintaining (battery replacement) such infrastructure remains high.
- **Adaptability and robustness**. The harsh environment of emergency scenarios makes sensing nodes susceptible to damage and failure. A pre-installed static network infrastructure cannot effectively adapt to the destruction of parts of the network and requires high redundancy.
- **Adaptive spatial coverage**. With only a limited number of sensors, spatial coverage of sensed data is also restricted at deployment time. Repositioning or re-tasking nodes as per situation is not possible.

Recent literature has proposed the vision for how mobile sensors or *mobiscopes* can be used to monitor human spaces [5]. One envisioned idea is that of actuated or controlled mobility. Controlled mobility enables a network to mobilize its nodes to suit its demands. Such needs could involve tasks such as data gathering or maintaining network connectivity. Since the network can deploy autonomously, it can replace faulty nodes or reorganize as per its application requirements, addressing many limitations of static sensor networks. Moreover, considering that no universal sensing infrastructure must be installed and maintained, a much larger number of devices can be deployed economically.

Sensing nodes like these that combine mobility with low-cost large-scale deployments can provide effective solutions for information gathering in emergency response scenarios.

While robotic platforms [7, 12] exist, they are unsuitable for use as *mobile carriers* for sensor nodes in indoor emergency response scenarios. This is because robotic platforms suffer from the following limitations:

- Single or monolithic robot platforms do not provide the robustness and coverage of highly distributed sensor networks. Furthermore, most robot platforms require sensors such as laser range finders or GPS for navigation, making them considerably more expensive than traditional sensor nodes and uneconomical for large or expendable deployments.
- Secondly, most existing swarm robot platforms are ground based. Ground based robots do not allow for 3-D sensing as well as have limited reach. Existing micro-aerial vehicles [27] or networked unmanned aerial vehicles [6], apart from their high cost, have large form factors that is ill-suited to indoor operation and hazardous to human occupants.

An aerial controlled-mobile sensing platform therefore, provides a better alternative for indoor emergency response. Designing such a platform however, raises the unique challenge of realizing a low-cost, small form-factor device capable of autonomous flight and collaborative sensing. The low cost and small form factor requirements limit the capabilities of individual nodes, making traditional robotics approaches less applicable. We contend that such a lightweight platform must take a network-centric view to collectively realize complex application goals with its simple sensing, navigation and processing abilities.

1.1 Contributions

In this paper, we present SensorFly, a controlled-mobile aerial sensor network for monitoring applications in indoor emergency scenarios. The main contributions of the work are:

- We present a miniature(29g), low-cost($200), aerial sensor networking platform. To the best of our knowledge, it is significantly smaller than any other realized flying sensor network platform and its cost is comparable to that of low-cost traditional static nodes [21].
- We present and evaluate our hardware design choices as well as examine trade-offs that achieve a delicate balance between individual node capability and node resources for miniature aerial controlled-mobile platforms.
- Focusing on the indoor emergency fire monitoring application, we use the realistic NIST CFAST [30] fire simulation model, to perform a comparative evaluation of the effectiveness of our mobile aerial sensing platform over traditional static approaches. We show that our approach provides sensing coverage and adaptability, while reducing risk to firefighters or need for established infrastructure through deployment autonomy.

The rest of the paper is organized as follows. Section 2 introduces our target application and its requirements that motivate platform design. Section 3 examines our constraints and hardware design tradeoffs. Section 4 describes our software architecture. Section 5 provides the implementation and characterization of the platforms capabilities. Section 6 compares our work with static sensor networks. Section 7 discusses related work. Section 8 presents a discussion of aspects for further study. Section 9 summarizes our work.

2. APPLICATION

The SensorFly system can be deployed in several sensing and monitoring applications such as survivor search after earthquakes, reconnaissance in urban combat, or indoor toxic plume sensing. We present on the indoor fire emergency monitoring application to evaluate the effectiveness of our platform in providing autonomous, timely, and high fidelity information to aid fire-fighter operations.

Fire response and fire rescue remain extremely challenging operations that annually claim the lives of over 100 firefighters in the United States [4]. Lack of situational information has been identified as a critical limitation for firefighters [16]. On average, firefighters reach sixty-one percent of urban fires in six minutes. However, the fire may spread extensively within that period. Firefighters have little or no knowledge of the location, extent, or advance of the fire in these situations.

Requirements

Sensors for real-time sensing and prediction of fire propagation are an active area of research. Researchers have proposed fire models to predict the advance of fire from in-situ sensor readings.

This information can be valuable for several fire fighting tasks:

- Firefighters can determine the extent of fire and predict its progression.
- Firefighters become aware of hazardous areas to avoid.
- Firefighters can effectively plan evacuation routes.

The fire model determines the type and spatial location of sensed data. A popular model is *Consolidated Model of Fire and Smoke Transport* – CFAST [30]. CFAST is a zone model where each space is split into the top zone (consists of the high temperature gases and smoke) and the bottom zone (consists of lower temperature gases). The height of the interface between the two zones is called the smoke layer height and this layer descends as smoke builds up in the room. Structural information such as the presence of shafts or open stairways is also of interest to firefighters.

Fire monitoring sensor networks seek to provide a 3-D profile of parameters such as temperature, gas, pressure, and ceiling height in each room as inputs to such a model.

While many sensor networking systems have been proposed for fire monitoring, they are essentially composed of pre-deployed static sensor nodes [1, 16, 20, 35]. We present a detailed comparison with related work in Section 7. Such infrastructure is expensive to deploy and universal adoption remains far into the future. SensorFly nodes can be deployed at the time and location of fire. While, unlike static pre-deployed sensors, current version of SensorFly nodes cannot know their exact location and provide data from regions obstructed due to closed doorways or building structure collapse, the SensorFly system can provide valuable information where a static infrastructure is absent. Firefighters can introduce SensorFly nodes into connected spaces as they enter the building structure, without the need to physically explore every region.

Moreover, a larger number of SensorFly nodes can be deployed at the actual point of emergency as opposed to maintaining universal infrastructures. Finally, the aerial mobility allows better spatial resolution of sensing. The fire-monitoring scenario requires 3-D sensing of vertical temperature profile and ceiling height in building structures. The SensorFly with its ability to hover vertically can provide higher fidelity data.

3. HARDWARE DESIGN

In order to demonstrate the feasibility of the SensorFly system, we have designed and built three generations of SensorFly nodes (Figure 1). This section focuses on the

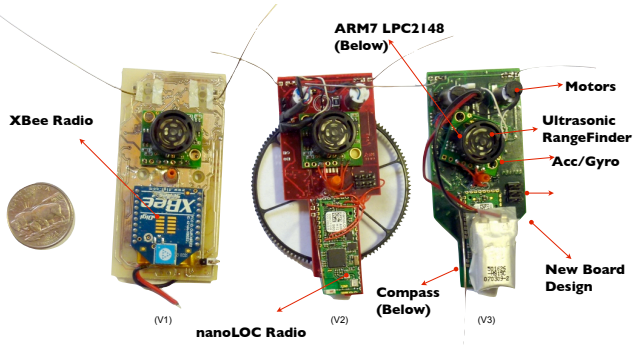


Figure 1: Three generations of SensorFly node design.

Table 1: Weight of several possible components of SensorFly nodes. X's mark the components included in the base configuration of SensorFly nodes.

	Component	Weight
X	Drive Motors and Propeller Assembly	15 grams
X	130mAh Lithium Polymer Battery	4 grams
	200mAh Lithium Polymer Battery	7 grams
X	Controller Board	10 grams
	Camera Board Add-on	3 grams
	Audio Board Add-on	4 grams
	LED Board Add-on	3 grams
	Ultrasonic Distance Sensor Add-on	4 grams
	Basic Sensorfly Total Weight	29 grams
	Absolute Maximum Takeoff Weight	34 grams

hardware design choices and the trade-offs involved in building controlled-mobile aerial sensing platforms.

3.1 Key Constraints

The low-cost, low-weight aerial sensing platform presents several constraints and challenges, occupying a new and unique design space. The addition of mobility and control introduces new constraints like *weight*, *sensor interference*, and *higher noise*, to the traditional low-cost COTS(commercial off-the-shelf) based sensor node hardware architectures. Careful consideration is required in new aspects of sensor network hardware design such as *component placement* and *weight balance*. To achieve the desired level of collective system capability given the minimum available resources of individual nodes, a delicate balance must be attained and trade-offs examined. The following factors affect our design approach:

Cost. Low per-device cost allows us to scale up senor node deployments. As a result, for the equivalent cost of an intelligent robot, many more sensor nodes can be used, enabling higher sensing coverage as well as discovery speed. Utilizing a low-cost flight mechanism, common to off-the-shelf RC helicopters, allows us to achieve a prototype cost of about $200. In mass production, similar RC helicopter assemblies are commercially available for about $20 [32]. While larger flying platforms such as the Parrot.AR Drone [29] can provide better capability, they have higher production cost (∼$300) and lower reach due to the bigger form-factor. The navigational capability of individual sensor nodes must be attained through low-cost COTS sensors. A trade-off must be made in forgoing accuracy for deployment scale to better realize our application objectives.

Weight. The miniature aerial platform adds a new metric, weight. The small weight enables longer flight times, greater reach, and better safety for indoor emergency response scenarios. The weight limit is decided by delicate trade-off point. Adding more weight requires bigger motors that in turn require a bigger battery. A larger craft eventually sacrifices the miniature form factor along with the mobility and scalability advantages that it provides. Table 1, shows the component-wise weight break-up of the 29 gram SensorFly node.

Table 2: SensorFly node's operation modes and their power usage breakdown.

Operation	Typical Power Usage
Mobile Mode	6.2W
Stationary Mode	
Data transmission	310mW
Data receive	330mW
Sensing only	225mW
Processing Only Mode	150mW
Idle Mode	1mW

This weight constraint limits the number of sensors that can be carried. In addition, the weight must be balanced to achieve stability of the node in flight requiring careful component layout and board design.

Energy. Similar to many battery-operated systems, the SensorFly platform is highly energy constrained. Unlike most other sensor systems, however, SensorFly has many different operating modes with vastly different energy characteristics. Table 2 shows the energy consumption characteristics for some of these modes. Of all these modes, the ones involving flight are the most expensive in terms of energy usage.

This further underscores the need for a lighter node, as it enables us to reduce the power consumption of motors. The current battery and weight profile provides about five minutes of airborne flying time. However, by optimizing movements for deployment and reconfiguration, and managing energy consumption when landed and sensing, the overall life of the network can be extended. Utilizing physical characteristics such as the *ground effect* can further reduce movement energy. We evaluate the flight time of the SensorFly nodes in Section 5.4.

Interference and Noise. The small form-factor requires placing sensors and components very close to each other on a miniature circuit board. At the same time, use of low-cost brushed motors creates large electromagnetic noise that interferes with the proper operation of the sensors. The placement of sensors and components must be done keeping in mind the effect and nature of induced noise. Additionally, software-filtering approaches are also be needed to obtain better sensor readings.

3.2 Design Trade-offs

The resource constraints and capability requirements force our component selections for the SensorFly platform. In this section, we examine our design choices and trade-offs. We also discuss the evolution of the current third generation hardware platform which incorporates learning from previous iterations.

Processor. The processor used in the SensorFly node is the ARM7 based LPC2148. The micro-controller is capable of running at 60 MHz and features 512 KB of Flash memory as well as 42 KB of RAM. It is limited in comparison to the computers that are currently used for most robotics applications. Through various iterations, we have found the processor to be capable of running both the flight control algorithms and the sensor filtering algorithms. The second and third versions of SensorFly nodes incorporate a secondary external processor, an AVR AtMega644, for radio functions and control. By moving the majority of the time critical processes off-board, concurrency is handled better and application integration is simplified. This significantly reduces conflict between time critical components such as the radio and the flight controller.

Table 3: Comparison of Navigational Sensors.

Component	Cost	Weight	Accuracy
Accelerometer	Low cost COTS component	Low < 1g	Analog inertial sensor. Unreliable for distance estimation due to accumulating error. Used to detect collisions.
Gyroscope	Low cost COTS component	Low < 1g	Useful for angular velocity measurement. Unreliable for absolute angular position measurement but not affected by magnetic fields. Used for feedback to for yaw controller.
Compass	Low	Low < 1g	Low indoors due to sensitivity to magnetic fields. Error does not accumulate. Used to provide absolute heading.
Ultrasonic Ranger	Low	Medium ∼4g	Fair. Depends on environmental factors such as interference and materials.
Nanotron nanoLoc RToF ranging	Low. Cost is amortized as radio is also used for communication.	Medium. (∼4g)	Better accuracy than RSSI based radio-ranging. Less accurate than ultrasound and laser range finders.
Laser Ranger	Medium	High 50g+	High. Not included due to weight constraints.
Vision	High	High	Accuracy depends on operation scenarios. Less effective in presence of smoke. Needs high processing power.

Navigation Sensors. The choice of sensors requires careful consideration, due to the limit on their size and numbers. Navigational sensors have been explored extensively for robotic platforms [10]. Navigational sensors are needed to detect the motion of the node itself as well as sense the environment or other nodes. For motion estimation, miniature MEMS-based inertial sensors such as the accelerometer and gyroscope have become popular in consumer devices like mobile phones, and as a result are commercially available at low-cost. However, their susceptibility to noise is higher and a trade-off must be made against accuracy. For navigating the environment, a number of higher accuracy range based options such as laser range finders, multiple-ultrasound sensors, camera, and lidar sensors are unsuitable for use in SensorFly. Table 3 summarizes the strengths and weaknesses of available sensors on the basis of cost, weight and accuracy. Radio based RF-ranging is an attractive technique, especially because the radio can also be used for communication. However, multipath effects limit the accuracy of radio ranging in indoor environments. This limits precise navigation and movement and calls for collective stochastic exploration approaches. We examine trade-offs further in Section 5. Furthermore, these sensors are re-used for multiple purposes as described in Section 3.3.

In the first version of SensorFly, a two-dimensional compass and a three-dimensional accelerometer were included as flight state sensors. Since the craft is passively stable, the five-degree-of-freedom measurement should be enough to capture the full possible motion of the node. However, the magnetic interference from the motors is large due to the nodes small physical size. This interference during flight renders the readings from the compass inconsistent and unsuitable for measuring rotation. To improve the flight controls, the later versions include a 3-D compass, a 3-axis accelerometer, as well as a 2-D gyro. In addition, the placement and physical design of the boards mitigate the noise characteristics as described later. These sensors provide a full eight degree-of-freedom measurement. During flight, the gyros provide a rotational sensor immune from magnetic noise, while the compass provides an absolute reading.

Radio. To aid navigational needs, the current version of SensorFly uses the nanoLOC TRX transceiver module [26]. Apart from offering better performance against indoor multi-path fading effects, the radio provides inter-node range estimates based on round-trip time of flight (RToF) computations.

A Digi XBee [9] was used for the V1 of the hardware. This radio is commonly used in sensor networks. Received signal strength indicator (RSSI) was used for range estimation in V1. Several factors prevented the success of this approach. First, in the indoor environment, the radio characteristics were extremely unpredictable, making the multi-path problem pronounced. Second, electromagnetic noise from the motor significantly increased the unpredictability of RSSI measurements. Finally, due to the physical orientations of the helicopter, the antenna cannot be placed at an omni-directional location. This increased the effect of node orientation on the RSSI. These factors prevented the use of RSSI as a viable solution for use in mobile sensor nodes.

Motors. A unique feature of the SensorFly nodes is the mechanical helicopter drives. We use two 7mm core-less motors. These motors drive two coaxial main rotors, and are a significant source of noise in the system, as we shall explore later. While brush-less motors would provide better noise and thrust performance, their size, cost, and circuitry requirements make them ill-suited for our target application.

Furthermore, the low-cost of these core-less motors implies substantial variations in their response to input voltages as well as degradation in performance with use. However, due to the coaxial helicopter design and its passive stability, simple control algorithms can be used to counter-effect these variations. A third motor can be added to the node to provide controlled forward flight. Currently, a weight difference, created by placement of components on the board, is used to provide a constant forward motion, while the craft rotates in small circles to hover in place.

3.3 Component Reuse

An important strategy for achieving the platform's stringent weight goals is component reuse. Several elements of the node's mechanical and electronic components are selected to be capable of perform more than one function.

On the *electronic hardware* side, the Nanotron nanoLoc radio module enables communication as well as Round-trip Time-of-Flight radio ranging. This ranging capability provides a primitive form of localization and is a substitute for having laser or ultrasound range finders. Although, a trade-off is made in the accuracy of range estimation, the weight and cost constraints are impossible to meet with the alternative ranging mechanisms mentioned.

Sensors such as the accelerometer are used as obstacle sensors, for their ability to detect contact. The lightweight and robust design of the SensorFly nodes enables them to tolerate bumping into obstacles without affecting their flight performance. Thus, dedicated obstacle sensors like infrared or ultrasound based detectors are avoided.

Amongst the **mechanical components**, the blades on the helicopter design serve to protect the body of the node from bumps. As the body is designed to smaller than the blades, it acts as a protective buffer for the on-board

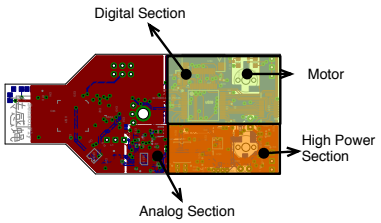


Figure 2: The SensorFly node layout. Analog sensors are isolated and the compass placed away from the motors to reduce interference. The digital, analog, and high power sections have separate power supply and the ground plane is interconnected through ferrite beads to filter noise.

electronics.

The circuit board housing the SensorFly electronics, itself acts as the fuselage for the node. This requires careful selection of the board to provide enough rigidity and in turn, prevent stress on the connections and traces when the node lands. At the same time, a thick board adds more weight to the node. In V1 of the design, a 20-mill double layer board is used. While the design is a 1.6x3.1 inches square, the size reduced airflow thereby degrading the lift of the nodes. A 20-mill, 4-layer circuit board shaped as a 'T' was subsequently used to allow more air to flow through from the blades. Due to the reduced per-layer thickness nodes experienced higher failure rate due to stress on the metal traces caused by takeoff and landings. In the third version of SensorFly, a 30-mill board was selected, stress relief added to the 'T' shape and component placement staggered to further reduce single stress points. This redesign greatly improved the fuselage strength.

The legs of the SensorFly node are built with gold plated spring wire and are used as charging terminals as well as landing supports.

Each node can have different weight distributions due to modular design of the SensorFly nodes. For example, different sensors can be added to the extension port of the SensorFly. To counter this effect on the stability of the craft, the battery and the ultrasonic sensor act as adjustable counterweights and are positioned so as achieve the desired node balance.

3.4 Board Layout

The SensorFly board layout involves careful consideration of the noise characteristics of the components and their weight. The analog sensor components such as the accelerometer and gyroscope must be placed so that they are isolated from the noise sources. The main source of noise in the platform is the high power source and the pair of brushed motors causing high electromagnetic interference. Similarly, the compass must be isolated from the motors as it is adversely affected by their rotation magnetic field.

Figure 2 shows the layout of the SensorFly board. The board is divided into 3 sections. The motors occupy the right (front) end, along with digital section, since the digital components are least affected by the EMI. The 3-D compass is placed towards the tail of the node to minimize the magnetic effect of the motors. The left-most section is the analog section which houses sensors such as the gyro and accelerometer that are adversely affected by noise in the power-lines due to the motors back-emf. The high power section that feeds the motors occupies the bottom-right corner of the board. Each section has separate power lines and a separate ground plane. These are interconnected through ferrite beads to filter out noise.

Another consideration in the placement of sensors is their weight. The board weight must be balanced for stability

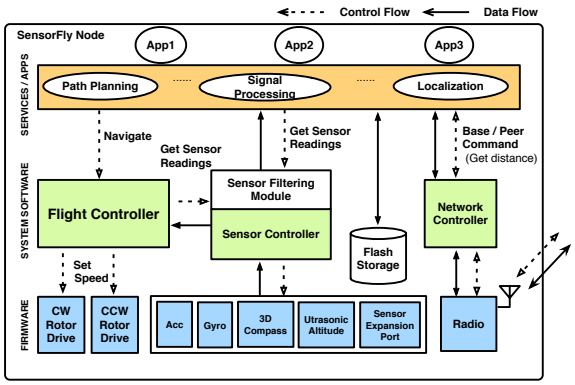


Figure 3: SensorFly Node Architecture

of flight. A slightly larger weight is maintained towards the front of the node, to substitute the tail-blade of the helicopter design, and enable forward movement. The 4g battery and 4g ultrasound height sensor are used as adjustable counterweights to balance the weight of the node. The battery placed towards the tail of the node while the ultrasound sensor is placed in the middle to counteract the weight of the motors in the front.

3.5 Extensibility

Several types of sensors can be added to each node using the provided expansion ports, in accordance to the needs of different applications. The expansion port supports serial, SPI, 10-bit parallel, and I2C for sensor interconnection, as well as provides regulated and unregulated power pin through the node battery. Due to weight constraints, we plan to only include one additional sensor module per craft. Several sensors have been designed, including the camera, speaker, microphone, and infrared detectors. We plan to explore the use of additional sensors as our research progresses.

4. SOFTWARE ARCHITECTURE

The SensorFly is designed to provide a platform for controlled-mobile and collaborative sensing for emergency response. The SensorFly system architecture, shown in Figure 3, consists of the firmware, the node-level system software, customizable network level services, and user application layers. The node system software consists of three major modules corresponding to the capabilities of the platform,

- **The sensor controller** provides access to on-board sensors and expansion ports. Includes filtering modules to mitigate noise caused by motors and motion.
- **The network controller** provides peer-to-peer aggregation and broadcast communication, with support for inter-node range estimation through RToF.
- **The flight controller** provides a high-level navigation API for hover, turn and single-direction flight. A biased random-walk dispersion and exploration algorithm is implemented utilizing the node ranging capability. The algorithm enables nodes to navigate and deploy in unknown environments without need for localization.

4.1 Sensor Controller

The sensor controller provides access to the on-board sensors that include the ultrasonic altitude sensor, 3-axis accelerometer, 2-axis gyroscope, and a 3-D electronic compass, as well as to the sensor expansion port. The module provides

an API for querying sensors, setting repetitive sample rates, as well as provides in-built filters for noise reduction.

The motors and miniature form factor of the SensorFly nodes affect the performance of sensors such as the compass, as described in Section 3. Moreover, some sensors have inherent noise characteristics that can be filtered with knowledge about the dynamics and mobility of the node. Thus, sensor filtering must be performed for achieving useful capabilities such as altitude control and pose estimation.

The ultrasonic range finder used to measure the altitude of the SensorFly node, is affected by several environmental factors such as absorption characteristics of the ground and interference from other sources such as fluorescent lamps. The sensor controller utilizes the vertical motion dynamics of our platform to discard erroneous readings, using a first order recursive digital filter, as described in Section 5.

Similarly, the SensorFly has a 3-axis electronic compass [14] for direction sensing. This is useful in estimating the pose of the node. However, the small dimensions of the SensorFly node require the compass to be placed close permanent magnet DC motors that distort compass reading. Analysis of error induced by the motor's moving magnetic field, points to a symmetric distribution which can be filtered out to a large extent through a moving average filter implemented within the sensor controller module. The window size and other filter parameters are tuned through empirical analysis of sensor data, as detailed in Section 5.

The sensor controller also provides a virtual sensor for detecting obstacles using an accelerometer. An algorithm based on thresholds is able to distinguish bump events from the acceleration signal vector magnitude from normal flight.

4.2 Network Controller

Our networking implementation supports two major capabilities namely peer-to-peer data communication and radio based inter-node ranging. The SensorFly has a dedicated AVR AtMega644 microcontroller for radio control. This enables better handling of packet transmit-receive and ranging operations that require timely processing. Especially, since flight control is the highest priority task on the primary microprocessor. The radio module, i.e. the nanoLOC transceiver and the AVR micro-controller, are connected via UART to the primary ARM7 LPC2148 microprocessor. The network controller implements the UART communication protocol, message queues for inbound and outbound packets, and provides a high level API for sending, receiving and forwarding data.

4.2.1 Data Communication

The network protocol for the mobile SensorFly nodes essentially consists of an aggregate and broadcast communication model. The monitoring network seeks to route all sensed data to the base station. Additionally, due to the constant motion of nodes, establishing routes and running explicit node discovery service is impractical. Nodes therefore periodically broadcast messages containing their sensor data. Neighboring nodes, on hearing the broadcast message aggregate the node's sensor data with their message.

Each node's sensor data consists of a sequence identifier and a time-to-live field. The time-to-live is decremented with the number of hops as well as on the expiration of a local time window, to control the time for which stale data propagates in the network. A node's data is propagated by other nodes only if the time-to-live is still not zero. Old sensor data from a node is replaced with fresh data, if it is received before the expiration of the time to live field. This scheme is akin to a controlled reverse-flood of data to the base station.

4.2.2 Ranging

The network controller also provides an API for node-to-node range estimation. The range estimates are used in the exploration algorithm currently employed by SensorFly nodes, which consists of biased-random walks [23] that disperse nodes away from each other based on their distance from each other. Inter-node ranging is a primitive used by topology estimation schemes.

The radio has the capability to compute distance using a round-trip time-of-flight (RToF) based technique called Symmetric Double Sided Two Way Ranging, or SDS-TWR [25]. The round-trip time-of-flight method measures the elapsed time between the host node sending a data signal to the remote node and receiving an acknowledgement from it. Using the estimated speed of propagation of a typical signal through a medium and the signal turnaround time, i.e. the time for the remote node to send out an acknowledgement packet, the host computes the distance from the remote node. Using physical layer timestamps and hardware-generated acknowledgements, the nanoLOC TRX radio achieves a predictable turnaround time.

Unlike other time-of-flight methods, this method does not require tight clock synchronization between nodes. The time elapsed is computed from timestamps of individual nodes themselves. This removes the need for extra hardware for global time synchronization, which is the source of complexity and higher cost in other systems. Likewise, no special antenna arrays are required such as angle-of-arrival ranging methods. We perform an experimental evaluation of the ranging performance in Section 5.2

4.3 Flight Controller

A SensorFly's miniature helicopter flying mechanism has many advantages like the ability to takeoff, land and turn in confined indoor spaces, maximizing sensing coverage. Realizing and controlling a helicopter-based sensor-networking platform presents many interesting aspects.

On one hand, the helicopter has highly coupled dynamics. Prior autonomous helicopters [15] have required more accurate feedback sensors and computationally expensive algorithms for precise control. On the other hand, the sensor-networking platform has a single CPU with limited computation (60MHz) and memory (42Kb) resources that must perform sensing, control and network processing tasks. Besides, as described before, the performance of control strategies is limited by sensor noise, attributed to node form factor, and cost constraints.

The SensorFly overcomes these challenges by using a lightweight damage-resistant node design and by sacrificing precise control and navigation. The SensorFly system is designed to approach tasks as a networked group that achieves system-wide objectives while tolerating errors in individual node motion. The robust design ensures that nodes can collide with obstacles and still be able to fly, removing the need for precise obstacle avoidance. In fact, the nodes detect obstacles through contact. This allows the flight controller component to implement computationally inexpensive proportional-integral-derivative (PID) control loops that provide *good enough* stability and utilize a biased-random walk approach to navigation.

The flight controller provides a high-level navigation API with commands for hovering, turning, and moving forward. The following sections briefly describe the node dynamics, the navigation and exploration approach used, and the control algorithms.

4.3.1 Exploration and Navigation

The fire-monitoring scenario required a group of resource

constrained SensorFly nodes to explore an unknown environment. No assumptions can be made regarding the availability of localization beacons or infrastructure in the indoor environments, for nodes to estimate their location. Detailed and updated maps of building structures may not always be available to the firefighters. Moreover, the harsh environments with smoke and fire make sophisticated vision based navigation capabilities ineffective. Thus, SensorFly nodes utilize their low cost and large numbers to spread out and maximize coverage through a biased-random walk based exploration algorithm, similar to that presented and analyzed by Morlok et al. [23].

The SensorFly nodes hop, i.e. takeoff, fly for certain duration, and land in the operating arena. The direction of hop is chosen randomly. For increasing coverage, the random walk is biased towards moving away from other SensorFly nodes while still maintaining connectivity. When a SensorFly node lands, it estimates its distance from other nodes in its vicinity. If a node exists within a *coverage range*, the node resumes its random motion. If no node exists in the *coverage range*, the node deploys and continues to sense data. In addition, if a node finds no other nodes within a *connectivity range*, it resumes it's random motion. This is to prevent networks from being partitioned. Nodes can hover at desired heights as per application sensing needs. The evaluation of this approach for fire monitoring is provided in Section 6.

The above exploration algorithm requires the capability of nodes to takeoff, control their altitude, follow a random path, and land safely. The SensorFly node design and design choices enable these capabilities to be attained in a computationally efficient fashion.

4.3.2 Altitude and Yaw Control

The SensorFly uses a kind of co-axial counter-rotating dual rotor design, which is passively stable for hover and forward flight [24]. This reduces the number of sensors and computing power required to stabilize it. In addition, this configuration and SensorFly's low weight allow the rotors to operate at relatively low RPM compared to conventional rotors, making them safer for indoor operation.

The control of the coaxial-helicopter based platform is simple compared to other helicopter design designs, with altitude and yaw being the controllable entities. A constant weight bias towards the front of the node enables it to move in the forward, when the yaw is held constant. The main features of controlling the node are:

- Altitude control is attained by controlling the speed of the two main rotors of the node.
- Yaw control, i.e. turning the helicopter from side to side is achieved by increasing the speed of one rotor and reducing the speed of the other rotor by the same amount.
- Forward flight of the helicopter is attained by placing the center of gravity towards the front of the aircraft. The main rotors follow the tilting of the node body and pull the helicopter forward. To hover, the helicopter rotates in small circles.

This flight controller provides SensorFly nodes with the capability to takeoff and maintains altitude, through a PID controller designed from first principle dynamic models of the node and empirical tuning. Similarly, another PID control loop is implemented for controlling the spin of the node and maintaining pose to achieve forward flight.

5. HARDWARE CHARACTERIZATION

This section describes the implementation and characterizes the distinctive characteristics of the SensorFly platform, in context of the fire monitoring application. Specifically, the altitude sensing, pose estimation, ranging and flight performance detailed and evaluated. Figure 4 shows the SensorFly node hovering using the height sensor and algorithms described bellow.

Figure 4: The 29g SensorFly node hovering in a hallway.

5.1 Height and Orientation Estimation

A 3-axis compass [14] with a maximum sampling rate of 10Hz is used to estimate the pose of the node and estimating yaw. We use the compass to offset the accumulating errors in gyroscope yaw measurements. However, compass readings are affected due its close proximity to motors. A moving average filter is used to minimize the distortion due to the periodic rotating magnetic field of the motor.

In addition to the orientation sensors, the SensorFly V3 nodes use a LV-MaxSonar-EZ1 [22] ultrasonic range finder mounted below the node fuselage, to measure the node's altitude from ground (for annotating sensed data) and provide feedback for altitude control. This is due to the inaccurate distance measurement of the radio described in the following section. The ultrasonic sensor weighs about 4 grams and has a narrow beam width that provides relatively stable readings. However, it is sensitive to building materials and interference from other sources. As described in Section 4, a recursive first order digital filter is used to reduce the effect of noise from the observations.

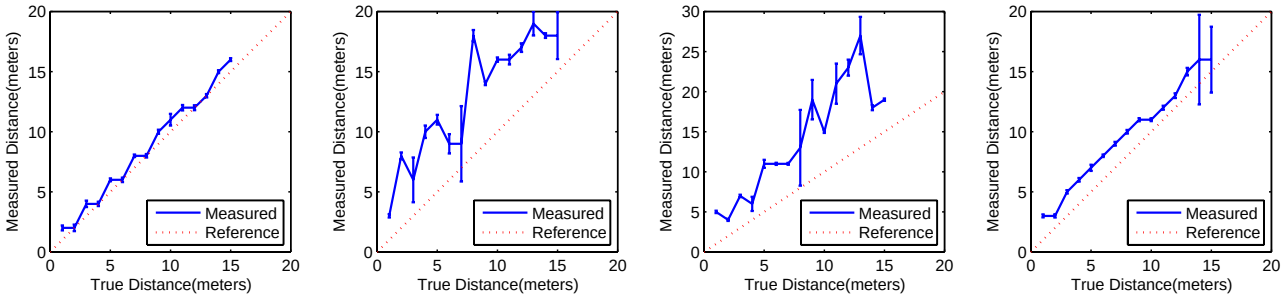
5.2 Ranging

Radio ranging enables us to attain navigation capabilities, while at the same time, meet our weight and cost constraints. However, our indoor and mobile operating environments introduce multi-path and Non-Line-of-Sight (NLoS) errors. These errors are a feature of the specific space configuration near the node's location and a general model cannot be assumed. Thus, to characterize the accuracy of ranging obtainable, we evaluate the SensorFly node radio ranging in typical operating scenarios.

We performed range measurement tests at four different locations, three indoor, and one outdoors. The indoor locations included a metallic cubicle area, a hallway, and a classroom with furnishings representative of multi-path rich RF propagation environment. An empty parking lot was selected for the outdoor test, to minimize the effect of reflections. At each location, measurements were made for inter-node distances of 1m to 15m, taking 100 readings for each distance.

Outdoor tests illustrated in Figure 5(a), show the baseline measurement to be consistently within 1m, with an average of 0.6m. Figure 5 also shows indoor measurements for three separate locations. Indoor tests exhibit a much larger error in measured range. Moreover, the errors are not consistent across locations. While a large university hallway, shown in Figure 5(d), presented a largely linear relationship between distance and RToF measurements, more constrained cubicle floors and corridors, illustrated in Figure 5(b) and 5(c), show higher variations. The average error for our test setup was around 4.2m from the true value.

Overall, our experiments indicate that RToF measurements cannot be utilized directly to obtain accurate indoor



(a) Open Parking Lot (b) Metallic Cubicle Area (c) Lounge (d) Hallway

Figure 5: Evaluation of RToF range measurements at different location. (a) Shows outdoor measurements, which characterize error sources other than multipath. (b) (c) and (d) show indoor locations. While the average error is high (4.2m), the measurements have a high correlation (94%) with distance.

Table 4: Performance of flight controller and sensor controller software in achieving stable hover on the SensorFly node.

Set Height	Maximum Overshoot	Settling Time (70%)	Avg. Steady State Height	Flight Time
1ft.	4ft.	25sec	1.5ft.	6:20min
2ft.	6ft.	40sec	2.5ft.	5:30min
3ft.	6ft.	50sec	3.5ft.	4:50min

locations. These errors are caused due to the specific configurations of the environment and a general model cannot be assumed. However, a higher correlation exists with distance (when compared to other distance metrics like RSSI, hop-count, etc.), enabling the SensorFly nodes to use the measurements for a biased-random walk exploration algorithm. This also provides a better metric that RSSI and hop-count for use in-network localization and topology estimation protocols [11, 28] while meeting the platforms relatively strict cost, weight and accuracy constraints.

5.3 Motion

We use PID controllers to implement the desired height and yaw control for SensorFly nodes. PID control is simple, does not require detailed dynamic models, and can be implemented using minimal computing power.

The PID controller for maintaining altitude is based on a first principles model of the SensorFly node given by,

$$X(s) = \left(\frac{\frac{1}{m}}{s^2 + \frac{N}{m}s} \right) F(s). \quad (1)$$

where x is the altitude, F is the input, m is the mass of a SensorFly node and N is the coefficient of viscous drag.

Blade rotation causes a torque to be applied to the helicopter body. In stable hovering condition, the top and bottom rotor torques, T_{top} and T_{bot}, should be balanced. The dynamic equation is given by,

$$I \times \alpha = T_{top} - T_{bot}, \quad (2)$$

where α is the yaw rate and I is the moment of inertia of the node about the vertical axis through the center of gravity.

A second PID controller, which adjusts the individual rotor speeds while keeping the total thrust constant, enables the craft to hold a certain direction (trim) or turn at a desired rate.

5.4 Flight Time

Table 4 shows the performance of the flight controller and sensor control software in achieving stable hover at a given height. The flight time is lower for hovering at higher target altitudes due to the higher overshoots and settling time. While the current system provides sufficient stability at very low computational cost, better control strategies remain the focus of our work in the future. The flight time of approximately 5 minutes is attainable with the prototype. Optimization of the mechanical design can extend flight time to 15 minutes as has been obtained by similar, although RF-controlled, flying crafts [2].

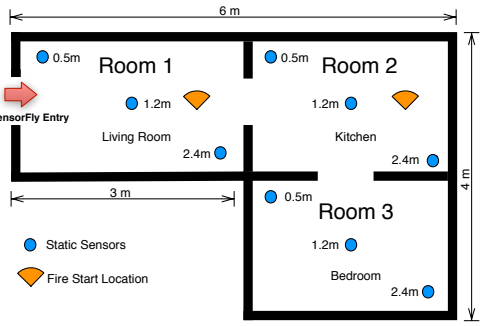


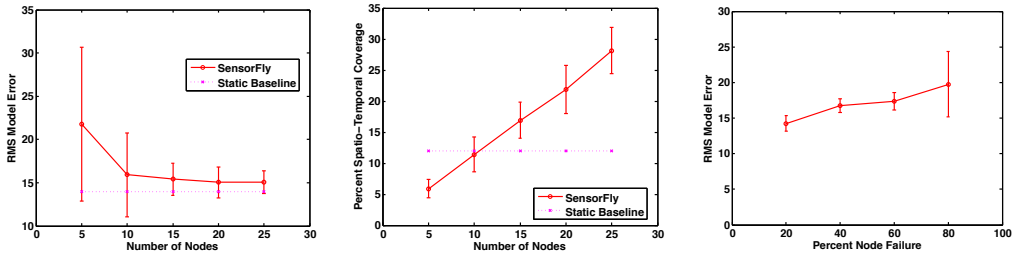
Figure 7: Arena for SensorFly and CFAST fire simulations. The figure shows the building geometry, the placement of nodes for the static network, the entry point for mobile SensorFly nodes and the point where fires are initiated at $t = 0sec$.

6. EVALUATION

In this section, we evaluate the performance of our platform in a realistic fire scenario. CFAST [30], is a computer simulation environment that fire investigators, safety officials, engineers, architects and builders use to simulate the impact of past or potential fires and smoke in a specific building environment. It is two-zone fire model used to calculate the evolving distribution of smoke, fire gases, and temperature throughout compartments of a building during a fire. Considering a multi-compartment building scenario, we extend the realistic fire growth model to include mobile SensorFly nodes with parameters and capabilities derived from actual experimental characterization presented in Section 5.

6.1 Methodology

We compare the performance of the autonomously deployed mobile SensorFly nodes with a statically pre-deployed network of sensors in a 3-room building shown in Figure 7.



(a) (b) (c)

Figure 6: (a) Average error in predicted model shown as a function of the size of SensorFly deployment.(b) The percentage coverage in time and height dimensions is shown as a function of deployment size.(c) Shows the error obtained in 25 node SensorFly deployments, when a percentage of nodes fail at $t = 900sec$.

The CFAST fire simulation runs for a total of $t = 1800sec$ seconds, providing fire evolution data such as layer interface height, temperature, pressure and gaseous composition along the height of each room. Fires are set in the Kitchen and Living room compartments at $t = 0sec$. Each room has typical furniture with their combustible properties as provided in the simulation environment. The CFAST zone model assumes the conditions at a certain height of the room to be uniform. Therefore, only variations along the vertical dimension in the building compartments are of interest in sensor placement.

The static network is pre-deployed consisting of 3 nodes placed at heights of 0.5m, 1.2m, and 2.4m in each of the 3 rooms. The nodes measure temperature at 10-second intervals and route the data back to the base node at the entrance of the building structure. The SensorFly nodes are introduced into the environment at $t = 0sec$ seconds into the simulation arena. The nodes deploy autonomously and route back sensed temperature readings to the base station at 10-second intervals, moving vertically to obtain data at different heights. Figure 7 also shows the placement of sensors in the simulation arena as well as the entry point for SensorFly nodes. The mobility models, network protocol, and radio link characteristics used for the simulation are described in the following subsection.

6.2 Evaluation Metrics

Both approaches provide discrete sensor readings in time and height. Sensor readings from both approaches are interpolated to give a continuous surface along the time and spatial dimensions. With the simulation data as ground truth, we define two metrics as a measure of the ability of the approach to provide accurate fire evolution predictions:

Average Model Error. This is defined as the root-mean-square error of the predicted model obtained by interpolating the discrete sensor readings reported by the static nodes and mobile SensorFly nodes. The readings are interpolated in both the height and time dimensions to the maximum resolution of the CFAST simulator, 0.1m in height and 10 seconds in time. This metric captures the performance of the system in terms of predicting an accurate model from sensed data.

Spatio-Temporal Percentage Coverage. The static nodes have limited resolution in the space dimension since only a few nodes can be economically installed as part of a universal infrastructure. Conversely, the mobile SensorFly nodes by virtue of being introduced into the environment and relying on autonomous means to deploy are constrained in the temporal dimension. That is, the mobile nodes may not be available at the desired location. The spatio-temporal percentage coverage captures the effect of both these characteristics. It is defined as the ratio of the number of sensor readings in the height-time plane to the maximum possible resolution obtainable.

First, we compute these metrics from deployments without accounting for the effect of network disruptions. Second, to evaluate the adaptability of the solution, we introduce network disruptions by randomly failing a subset of nodes. The failure of nodes causes loss of sensed data from the nodes themselves, as well as from the nodes, which are partitioned from the base station. We also consider the effect of the number of deployed SensorFly nodes on sensing effectiveness.

6.3 Simulation Framework

The simulation framework closely incorporates the real and experimentally evaluated capabilities and characteristics of the SensorFly platform. We describe the various components of the framework in the following paragraphs.

SensorFly Mobility Model

The SensorFly platform has the ability to measure height through the ultrasonic range finder, measure its pose through an integrated 3-axis compass, and measure the distance from other nodes through the radio's time-of-flight capability. Using only these capabilities, the simulated SensorFly nodes implement the following motion model:

- The SensorFly nodes follow a biased random walk for exploring and deploying in the on-fire building. The nodes move at speeds of 1 m/s in random directions until they are within a minimum specified distance of 3m from only one other SensorFly node, to maintain connectivity as well as spread out through the building for exploration.
- Nodes again move in random directions if no other node is within a maximum specified distance of 6m or when no data route exists to the base station. This behavior is to guard against partitioning of the network.
- Nodes move vertically at speeds of 0.5 m/s in a periodic fashion to obtain readings at different heights. The vertical position corresponding to a given reading is obtained from the height sensor.

Network Protocol

The static nodes as well as the SensorFly employ a simple network protocol, using aggregated broadcasts to route data back to the base station as described in Section 4.

We use shadowing with a path loss exponent of 3 as the radio link model for our simulations. This is similar to that employed by previous indoor fire monitoring work [35] and is an estimate for a single floor multi-room scenario.

Localization

We do not require any accurate localization for navigation or mobility as nodes follow random paths through the building. The capability of nodes to measure inter-node

distances through time-of-flight ranging is used to obtain coarse-grained location for tagging sensor readings. Knowing estimates of inter-node distances we employ iterative multilateration to classify sensor location into the three building compartments [31].

6.4 Results

The simulation starts at $t = 0$ with a fully deployed static network with nodes in all 3-rooms, and all SensorFly nodes introduced into room 1. Figure 6 (a), shows the average model error as the number of nodes introduced into the arena is increased. The error from the static deployment of 9 nodes, 3 in each room, is shown for comparison. The error is large for a small number of nodes primarily because of the lower coverage. The error and the variance in error decreases as the number of nodes in increased. The error stabilizes at 10 nodes, about the same as the size of the static node deployment. The difference in error between the autonomous SensorFly deployment and the base line static one is about 14% with similar sized deployments. It decreases further as the number of nodes increase. The stabilization can be attributed to the fact that once nodes are present in all 3 rooms, additional nodes increase redundancy but improve the error only slightly given the uniform model. A scenario with higher resolution sensing needs will benefit more with a larger number of nodes.

Figure 6 (b), shows the percentage coverage of SensorFly deployments of various sizes. The percentage coverage, as defined earlier, is the total points, in the height and time dimension, available from sensing to the maximum points provided by the CFAST simulator. In a real world scenario, the resolution would be infinite. However, this baseline corresponds to an ideal case sensing resolution suitable for the phenomena being sensed and its spatial distribution. The coverage increases with the number of nodes as expected, with SensorFly coverage being about equal to that of static nodes at the deployment size of 10 nodes. The coverage increases almost linearly thereafter. This is because the mobile SensorFly nodes can provide very high resolution in the height dimension. However, when the number of nodes is fewer, the nodes are slower to enter into a compartment and therefore have less temporal coverage. Arguably, this is a result of the biased random exploration and deployment method employed.

Figure 6 (c), shows the adaptability of the network to node failure and network disruptions. In a 25 node deployment, a percentage of nodes are failed at $t = 900sec$. The actual nodes to fail are picked at random from the total nodes for each run. The failure of nodes causes loss of sensed data from the nodes themselves, as well as from the nodes that are partitioned from the base station. The mobile node deployment shows a sub-linear increase in error for node disruptions. Increasing the number of nodes failed from 20% to 80% causes a 5-degree increase in error. As nodes not within range of other functioning nodes resume random motion, until a connected network is re-established. This makes the network adaptive to node failure.

Figure 8 shows the spatial coverage that is obtained by SensorFly nodes in the 3-room simulation arena over a duration of 1800 seconds. The arena volume is divided into 1 cubic meter volume regions. A region is covered if a SensorFly node visits it during the simulation run time. The coverage is computed as the percentage of regions covered by the node to the total number of regions in the simulation arena. We observe that for a given time a larger number of nodes, following our biased random-walk exploration scheme, can obtain higher coverage. However, diminishing returns are observed, for spatial coverage alone, once the number of

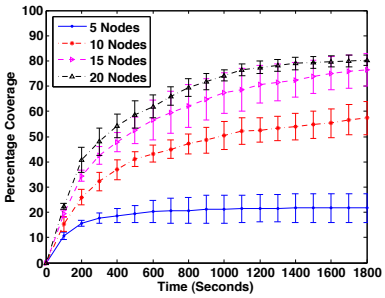


Figure 8: The figure shows the percentage 3-D coverage for SensorFly nodes in the 3 room simulation arena over a duration of 1800 seconds. The error bars show the standard deviation of coverage obtained.

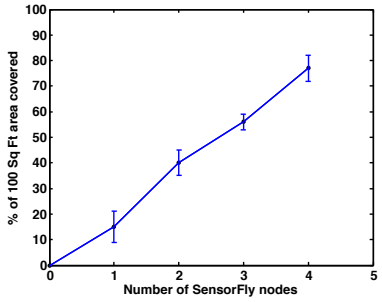


Figure 9: The 2-D space coverage in an area of 100 square feet over a duration of 2 minutes using the biased random walk exploration scheme.

nodes increases above 15 nodes. For the 5-node scenario, the coverage remains flat with time due to the inability of nodes to explore further while maintaining a network route to the base station. SensorFly nodes achieve better performance closer to areas where they are introduced.

6.5 Experiment

We performed a small-scale controlled experiment with real SensorFly nodes to validate the coverage trends obtained from the simulation. SensorFly nodes programmed with the biased random-walk exploration scheme were introduced into an enclosed area of 100 square feet as shown in Figure 10. The area was divided into square regions of 2 square feet each. A region was designated as being covered if a SensorFly node visits it during a time of 2 minutes. Coverage was determined visually as a percentage of visited regions to the total regions in the area. The number of nodes was varied from 1 to 4. Each experiment was performed 10 times and the average coverage with standard deviations is plotted in Figure 9. We observe a similar trend of increasing coverage with increasing size of deployment as seen in the simulation.

7. RELATED WORK

Research in sensor networks has been actively conducted for the better part of ten years. By far, the most explored area has been fixed networks [21, 34, 36]. These sensing applications have some similar characteristics to SensorFly, in that they are mostly networks with multiple neighbors, and nodes are composed of microcontrollers, radios, and sensors. The main metrics of the system are energy usage, data flow, and data aggregation.

Mobility has been explored in sensor networks largely in context of sensor nodes being carried by human beings or animals [17]. However, unlike SensorFly, the work focuses

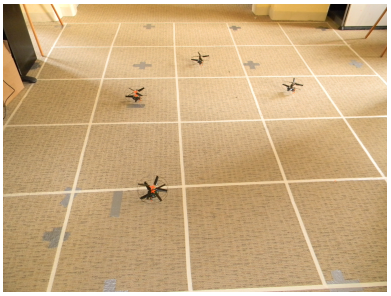


Figure 10: Experimental setup for determining 2-D space coverage in a 100 sq. ft. space with 4 SensorFly nodes.

mainly on adapting the system to user mobility.

Controlled-mobility has been envisioned by previous work in wireless sensor networks that proposes the idea of using controllably mobile elements in the network to alleviate resource limitations and improve system performance by adapting to deployment demands [18]. Somasundara et al. provide a theoretical analysis of the advantages of controlled-mobility in improving sensing fidelity and node lifetimes with prototype deployments using ground robot platforms [33]. The SensorFly, on the other hand, focuses on the hardware platform designed for enabling controlled-mobile indoor aerial sensing.

Mobility has often been explored as part of robotics research. A segment of research focuses on monolithic or a small team of robots, which may be remote-controlled or autonomous. Typically research on these devices focuses on individual stability and independent navigation, requiring sophisticated sensors [12]. Robotic platforms tend to have a higher per device cost and are economical for deployment in much smaller numbers compared to traditional sensor networks. Consequently, they are constrained in their ability to cover large areas simultaneously and rapidly.

The idea of miniature indoor flying platforms has been proposed before in literature. One work explores using a miniature electric helicopter to combine UAV flocking and wireless cluster computing [13]. Allred et al. present wireless link characterization for a network of semi-autonomous MAV's for atmospheric plume sensing [6]. SensorFly is the lightest functional system by at least a factor of 5, and targets a different design space. The SensorFly provides a platform for indoor sensing, accomplishing navigation, and networking under strict resource constraints, with a high degree of collaboration.

Wood et al. have worked on flapping-wing micro-mechanical flying devices capable of autonomous flight [37], weighing under 200mg. This is the target hardware platform for the RoboBees [3] project. We believe these flying mechanisms represent exciting advances, and underscore the need for research into controlled-mobile, highly resource constrained collaborative sensing and coverage algorithms. With a fully functional hardware platform, SensorFly allows us to validate assumptions and determine true tradeoff points in realistic deployments.

For fire monitoring, the FIRE [1] project proposes SmokeNet, a pre-deployed network of nodes with smoke and deferential temperature sensors. The system also consists of nodes with LED's to visually alert or guide firefighters. The SIREN [16] project focuses on improving the firefighter's access to information, using a WiFi-enabled PDA with peer-to-peer networking capabilities to communicate with an infrastructure of sensors. These sensors warn firefighters of hazards as well as help with navigation and localization. The FireGrid [35] project employs zone models and utilizes an array of static sensors positioned from ceiling to the floor. Using the correlation between the sensors, the system proposes a communication protocol for emergency response that minimizes congestion. The assumption of a universal pre-established infrastructure, in all of the above-mentioned work, limits their adoption in the near-term.

Another proposed approach has been that of automatically deploying nodes as firefighters advance into the building on fire [20]. These sensor nodes are primarily concerned with relaying firefighter data back to the incident commander. Since this approach involves firefighters entering the building in order for sensor deployment to take place, such a system would have limited utility for providing situational a priori information without risking the rescuers' lives.

8. DISCUSSION

Having presented a description and evaluation of our controlled-mobile aerial platform, we note that several aspects warrant further discussion and could result in possible extensions to this work.

Interoperability with Static Networks. Pre-deployed static sensor networks have certain advantages such as knowledge of exact locations and ability to provide data from regions that may be occluded due to closed doorways or structural collapse. On the other hand, the point-of-emergency deployment capability and mobility of SensorFly nodes allow larger deployments and higher resolution sensing of spaces where they can be introduced.

A hybrid approach with SensorFly nodes working in collaboration with static sensing infrastructure, where it exists, can combine the advantages of both approaches. Thus, research into making controlled-mobile and static networks interoperable, as well as, work on leveraging the static network to augment the mobile-network's localization protocols could be beneficial.

Communication Protocols. Networks of controlled-mobile nodes present new opportunities and challenges for design of communication protocols. In this paper, we present a simple aggregate and broadcast scheme that is sufficient for relaying the low-bandwidth temperature data. Nevertheless, we envision more complex communication scenarios with heterogeneous sensor nodes with varying bandwidth requirements. The network must provide quality-of-service for both higher-bandwidth data such as camera streams as well as low-bandwidth temperature readings. Another challenge is the co-existence of delay-tolerant communication with communication that has timeliness requirements such as that required for localization of nodes. Furthermore, the constant but controlled mobility of nodes provides opportunities in designing routing and discovery schemes better suited to such networks.

Radio Propagation. Radio propagation may be affected adversely in emergency response environments such as in presence of smoke and fire. This may impact the speed and extent of coverage obtained for a deployment of specific size. Recent research has studied the effect of fire on wireless propagation in wildfire environments [8]. The work suggests that ionization in flames causes particular frequency bands to be attenuated. An empirical study of the characteristics of fire propagation in indoor environments for our chirp spread spectrum radio would be helpful in obtaining a more accurate estimation of system performance in real deployments.

Energy and Flight Time. Miniature aerial platforms remain limited in their flight time due to high power consumption of motors. Improvements in mechanical design and battery technology can extend flight times to the order of 15-20 minutes, which however may not be sufficient for many applications. We seek to explore collaborative

techniques to distribute movement tasks and manage energy across the group. For example, in the fire monitoring scenario examined, nodes can collaborate with nodes in their close proximity, and *duty cycle* their flying task. One node can land and act as relay, while another flies and senses the temperature. When the battery level of the flying node becomes low, the roles can be reversed.

9. CONCLUSION

In this paper, we presented a novel 29g controlled-mobile aerial sensor network platform for indoor emergency fire monitoring applications. We identify the challenges in low-cost low-weight aerial sensing platform design and propose an architecture which utilizes limited-capability resource-constrained individual sensing nodes to autonomously and quickly achieve network wide sensing objectives.

We evaluated the platform in the fire-monitoring scenario using realistic CFAST indoor fire simulation models. We show that autonomously deployed SensorFly nodes can achieve performance in both sensing quality and coverage that matches or exceeds pre-deployed static network infrastructures. The autonomy and adaptability of SensorFly-like networks, can eliminate the cost of building large sensing infrastructures and the reduce risk to firefighters, as compared to prior static sensor network approaches.

10. ACKNOWLEDGEMENTS

Special thanks to our shepherd Dr. Thomas Schmid and the anonymous reviewers for their insightful and constructive comments. This Research was supported by the CyLab Mobility Research Center at Carnegie Mellon under grant DAAD19-02-1-0389 from Army Research Office. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies.

11. REFERENCES

[1] FIRE project. In *http://fire.me.berkeley.edu*.
[2] Intelli Mini RC Helicopter. In *http://www.hobbytron.com*.
[3] Robobees. http://robobees.seas.harvard.edu.
[4] USFA Firefighter Fatality Reports and Statistics. In *http://www.usfa.dhs.gov/fireservice/fatalities/statistics/index.shtm*.
[5] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for Human Spaces. *IEEE Pervasive Computing*, 6(2):20–29, Apr. 2007.
[6] J. Allred, A. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni. An Airborne Wireless Sensor Network of Micro-Air Vehicles. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, page 129. ACM, 2007.
[7] B. Bethke, M. Valenti, and J. P. How. Uav task assignment. *Robotics & Automation Magazine, IEEE*, 15(1):39–44, 2008.
[8] J. Boan, Jonathan Alexander. Radio experiments with fire. *IEEE Antennas and Wireless Propagation Letters*, 6:411–414, 2007.
[9] Digi International. XBee Multipoint RF Modules Datasheet. `www.digi.com`, 2009.
[10] C. H. Everett. Survey of Collision Avoidance and Ranging Sensors For Mobile Robots. *Robotics and Autonomous Systems*, 5(1):5–67, May 1989.
[11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 4(4), 2005.
[12] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin. The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC). In *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, volume 2, 2004.
[13] O. Holland, J. Woods, R. De Nardi, and A. Clark. Beyond swarm intelligence: the UltraSwarm. In *Swarm Intelligence Symposium, 2005*, pages 217–224, 2005.
[14] Honeywell. HMC6346 3-Axis Compass Datasheet. `www.honeywell.com`, 2008.
[15] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering. *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, 2009.
[16] X. Jiang, N. Y. Chen, J. I. Hong, K. Wang, L. Takayama, and J. A. Landay. Siren: Context-aware Computing for Firefighting. In *proceedings of second internatonal conference on pervasive computing, pervasive 2004*, pages 87–105, 2004.
[17] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences With ZebraNet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, volume 37, pages 96–107, New York, NY, USA, Oct. 2002. ACM Press.
[18] A. Kansal, M. Rahimi, W. J. Kaiser, M. B. Srivastava, G. Pottie, and D. Estrin. Controlled mobility for sustainable wireless networks. nstitute of Electrical and Electronics Engineers, Inc., 2004.
[19] M. Klann, T. Riedel, H. Gellersen, C. Fischer, G. Pirkl, K. Kunze, M. Beuster, M. Beigl, O. Visser, and M. Gerling. LifeNet: an Ad-hoc Sensor Network and Wearable System to Provide Firefighters with Navigation Support. In *Proceedings of Ubicomp 2007*, 2007.
[20] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu. Automatic and robust breadcrumb system deployment for indoor firefighter applications. In *International Conference On Mobile Systems, Applications And Services*, pages 21–34, 2010.
[21] R. Mangharam, A. Rowe, and R. Rajkumar. FireFly: a cross-layer platform for real-time embedded wireless networks. *Real-Time Systems*, 37(3):183–231, Dec. 2007.
[22] MaxBotix. LV-MaxSonar-EZ1 High Performanxe Sonar Range Finder Datasheet, 2007.
[23] R. Morlok and M. Gini. Dispersing robots in an unknown environment. *Distributed Autonomous Robotic Systems 6*, 2007.
[24] P. Muren. Passively stable rotor system for indoor hovering UAS. *International Powered Lift Conference*, (July), 2008.
[25] Nanotron Tecnologies Gmbh. Real Time Location Systems White Paper Version 1.02. pages 1–20, May 2007.
[26] Nanotron Tecnologies Gmbh. nanoLOC AVR Module Technical Description, 2008.
[27] R. D. Nardi, O. Holland, J. Woods, A. Clark, and W. Park. SwarMAV : A Swarm of Miniature Aerial Vehicles. *21st Bristol International UAV Systems Conference*, 2006.
[28] D. Niculescu and B. Nath. Ad-Hoc positioning systems (APS). *Proceedings IEEE GlobeCom*, pages 2926–2931, 2001.
[29] Parrot SA. Parrot AR.Drone. http://ardrone.parrot.com/parrot-ar-drone/usa/.
[30] R. D. Peacock, W. W. Jones, P. A. Reneke, and G. P. Forney. CFAST, Consolidated Model of Fire Growth and Smoke Transport (Version 6) User Guide. *Nist Special Publication*, (Version 6), 2005.
[31] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.
[32] L. Shantou Syma Toys Industrial Co. Syma rc helicopter. http://www.symatoys.com/.
[33] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5:958–973, August 2006.
[34] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226, New York, NY, USA, 2004. ACM Press.
[35] R. Upadhyay. Towards a Tailored Sensor Network for Fire Emergency Monitoring in Large buildings. *1st IEEE International Conference on Wireless Emergency and Rural Communications, Rome*, 2007.
[36] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.
[37] R. J. Wood. The First Takeoff of a Biologically Inspired At-Scale Robotic Insect. *IEEE transactions on robotics*, 24(2):341–347, 2008.