SensorFlock: A Mobile System of Networked Micro-Air Vehicles

Ahmad Bilal Hasan, Bill Pisano, Saroch Panichsakul, Pete Gray, Jyh Huang,
Richard Han, Dale Lawrence, Kamran Mohseni

Department of Computer Science
University of Colorado at Boulder

Technical Report CU-CS-1018-06

December 2006

SensorFlock: A Mobile System of Networked Micro-Air Vehicles

[1]Ahmad Bilal Hasan, [2]Bill Pisano, [1]Saroch Panichsakul,[2]Pete Gray, [1]Jyh Huang,
[1]Richard Han, [2]Dale Lawrence, [2]Kamran Mohseni
[1]Department of Computer Science, University of Colorado, Boulder.
[2]Department of Aerospace Engineering Sciences, University of Colorado, Boulder.
{Ahmad.Hasan, William.Pisano}@colorado.edu,

ABSTRACT

A SensorFlock composed of bird-sized micro aerial vehicles (MAVs) enables low cost high granularity atmospheric sensing of toxic plume behavior and storm dynamics, and provides a unique three-dimensional vantage for monitoring wildlife and ecological systems. This paper describes a complete implementation of our SensorFlock, spanning the development of our MAV airplane, its avionics, semi-autonomous flight control software, launch system, flock control algorithm, and wireless communication networking between MAVs. We present experimental results from flight tests of flocks of MAVs, and a characterization of wireless RF behavior in air-to-air communication as well as air-to-ground communication.

Keywords

Wireless Sensor Networks, MAVs, Applications, Deployments

1. INTRODUCTION

Large wireless networks scaling to hundreds of low cost airborne vehicles are largely still a vision today rather than a reality. In SensorFlock, our research goal is to make a substantial leap forwards towards this vision of hundreds of inexpensive, semi-autonomous, and cooperating airborne vehicles that sense and relay data over a wireless communication mesh network. We present in this paper our progress towards this goal, namely the design of our micro-air vehicle (MAV), the semi-autonomous flight control algorithm capable of hovering individual MAVs in loiter circles, flight validation tests of the MAVs, and an in-depth study of the RF characteristics of air-to-air and air-to-ground communication between MAVs. The benefits that will accrue to the research community from the SensorFlock project include the ability to enhance scientific applications with fine-granularity three-dimensional sampling, the distribution of the MAV aircraft design and software to the wider community, the eventual creation of airborne testbeds that scale up to hundreds of MAVs, and increased understanding of wireless propagation characteristics and networking connectivity behavior between large numbers of MAVs that are rapidly banking, rolling, and changing attitude in flight. The latter measurement results will aid the computer science community in developing more realistic RF models for in situ air-to-air and air-to-ground communication, thus leading to improved simulation and design of more robust protocols for practical airborne sensor networks.

SensorFlock provides the capability to enhance many applications of interest to the scientific community by providing finer granularity three-dimensional sampling of phenomena of interest than would otherwise be feasible. One such class of applications is chemical dispersion sampling. As shown in Figure 1, a deployment of a flock of MAVs sensing and communicating their data back to a network of ground stations enables scientists to study the rate of dispersion of a toxin, pollutant, or chemical, natural or man-made. In another example, MAV flocks may provide the ability to study the distribution of CO_2 concentrations in the atmosphere and its relation to global warming. In all these cases, a flock of MAVs enables accurate sampling of the parameter of interest simultaneously over large regions of a volume. In addition, since MAVs are independently controllable, they can be targeted to track the toxic plume to study the rate of dispersion, fly towards the source of the plume if unknown, and re-distribute to map the boundaries of the plume.

Another class of applications that would benefit from SensorFlock are those involving atmospheric weather sensing. A flock of MAVs - each MAV equipped with temperature, pressure, humidity, wind speed/direction, and/or other sensors - can provide detailed in-situ mapping of weather phenomena such as hurricanes, thunderstorms, and tornados, and return data that would be useful in improving storm track predictions and the understanding of storm genesis and evolution. Other such examples include: modeling the local weather produced by wildfires to better predict their evolution and improve the deployment of firefighting resources; sensing and modeling of thermodynamic plumes over open ice leads in polar regions to better understand interactions between sea, ice, and atmosphere which contribute to climate change; and improved characterization of heat islands above cities and their impact on local weather patterns.

While many technologies exist that can contribute to the SensorFlock vision, they are either too costly, too restricted, or too limited to fully achieve by themselves a low cost and retargetable airborne sensor network. Passive sensors such as weather balloons and dropsondes cannot be retargeted

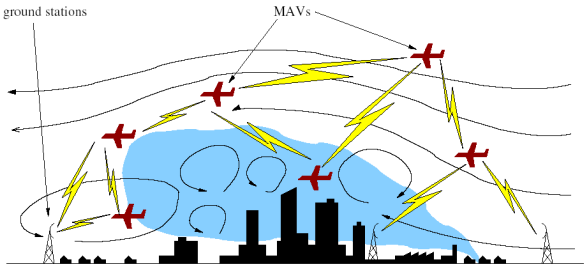


Figure 1: SensorFlock for 3-D sensing of toxic plumes.

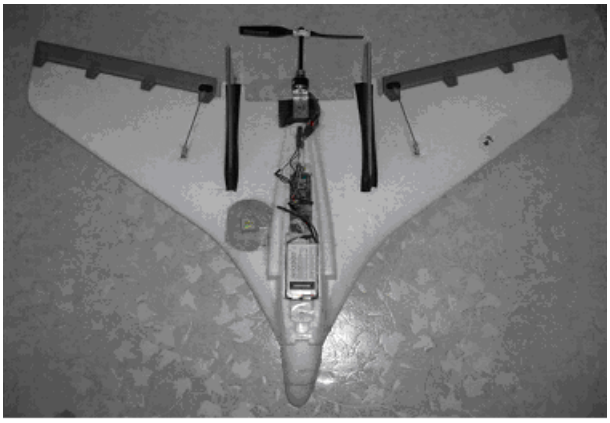


Figure 2: Micro Air Vehicle (MAV) designed and built for flight tests of SensorFlock at CU-Boulder.

to phenomena of interest. Large Unmanned Air Vehicles (UAVs) of 2-3 meter wingspan or more pose a hazard to conventional air traffic and ground personnel. Small, bird sized sensor vehicles have the potential to reduce the consequences of failure to levels that are considered of equivalent safety to FAA approved manned vehicles.

We have therefore pursued a vision of building a SensorFlock composed of many low cost small bird-sized MAVs on the order of a half a meter in wingspan. An example of the MAV that we have built is shown in Figure 2. This MAV would pose little danger to personnel and property on the ground or other air vehicles. They do not need specialized take-off or landing facilities or runways. They are reusable, and could be produced in large numbers at low cost. With a few enhancements to our current prototype's airframe and propulsion system, such small vehicles could potentially remain in flight for periods of about 90 minutes, sufficient to provide highly accurate data for decisions in the critical initial period after a toxin release event. Subsequently, fewer numbers might be used to monitor dispersions over longer periods.

Deploying a Sensor Flock with large numbers of vehicles (e.g. tens or hundreds) raises unique research challenges in command and control. Each MAV carries very limited on-board power and computing resources. Flight control, toxin sensing, information processing, communication, and decision making must be extremely simple and decentralized. Yet rather sophisticated aggregate behavior is desired, so that the flock can semi-autonomously seek out plumes, guided by supervisory human operators and real-time models of plume evolution.

This paper describes our solution to large-volume atmospheric sensing, that takes the approach of using a "minimal" autopilot combined with a globally stable and convergent vector field guidance system on each vehicle. This provides a small, low mass, and low cost autopilot system that requires very little human interaction in the form of flight control or path planning. This combination provides a semi-autonomous capability for each UAV, where the operator or an overseeing algorithm can provide the desired center of loiter coordinates and a loiter radius infrequently, decentralizing the vehicle control by moving the management of the flock to a higher level in the control hierarchy. Once the plane reaches its destination it will fly loiter circles around the target point until it is told to do otherwise. With this approach many vehicles can be controlled by a single operator without the threat of failure due to a lack of command or loss of communication. In this manner, the SensorFlock can scale to large number of MAVs launched and overseen by a small number of human operators. The sections below describe how this system is implemented and show experimental results of the system in action.

2. RELATED WORK

Detailed measurements of wireless behavior are essential to understanding real-world performance of wireless networks. For 802.11 2.4 GHz networks, measurements have been performed on static WiFi LANs [1, 4, 2, 9, 8], and an entire workshop has been devoted to the topic [10]. Mobile vehicular 802.11 ground networks have recently been studied [7].

In wireless sensor networks, detailed measurements have been performed on testbeds of static wireless motes at 900 MHz [11, 27, 22, 29]. Experimental papers studying 802.15.4 radios at 2.4 GHz in static sensor networks have also been reported [19, 20, 28].

Prior work has explored using large UAVs for toxin dispersion characterization [16], though this is in simulation only. Other prior work have simulated UAV networks [24, 23, 12].

Practical airborne systems of wireless networked planes are largely in their infancy. A system with several small helicopters has been reported [3]. The AUGNET project reports results for two of the larger UAVs [5, 6, 14] networked via 802.11, not for the smaller bat-sized MAVs.

As far as we are aware, there is no prior work studying the network dynamics of a Sensor Flock composed of bat-sized MAVs. MAVs have attracted significant attention since the mid-1990's for both civilian and military applications. Pioneering work in this area was conducted by AeroVironment [13] and the University of Florida [15] among others. MAVs are by definition small (by weight or size) aircrafts which fly at relatively low speeds. Such flight characteristics will result in flow regimes with Reynolds numbers below 200,000. Another aerodynamic signature of MAVs is wings with small aspect ratio; in most cases the chord is roughly equal to the wingspan. This combination of low Reynolds number flight and low aspect ratio wings results in a flow regime alien to conventional aircraft. Although small birds and insects have been flying under these conditions for quite some time, this

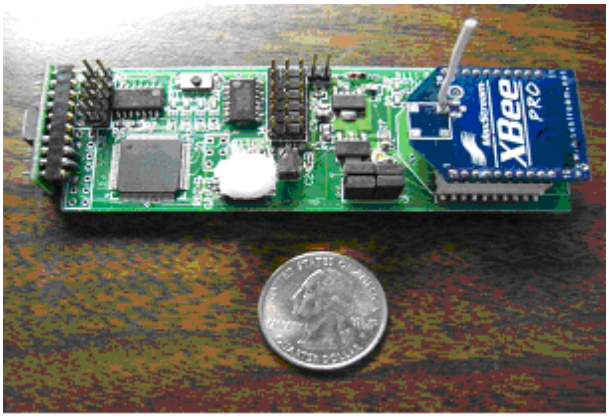


Figure 3: CUPIC Autopilot board. This view shows the top of the board which houses the CPU, pressure sensor, radio, and rate gyro. The integrated GPS receiver is on the bottom of the board.

is a new flight environment for man-made aircraft.

Our early work on pursuing a Sensor Flock focused on developing control algorithms to manage a flock of MAVs, wherein the location of each MAV was governed by an equation that was a function of the sensing phenomena of interest, the concentration of MAVs in a given area, communication requirements, and energy [18].

3. MAV SYSTEM

The design of the MAV system faced three significant challenges: building a sufficiently lightweight plane capable of flight; designing algorithms into software to achieve semi-autonomous flight; and integrating disparate subsystems such as propulsion, flight control, and wireless networking into a fully functional SensorFlock solution.

3.1 MAV Plane

As shown in Figure 2, the small size of the MAVs designed at the University of Colorado has the advantage that in the event of an accident, the potential for damage to property and personal on the ground is minimal. By keeping the vehicle mass under 500 grams and the maximum speed, even in failure, under 20 m/s, these vehicles fall within NASA's "Inert Debris" range safety classification. Adding that the plane is made of polyproplyene foam, and the propeller is in the rear, the potential for collision damage is minimal. Combine this with a production cost of the entire aircraft, including the autopilot, which is less than $600, and it is clear that the cost of failure of one of these vehicles is minimal. It is for this reason that we do not require redundant systems such as those present on larger aircraft autopilot systems.

3.2 Avionics

The CUPIC avionics board is structured around the Microchip PIC18F8722 8-bit microcontroller. The addition of an RC class receiver for testing and fail-safe purposes, and an antenna for the GPS receiver completes flight package. Analog flight sensors consist of a single roll rate gyro and an absolute pressure sensor, while the GPS sends binary navigation information to the CPU. The control system runs on the PIC and outputs commands to the motor and servos via the on-board PWM interface. An XBee Pro Zigbee class 2.4 GHz radio integrated is used in the study of mobile networking. The system is capable of being flown entirely through the XBee, but the backup RC link is maintained for network testing when there is a possibility of losing contact with an aircraft.

Figure 4: Plane-A-Pult Automatic Aircraft Launcher performing a fully autonomous launch.

3.3 Fail-safe Operation

Failure is an important consideration when operating any aircraft. There are many events that can occur with an unmanned aircraft causing different types of failures. Several fail-safes are present in the autopilot system to cope with these contingencies. A watchdog timer is present that will restart the CPU if the timer is not periodically serviced. This avoids the case where a software bug causes a system crash, preventing the control system from executing regularly. The watchdog is set for 1/2 second and should nominally be serviced about 300 times per second. The current setup uses two RF links so that the networking link can be tested. With this setup no fail-safes are applied to the XBee radio. In fact this radio could be removed entirely and the aircraft would still be able to fly normally. The second RF link is a modified Pulse Code Modulated Remote Control radio, commonly used in the RC modeling community. The range of this radio is approximately 1.5 kilometers. Current operating procedure dictates that the aircraft be within visual range at all times. Visual range is about 1/2 kilometer, which is much less than the range of the radio, thus loss of contact due to range should not be an issue. In the potential case where the plane starts flying away from the operator, there is a fail-safe in place that will turn the motor off when the plane loses the RC link. With the motor off, the aircraft will gently glide down. In the case where the CPU stops responding altogether, the electronic speed controller for the motor automatically turns the motor off when it stops receiving commands, thus in this case the aircraft will glide down as well. A battery monitor circuit monitors the voltage of the flight battery pack. When the voltage drops below a predetermined level during the flight the remaining capacity of the pack, and thus remaining flight time can be inferred. When missions requiring flight farther away from the base station are being flown, a level can be set that gives the aircraft sufficient power to return to the base station. Once it has returned it can initiate landing mode which turns the motor off and descends slowly until the ground is reached.

3.4 Launcher

To further make the MAV system operable with as little human interaction as possible, an automatic launching system was developed. The design makes use of a rugged aluminum frame propelled by a constant force coil spring. A "V" shaped aircraft carriage is used that holds the aircraft

by the wings while allowing the propeller in the rear of the plane to spin-up prior to launch. A release servo is mounted to the Plane-A-Pult, and connected to the avionics board with a 3-wire interface. The avionics board then sends a signal to the actuator to initiate release. When released the 3-wire connecting plug is pulled disconnected, thus leaving the release actuator behind. Using this method, no human interaction is required for the plane to take off. The avionics can determine when all of its sensors are on-line and determine when to release itself. The autopilot is in control of the aircraft the entire time, thus no pilot is required. Any number of aircraft could be launched simultaneously using this method and a single operator sending a "launch all" command.

4. CONTROL SUBSYSTEM

The key to the structure of this control system is the accurate estimation of roll angle from a noisy roll rate sensor and GPS. By structuring the control system in this manner, it is more dependent on the kinematics of flight than on the dynamics of the aircraft being flown. In fact the only term that is specifically aircraft dependant is the moment of inertia in the roll direction. Other than that the system assumes only that the aircraft is passively stable in pitch and yaw, and that the aircraft has a natural tendency toward coordinated turns, which is true for most well designed non-aerobatic airframes. Given this structure, the autopilot could theoretically be used on a completely different airplane by simply calculating and inserting the roll axis moment of inertia.

4.1 Flight Model

A functional block diagram of the control system is shown in Figure 5.

At the center of the model is the aircraft roll axis model, which is structured as a simple inertial system. The Laplace transfer function is shown below.

$$F(s) = \frac{1}{Js + \omega_0} \tag{1}$$

where J is the moment of inertial about the roll axis and ω_0 is the damping coefficient of the aircraft in the roll direction. The remainder of the model shown in green shows the kinematic relationship between the roll axis, heading, and position. The key kinematic relationship is the derivation of heading rate from roll angle. This relationship is illustrated in Figure 6.

In any aircraft, the lift vector $\vec{L}$ changes with the roll angle as shown in the figure above. It is apparent that in order to maintain altitude, the y-component of the lift vector must cancel out the weight. The x-component of the lift in the diagram above does not cancel with any other forces, thus the aircraft feels an acceleration in the direction of the vector above. The magnitude of that acceleration is given by:

$$L_x = |\vec{L}| \sin\theta \tag{2}$$

where θ is the roll angle. This net force results in a lateral (centripetal) acceleration that causes the aircraft to turn, i.e. to produce a rate of change in heading $\dot{\psi}$:

$$a_c = L_x = \frac{|\vec{V}^2|}{r}; \omega = \dot{\psi} = \frac{|\vec{V}|}{r} \tag{3}$$

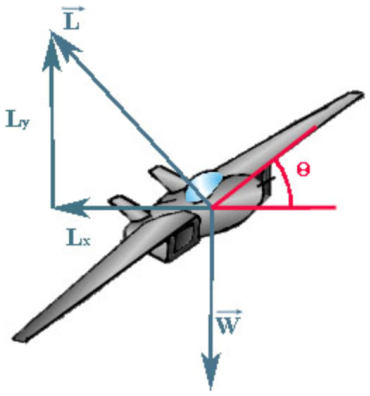


Figure 6: Kinematic relationship for a turning airplane.

where $\vec{V}$ is the forward velocity of the aircraft, and r is the instantaneous radius of curvature of the turn. When these three equations are combined, the result is the relationship between roll angle and heading rate, shown below.

$$\dot{\psi} = \frac{|\vec{L}|}{|\vec{V}|} \sin\theta \tag{4}$$

A simple way of estimating heading rate from roll angle and vice-versa is achieved by linearizing this relationship about two conditions: (1) the aircraft roll angle is small, thus $\sin\theta \rightarrow \theta$, and (2) the lift and velocity are nearly constant. With these two assumptions, the kinematic relation becomes

$$\dot{\psi} = k\theta \tag{5}$$

4.2 Autopilot Software Control

The blocks shown in green in Figure 5 represent the model of the aircraft. The outputs of these blocks correspond to the outputs of the sensors on board; the roll rate as measured by the roll gyro, the heading angle as measured by GPS, and the vehicle position is measured by GPS. The blocks that are not colored green are part of the control system and represent calculations done in software on board the aircraft.

The roll rate feedback loop (pink block) works as a high-gain rate damper to increase the viscosity of roll motion caused by external sources such as wind. This loop runs at a very high rate, approximately 100Hz.

The heading hold feedback loop (blue blocks) calculates a desired heading from the Lyapunov vector field (described below), compares it to the actual heading as measured by the aircraft, then uses this error to generate a roll command that will guide the aircraft toward the desired heading. This loop runs at a rate of 10Hz.

The roll angle feedback loop (orange blocks) is the key to stabilizing the motion of the aircraft. First the rate gyro measurements are integrated to get a measurement of roll angle. This measurement is susceptible to drift over time, so it is passed through a high-pass filter that removes low frequency drift errors. The heading angle from GPS is differentiated to find the heading time rate of change. This heading rate is then converted to an estimated roll angle according to the vehicle kinematics. This estimate is suspect

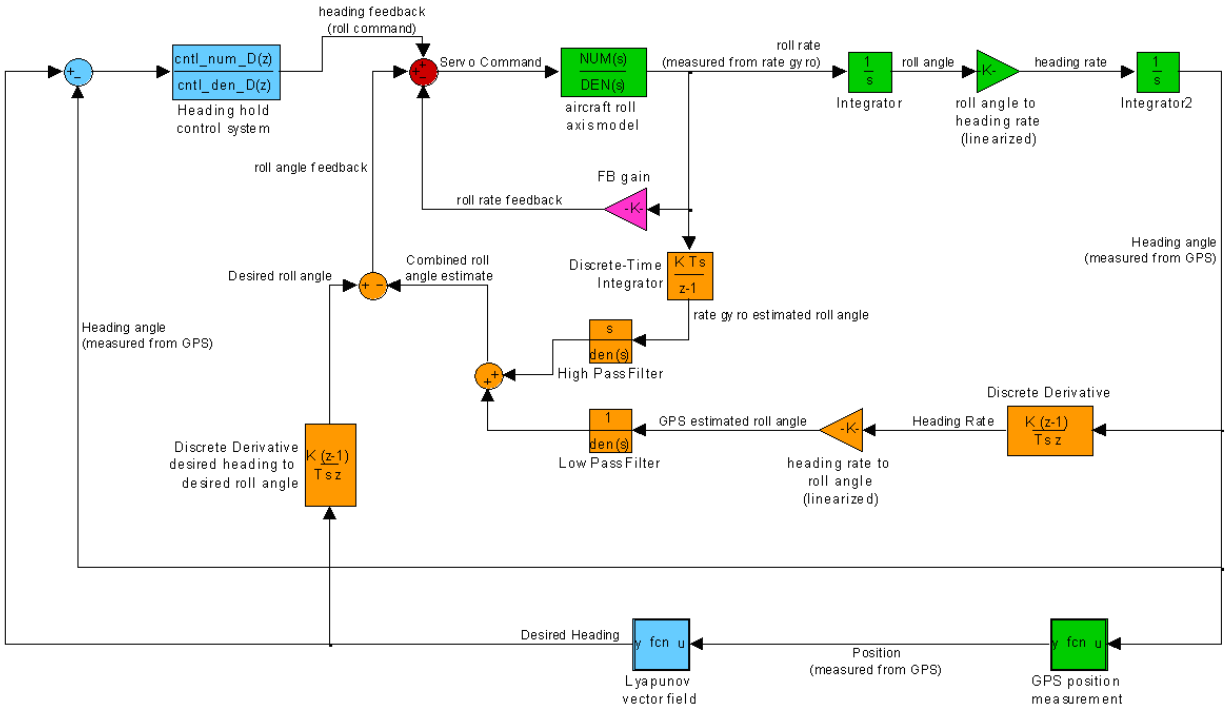


Figure 5: Simplified block diagram of the heading hold control system.

to high frequency noise due to the discrete time differentiation, so it is low pass filtered. The low frequency roll angle estimate is very accurate because it essentially uses several consecutive GPS measurements, however the time it takes to make this estimate is too long to enable high gain rate feedback. Thus the high-passed rate gyro estimate of roll angle and the low-passed GPS estimate of roll angle are combined to form an estimate of roll angle. By matching cutoff frequencies for the filter pair, no pertinent roll information is lost, but the weaknesses of both sources of the estimate are removed. This combined roll angle estimate is compared to a desired roll angle found by performing a differentiation on the desired heading output of the Lyapunov vector field and applying the kinematics described above. This comparison is then used to drive the aircraft to the desired roll angle. The combination of these three feedback control loops drives the aircraft to the desired heading and the desired roll angle, and provides viscous damping of roll motion for smooth flight.

The vehicle is guided by a vector field that has two terms that define vehicle motion: contraction to a desired loiter circle, and circulation on this circle to maintain desired air speed. For any known position of the vehicle, a velocity vector is defined by this vector field. The heading of this vector is given as the desired heading input to the heading-hold autopilot. When the next position sample is taken from GPS, a new vector is computed and fed into the autopilot. The vector field is designed to be a smooth function that will guide the aircraft from any initial position to the desired loiter circle. The autopilot takes further advantage of this fact by locally differentiating the vector field to determine the rate of change of the commanded heading vector, which acts as the desired roll angle for the control system. The following set of equations define the general case Lyapunov vector field that is being used.

$$\dot{\vec{r}} = \dot{\vec{r}}_d = h(\vec{r}) = -[\frac{\partial V_F}{\partial \vec{r}}\Gamma(\vec{r})]^T + S(\vec{r}) \tag{6}$$

$$\frac{\partial V_F}{\partial \vec{r}} = |\vec{r}_n|\hat{n}^T + (|\vec{r}_t| - \rho)\hat{r}_t^T \tag{7}$$

$$\Gamma(\vec{r}) = \frac{1}{\alpha(\vec{r})}I; S(\vec{r}) = \gamma\frac{\hat{n} \times \vec{r}_t}{\alpha(\vec{r})} \tag{8}$$

$$\alpha(\vec{r}) = \frac{1}{\nu}(|r_n|^2 + (|r_t| - \rho)^2 + \rho^2\gamma^2)^{\frac{1}{2}} \tag{9}$$

The quantity of interest here is $\dot{\vec{r}}_d$, or the desired velocity vector. $\vec{r}$ is the aircraft's position vector relative to the center of the desired loiter circle. $\vec{r}_n$ and $\vec{r}_t$ are the normal and tangential components of $\vec{r}$ with respect to the loiter circle (in plane and out of plane components where the plane is defined by the plane containing the target circle, the x-y plane for the 2-D case). ρ is the radius of the desired target loiter circle, $\hat{n}$ is a unit vector in the direction normal to the circle, V_F is a Lyapunov function, whose gradient $\frac{\partial V_F}{\partial \vec{r}}$ forms the contraction term in the vector field. I is the identity matrix, and ν is the speed of the vehicle, in this case a constant. γ is the factor that determines the amount of circulation vs contraction of the vector field, as well as the direction of the circulation. T is the standard vector transpose operation. A simple change of coordinates allows the center of the target circle to be placed at any arbitrary

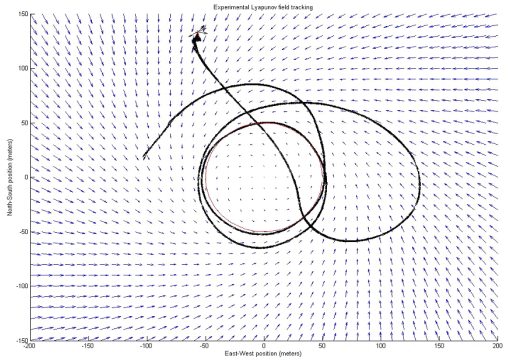


Figure 7: Experimental implementation of the autopilot system and Lyapunov vector field tracking algorithm.

Figure 8: Two loiter-circling MAVs, one following the other.

point. Thus by pre-defining the loiter center, radius, and circulation coefficient; a fully deterministic vector field is produced that requires only the current vehicle position to calculate a desired velocity vector. This approach has been shown to produce global convergence to a limit cycle loiter circle [17].

4.3 Experimental Validation

The plot below shows a semi-automated flight from takeoff to landing. The loiter center and radius were pre-programmed, so the only human interaction was to send the takeoff command and the land command. The aircraft was launched using the above described Plane-A-Pult from the red plus sign near the top of the plot. Take off mode controlled the plane to climb straight out until it reached a pre-defined target altitude of 30 meters. When this altitude was reached the autopilot was changed to its main loiter mode and the plane initiated a left turn to find and track the target loiter circle. After two full circles about the center point the land command was sent by the operator which caused the autopilot to descend slowly with the motor off until the ground was reached.

A flight test was performed with two aircraft on the same loiter circle with one following the other. The snapshots in Figure 8 are from a video taken of this event.

4.4 Controlling Flocks of MAVs

One application of the sensor flock is the measurement of toxic concentrations in atmospheric plumes. One approach uitilizes an on-board de-centralized potential field algorithm where vehicles share information and avoid collisions. The vehicles use this algorithm to determine where best to place the center of their loiter circles. Figure 9 shows a 147 MAV simulation as the vehicles attempt to characterize a three-dimensional toxic plume, represented by the isosurface in the figure below. The vehicles in Figure 9 are color coded according to their current assignments. The green vehicles are locating the source of the plume, blue are within a few percent of the isosurface of interest (shown as the red shell). Red vehicles are inside the plume and searching for the isosurface, and the black vehicles are spread outside of the plume in case a new source of concentration emerges. For more information on this particular application, the reader is referred to [21].

5. WIRELESS NETWORKING

Our goal in the networking implementation was to design and build the software and experiments necessary to characterize RF performance and networking connectivity of air-to-air and air-to-ground behavior. This approach will improve the accuracy of new models developed to simulate realistic airborne communication and movement.

We designed a series of experiments to collect received signal strength indicators (RSSI), packet loss and throughput statistics, and GPS-based position and time. From these sets of data, we were able to infer several fundamental factors such as the relative distance between the transmitter and receiver, either one or both of which could be a flying MAV, and the orientation of the MAV, e.g. banking angle, which affects the orientation of the antenna. We investigated 1-plane, 2-plane, and 5-plane scenarios. In each experiment, MAVs would periodically report their GPS-based (x,y,z) positions and GPS-assigned time, as well as other information.

The MAVs are equipped with 802.15.4-compliant radios, namely the XBEE Pro Zigbee radios from Maxstream. These radios were chosen for their combination of light weight, long transmission range, serial interface compatibility with the PIC processor, and packet interface. The radios offer a range of over 1 mile at 60 mW, though we measured at times ranges of 2 miles or more in early balloon-based testing. This is also fit the anticipated maximum spacing between neighboring MAVs deployed in our toxic plume scenario. The radio weighs just 4 grams. The packet interface offered by the XBEE Pro substantially simplifies packet processing, as the PIC is not burdened by processing each bit that arrives, as would be the case in more primitive bit-interface radios, such as the Chipcon CC1000 common to MICA2 motes. The packet interface reduces the memory cost and CPU time required for networking on the PIC, and is important for enabling the PIC to sense, process and actuate flight control in real-time. Power consumption of the radio is relatively modest compared with powering the aircraft's propellor propulsion system.

The software controlling the plane's flight via the PIC also is used to route packets through the multi-hop wireless network. In order to meet these real-time constraints of flight control, it was necessary to design the networking software to complete its time quantum in less 10 ms, where the autopilot controller needs to operate at 100 Hz, though in reality 40 Hz should suffice. To maintain simplicity, we implemented a simple software control loop that first consulted the controller and then invoked the networking subsystem, alternating between these two subsystems in an indefinite

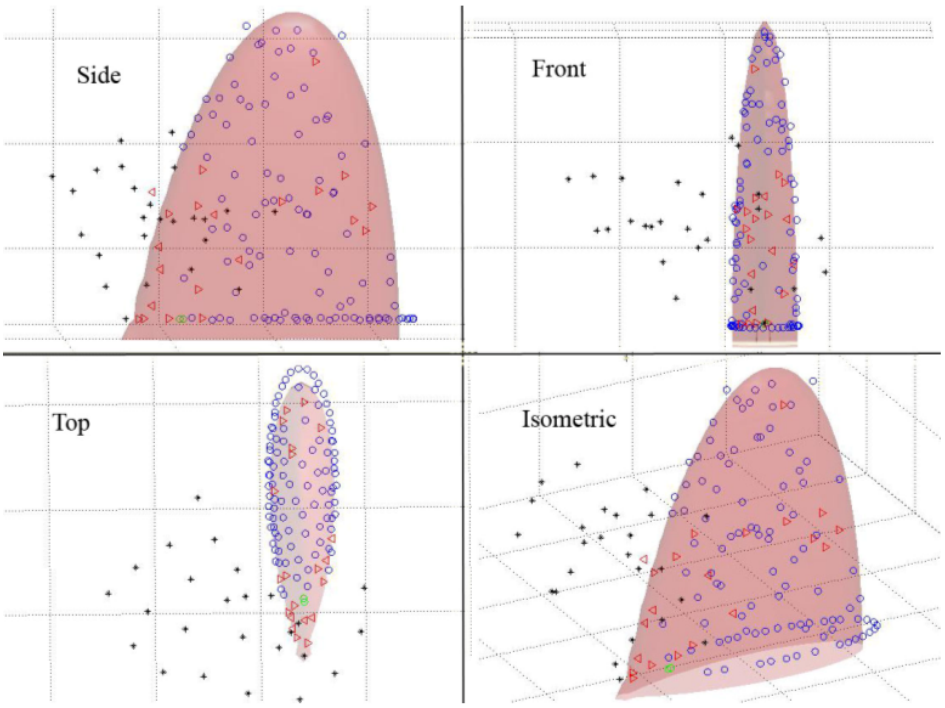


Figure 9: Multi-vehicle flocking. Vehicle are distributed with gradient field modifications.

loop. The networking code was written to quickly execute in its time quantum of 10 ms, and we estimate it did not take more than 0.5 ms in practice. Our networking code only processes one packet per invocation, so as to relinquish control back to the autopilot as soon as possible. This constrained the networking code from taking over the processor. The entire code uses about 1 KB out of the total of 4 KB RAM for the PIC, with about 150 bytes devoted to the networking. We found that this approach was acceptable for maintaining flight while also simultaneously reporting and routing networking data on the PIC processor.

In the future, we plan to investigate software mechanisms, such as are available in real-time operating systems, to enforce real-time operation of the autopilot, and constrain the CPU time devoted to networking.

6. WIRELESS EXPERIMENTS

The experiments were performed on sets of one, two and five MAVs. The networking code sourced packets at a rate of 10 packets/sec. Each packet contained at least the following: GPS (x,y,z) coordinates; GPS time; packet sequence number; source ID; hop count; and a field for substituting the RSSI upon reception. Application packets are of fixed length of 24 bytes. In each MAV, the antenna is a quarter wave whip with a very small ground plane, so that the pattern can be approximated by the donut-shaped antenna pattern of the half wave dipole, with the null on the bottom a little bigger than the null on top. The antenna is oriented vertically, pointing upwards when the plane is at rest. Transmit power was set to the lowest value of 10 dBm, in order to achieve multi-hop over a reasonable area. The 802.15.4 radios were configured for a rate of 115 kbps. API mode was enabled for the radios, so that data could be sent and retrieved in a pre-defined packet format. API mode also

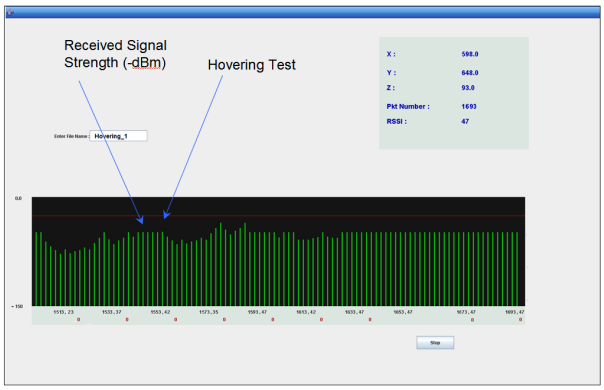


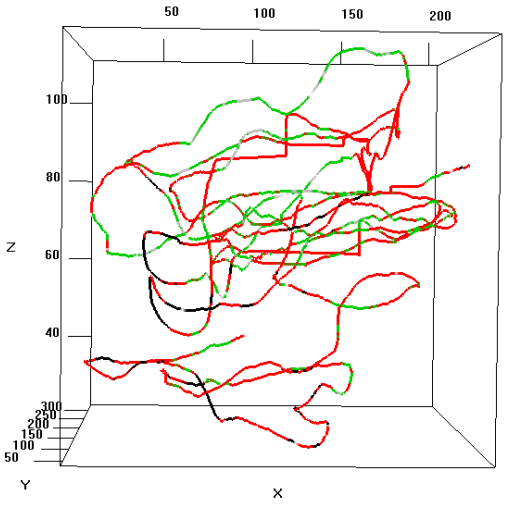
Figure 10: Java application for visualizing real-time MAV data.

conveniently returns the RSSI of received packets. Packets are only returned by the radio if they pass the 802.15.4 error detection check.

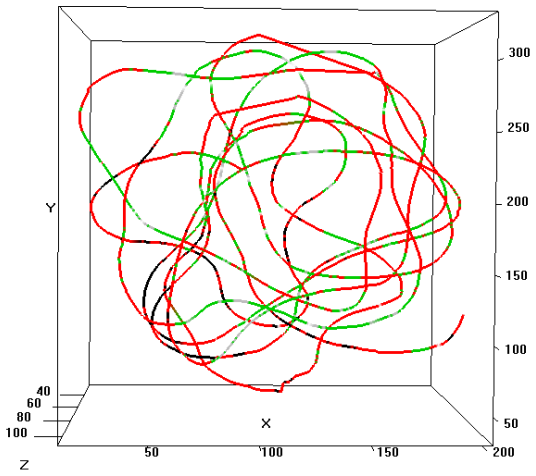
We built a Java program for capturing the stream of packets from the radio's serial interface, archiving the data to the PC, and visualizing the data on the laptop for in-the-field debugging and management. The Java program plotted the RSSI versus sequence number in real time, with gaps included. In addition, the coordinates of the plane are drawn in real time. Figure 10 shows a screen capture of the application.

6.1 One-Plane Tests

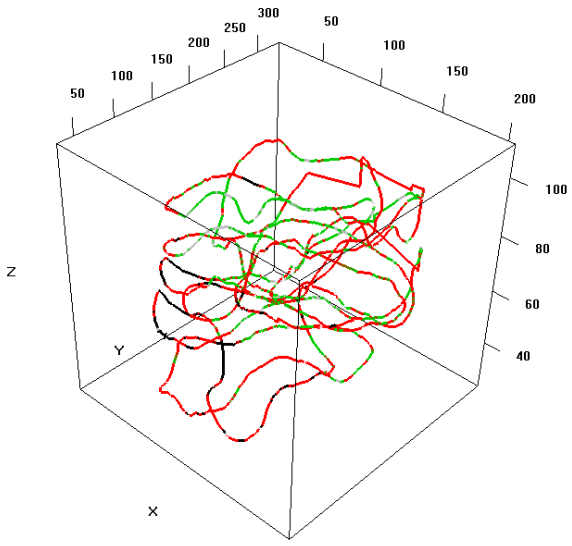
In the one-plane test, the MAV was assigned to hover around a fixed point, which was near the base station, at an



(a) Front view of loiter circle/hovering



(b) Top view



(c) Side view

Figure 11: Loiter circling trajectory of MAV.

altitude of no more than 50 meters. The base station was located within the hover circle, but was not the central point on that circle. The hover radius is also 50 m. The MAV was launched and immediately would begin streaming its coordinates back to the ground station, where RSSI values would be calculated for each packet.

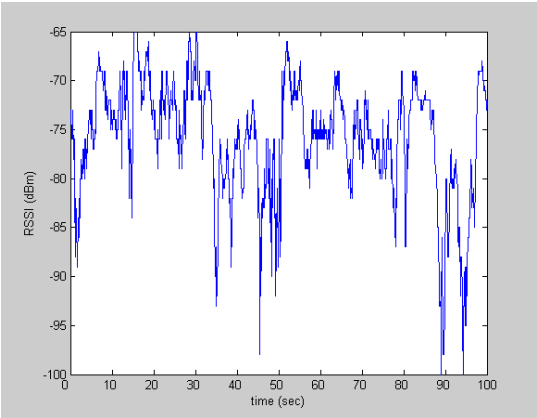
The three-dimensional trajectory traced by the MAV plane in one of the one-plane flight tests is shown in Figure 11 from three different perspectives. The trajectory shown captures a window of 6 minutes during the flight test. The front view in Figure 11(a) shows the MAV gradually rising in circles to its loiter altitude of approximately 50 meters. As can be seen, the ascent is not smooth, and is marked by considerable variation even within a single loop. The top view in Figure 11(b) reveals the loiter circling behavior of the MAV. Clearly the loitering is far from a perfect circle. The angled side view is shown in Figure 11(c), and is from the perspective of looking down on the cube of Figure 11(a) from the upper right vantage point. These different vantage points were facilitated by plotting the raw data using the R project's 3-D graphical plotting software [26]. Together, these viewpoints provide a good sense of the dynamic behavior of MAV flight paths.

The RSSI is also plotted via color shading of the trajectory in Figure 11, providing insight into the RF dynamics of MAV flight behavior. Darker RSSI values correspond to stronger RSSI readings, e.g. black means RSSI in the range from -60 to -70 dBm, red covers the range from -70 dBm to -80 dBm, green equals -80 to -90 dBm, and -90 to -102 dBm (minimum reception level of the radio) corresponds to gray. We observe that as the MAV hovers, its RSSI readings tend to stay within one band for a long time, before transitioning to another band. We did not observe rapid transitions back and forth from one quantization level to another.

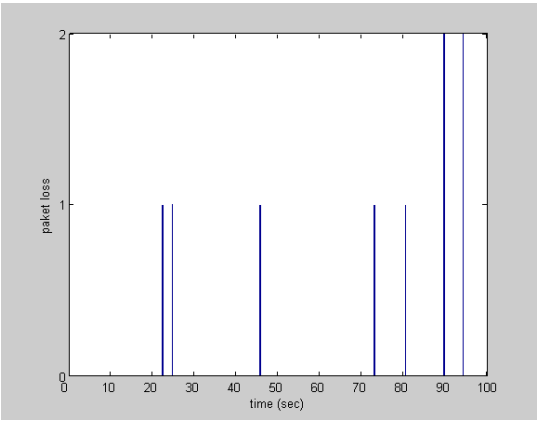
Further, as the RSSI trends towards green, we notice increasing discontinuity as more packets are lost, thus forming gaps in the RSSI trajectory, especially noticeable in the top view. The size of the gaps is relatively small, lasting only a few seconds, where each hover is completed approximately every 50 seconds. There are few gray packets, as the network connectivity tends to drop rapidly from the -80 to -90 dBm band to a disconnected state. We see that most of the RSSI variation is confined to the bands between -70 and -90 dBm, e.g. red and green, for those packets that were received.

Figure 12 gives a clearer view of the RSSI variation with time, showing the time-alligned plots of RSSI vs. time and packet loss vs. time for a 100-second snapshot of the 6 minute trace. In particular, Figure 12(a) shows just RSSI as a function of time. We see that RSSI is largely varying between -70 and -90 dBm. Moreover, we observe that there is an apparent sinusoidal behavior in received signal strength. We believe that this is due to the hovering or loitering circles of the MAV. Indeed, a simple calculation using the loiter radius of 50 m, plane speed of 15 m/s, shows that a loiter circle is completed about every 50 seconds. The sinusoid in Figure 12(a) has just such a period of about 50 seconds. As another graph will show, the plane varied in distance from 20-160 m from the base station, so the hovering circle was approaching and receding from the collection point. This behavior would lead to the sinusoidal variation in RSSI.

We also observe that RSSI, while observing a general trend, varies substantially from moment to moment. For example, at t=33, RSSI dropped about -70 dBm to below



(a) RSSI vs. time for one loitering MAV



(b) Packet loss vs. time for one loitering MAV

Figure 12: This is a plot of RSSI and packet loss vs time for a single MAV hovering and communicating back to the ground station.

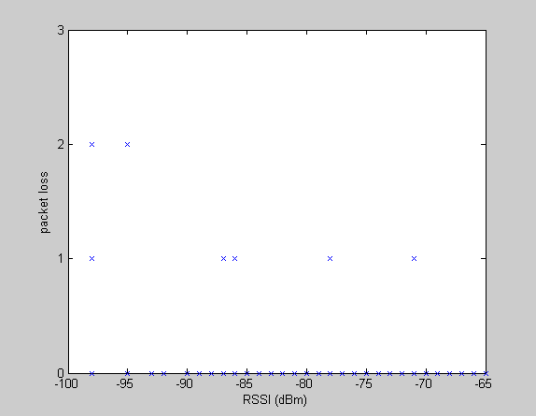


Figure 13: This is a plot of packet loss vs RSSI for 2.

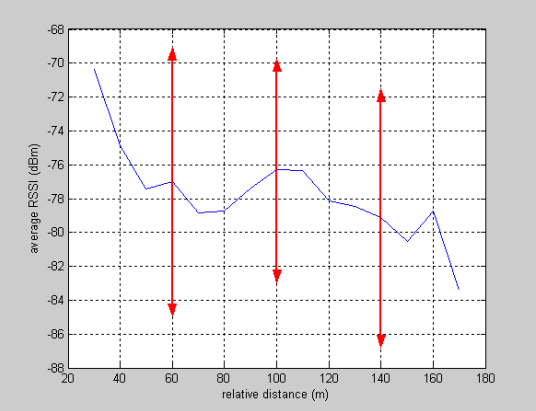


Figure 14: This is a plot of RSSI vs. distance from the single-MAV experiment.

-90 dBm in just a few seconds. This could be due to the instability of small MAV flight, which is characterized in a later graph.

Figure 12(b) shows packet loss as a function of time for the same 100-second snapshot, and can be time-alligned with the above RSSI plot. Packet losses are generally well-correlated to drop-outs in RSSI, in particular at times t=45, 80, 90, and 95. Packet losses were relatively rare in this test, with the loss of only 17 packets in this 100-second window of 1000 packets, or 1.7%.

Figure 13 shows packet loss as a function of RSSI for the one-plane experiment. As received signal strength increases, the packet loss rate decreases as expected. As the RSSI decreases, packet loss rises. This "L"-shaped curve is similar in shape to other reported results from the wireless measurements literature for static sensor networks [27].

Figure 14 shows RSSI as a function of distance for the single-MAV experiment. The distance between the MAV and the base station ranged from 20-160 m. The general trend that emerges is that RSSI falls off as expected when the distance increases. However, the decrease is not smooth or monotonically decreasing. The error bars (+/- two standard deviations) show considerable variation in RSSI around the average. We believe the mobility of the MAVs strongly influenced this large spread in RSSI around a given distance. In particular, we believe that the many different orientations of the plane, and hence the MAV's antenna, as it passes in and out of a particular range from the ground station cause a large variation in the RSSI. This should accounted for in the design of MAV network protocols.

To understand the impact of banking and rolling on the RSSI, we illustrate in Figure 15 the roll angle as the MAV flies along its trajectory. We would expect that a small plane such as the MAV would exhibit considerable instability. The antenna is vertical, so slight changes in the roll angle will cause considerable change in the antenna orientation. As in the RSSI trajectory, our goal here is to assess the rapidity of change of the parameter of interest, namely the roll angle. We labeled the following color bands: black corresponds to a roll angle of +/- 10 degrees; red corresponds to +10 to +20 or -10 to -20; green corresponds to +20 to +40 or -20 to -40; and gray is any roll angle whose magnitude is greater than 40 degrees.

We observe that there is frequent change in the roll angle, as the colors are intermingled and there are few continuous bands of color, unlike the RSSI trajectory of Figure 11. All three viewpoints confirm that the roll angle changes quite frequently, almost every second. This confirms that the MAV is indeed wobbling frequently during flight. This provides support for our belief that Figure 14's large spread in RSSI around distance is due to many changing MAV orientations. This may also explain the considerable variation in RSSI versus time shown in Figure 12.

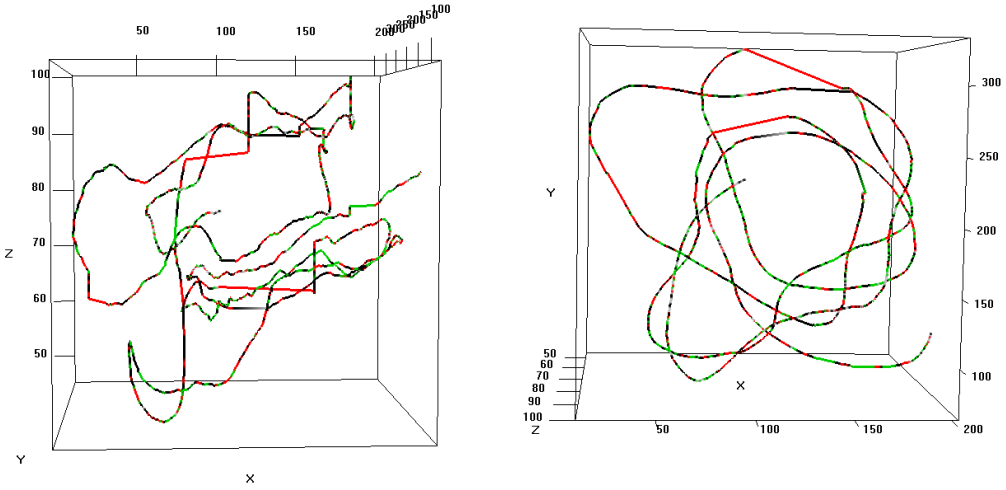
Summarizing the experimental results of our one-plane tests in this section, RSSI varies in a relatively continuous manner in flight when viewed via suitably coarse quantization bands (Figure 11), but this masks the considerable instantaneous variation in RSSI (Figure 12). The overall trend in RSSI is to drop off as expected as a function of distance (Figure 14), but the average behavior is obscured by substantial variation in RSSI (error bars in Figure 14 and instantaneous variation in RSSI of Figure 12). This is likely due to the constant rolling of the MAV plane (Figure 15).

6.2 Two-Plane Tests

For our two-plane tests, we constructed a scenario where plane #1 is streaming numbered packets to plane #2, who is then relaying the packets back to the ground station. Plane #2 appends its own packets to plane #1's relayed packets, using its own sequence number. The packet format is similar to the earlier described format. Both planes are hovering in loiter circles. Plane #2 is located near the base station, circling around it, while plane #1 is loitering further away.

Figure 16 shows the trajectories of both MAVs simultaneously flying during the 2-MAV flight test. The blue-colored MAV 2 circles the base station, while the red-colored MAV 1 circles further away. The loiter radiuses were set to 50 m. The loiter circles are approximately shown via the top-down view of Figure 16(b). As can be seen, there is overlap between the two loitering patterns. The altitude of both flights reaches a maximum of 80 m.

RSSI variation as a function of time between plane #1 transmitting and plane #2 receiving is shown in Figure 17. Note that both endpoints are now hovering. In particular, Figure 17(a) shows that the RSSI variation has increased compared to same RSSI plot vs. time for the one-plane test in Figure 12(a). There also appears to be a similar

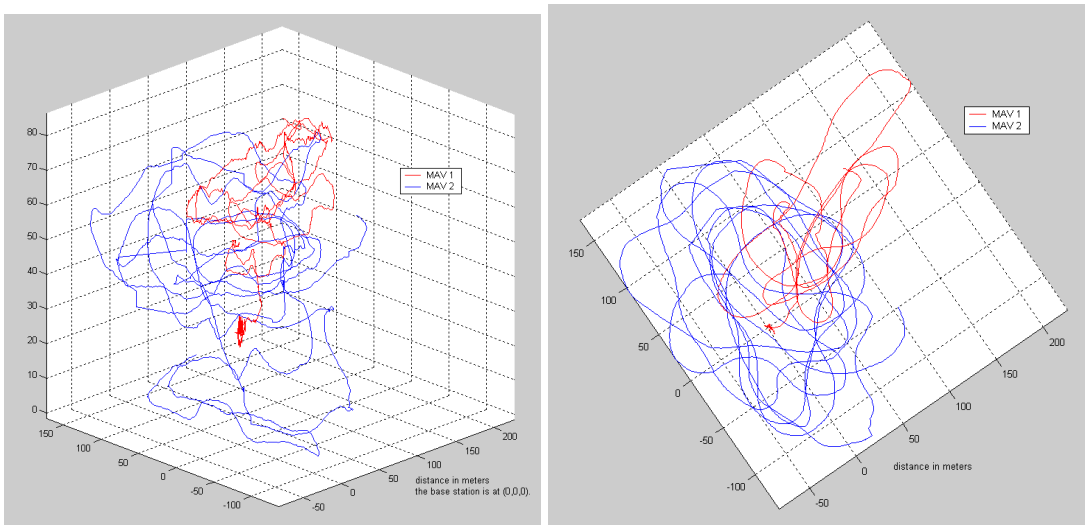


(a) Front view of loiter circle/hovering

(b) Top view

(c) Side view

Figure 15: Roll angle as a function of the loiter circle trajectory of a MAV.



(a) Trajectory of both MAVs. (b) Trajectory of both MAVs (top)

Figure 16: Trajectory of 2-MAV flight test.

average sinusoidal pattern, which is hard to see. Due to the hovering behavior of both MAVs, which are hovering at the same speed and the same radius, we would expect to see some type of sinusoidal behavior. In our next version of the paper, we seek to clarify the average and the period of this seemingly cyclical behavior.

Figure 17(b) shows packet loss alligned in time for the same 100-second snapshot as the RSSI above. Again we observe that the packet losses are correlated with drops in RSSI. In comparison with the packet loss statistics of the one-plane experiment, we see more severe packet-loss in the air-to-air scenario. The two MAVs were separated on average the same distance as the MAV-to-ground experiments. Thus, over approximately the same separation in distance, the packet losses of the MAV-to-MAV communication were greater than the MAV-to-ground losses. This is possibly due to the increased instability of both MAVs.

Figure 18 shows packet loss as a function of RSSI. Again we see the familiar "L"-shaped pattern. The losses are greater and more spread when there is a weak received signal.

Figure 19 shows RSSI as a function of distance for the two-MAV experiment. We see that the average received signal for MAV-to-MAV communication is lower on average by about 6 dBm than for MAV-to-ground communication over the same distance. The variation is about comparable between this Figure and Figure 14. We expected to see more of a spread in 2-MAV communication. However, this may be due to increased packet loss, since the spread only counts packets that have correctly received.

6.3 Five-Plane Tests

Figure 20 shows 5 MAVs that we have built as part of our SensorFlock. We were eager to deploy a flight test of 5 MAVs. The plan was to use 3 human controllers to guide the 5 in-flight MAVs. However, we were sidelined by inclement weather throughout the week prior to this paper submission. It snowed twice in Boulder during this week, and the other days were too windy and/or too cold to fly. We will be carrying out 5-plane full flight experiments, whose results will be posted at [25]. We will also be issuing a technical report containing the information from these flights.

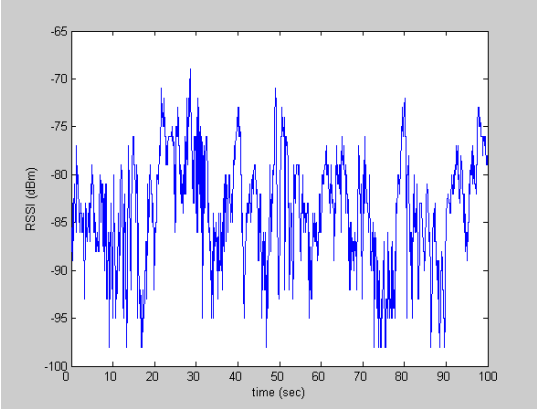
We had designed a 5-plane experiment in which packets were periodically flooded throughout the network as the most thorough way to test the connectivity of all links. Nodes would use the same flooding mechanism as a primitive means of routing back to the ground station. As before, RSSI and packet loss statistics would be collected. In addition, multi-hop path information was to be gathered.

As part of future work on RF characterization, we also intend to test different antenna orientations, flight patterns and speeds. We also intend to capture more precise instantaneous roll and heading information from the control software directly, rather than inferring this after the experiment.

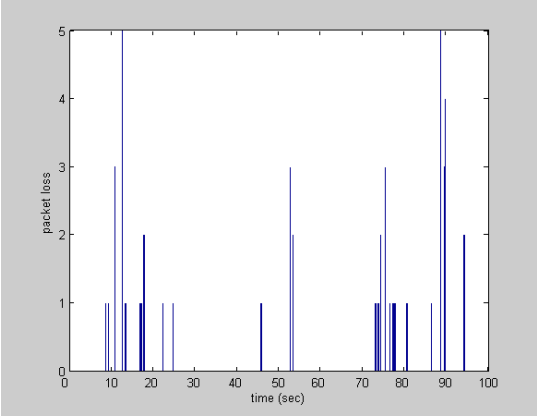
One of the most difficult aspects is achieving fully autonomous flight. In many cases, we needed an experienced RC pilot to handle severe situations where the MAV was buffeted by severe winds. Thus, while the SensorFlock system is designed to scale, we must still be cognizant that some measure of human intervention in the form of RC-based pilot backup is needed. As our experience with the SensorFlock grows, we expect to streamline the ratio of human controllers to MAV planes.

7. CONCLUSIONS

We have presented SensorFlock, a mobile system for networked micro-air vehicles. We have described the design of MAVs, including the avionics, the flight control software, and the launch system. We have shown the capability to perform controlled loiter-circle hovering of MAVs. We have provided a thorough characterization of the RSSI performance of both MAV-to-MAV and MAV-to-ground communication. Summarizing our experimental results, RSSI varied in a relatively continuous manner in flight when viewed via suitably coarse quantization bands, but this masked the considerable instantaneous variation in RSSI. The overall



(a) RSSI vs. time for plane 1



(b) Packet loss vs. time for plane 1

Figure 17: This is a time-alligned plot of RSSI and packet loss vs time for the two-MAV experiment, plane 1 sending to plane 2, who relays the results to the ground station. Both MAVs are loitering in circles.

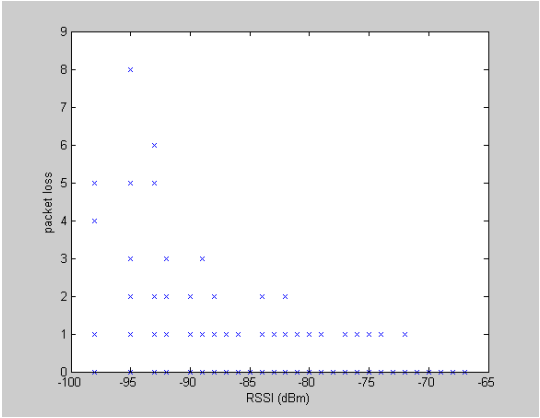


Figure 18: This is a plot of packet loss vs RSSI for 1.

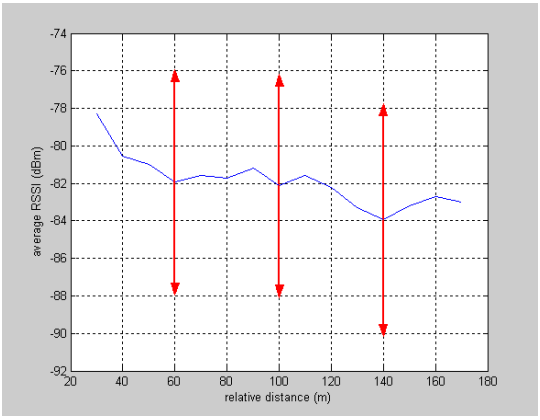


Figure 19: This is a plot of RSSI vs. distance from the two-MAV experiment.

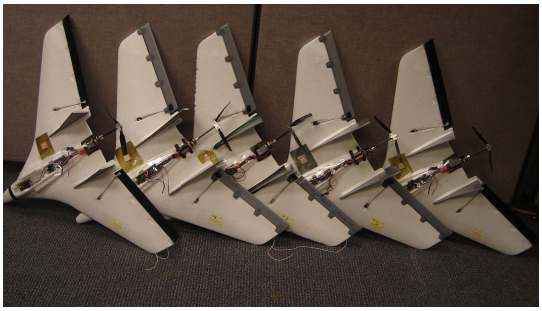


Figure 20: Our first SensorFlock of 5 MAVs. The design of each MAV is relatively simple, so scaling to produce larger flocks can be quickly achieved.

trend in RSSI is to drop off as expected as a function of distance, but the average behavior is obscured by substantial variation in RSSI. This is likely due to the constant rolling of the MAV plane. MAV-to-MAV communication showed even more variation in RSSI and greater packet loss over the same distances than MAV-to-ground communication.

8. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, August 2004.

[2] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *ACM MobiCom*, 2005.

[3] B. Bellur, M. Lewis, and F. Templin. Tactical information operations for autonomous teams of unmanned aerial vehicles (UAVs). In *IEEE Aerospace Conference Proceedings*, volume 6, pages 2741–2756, 2002.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM MobiCom*, 2005.

[5] T. Brown, B. Argrow, S. D. R.-G. Thekkekunnel, and D. Henkel. Ad hoc uav ground network (augnet). In *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conference*, Chicago, IL, September 2004.

[6] T. Brown, S. Doshi, S. Jadhav, and J. Himmelstein. Test bed for a wireless network on small uavs. In *Proc. AIAA 3rd "Unmanned Unlimited" Technical Conference*, Chicago, IL, September 2004.

[7] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In *ACM MobiCom*, 2006.

[8] K. Chebrolu, B. Raman, and S. Sen. Long-distance 802.11b links: Performance measurements and experience. In *ACM MobiCom*, 2006.

[9] Y.-C. Cheng, J. Bellardo, P. Benko, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *ACM SIGCOMM*, 2006.

[10] ACM E-WIND (Experimental approaches to wireless network design and analysis) Workshop. http://acm.org/sigcomm/sigcomm2005/w1-e-wind.html.

[11] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. In *UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013*, February 2002.

[12] M. Gerla, Y. Yi, and K. Xu. Team communication among airborne agents. In *Proc. Of American Helicopter Society 59th Annual Forum*, pages 163–167, 2003.

[13] J. Grasmeyer and M. Keennon. Development of the Black Widow micro air vehicle. AIAA paper 2001-0127, Reno, NV, 2001. 39th AIAA Aerospace Sciences Meeting & Exhibit.

[14] D. Henkel, C. Dixon, J. Elston, and T. Brown. A reliable sensor data collection network using unmanned aircraft. In *Second International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN)*, Florence, May 2006.

[15] P. Ifju, D. Jenkins, S. Ettinger, Y. Lian, W. Shyy, and M. Waszak. Flexible-wing-based micro air vehicles. AIAA paper 2002-0705, American Institute of Aeronautics and Astronautics, January 2002. 40^{th} Aerospace Sciences Meeting & Exhibit, Reno, Nevada.

[16] M. Kovacina, D. Palmer, G. Yang, and R. Vaidyanathan. Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles. In *Proc. IEEE/RSJ Conf. Intelligent Robots and Systems*, pages 2782–2788, Lausanne, Switzerland, October 2002.

[17] D. Lawrence, E. Frew, and W. Pisano. Lyapunov vector fields for autonomous uav flight control. To be submitted for publication in AIAA Journal of Controls, 2006-2007.

[18] D. Lawrence, K. Mohseni, and R. Han. Information energy for sensor-reactive uav flock control. In *3rd AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibit*, September 2004. AIAA paper 2004-6530.

[19] G. Lu, B. Krishnamachari, and C. Raghavendra. Performance evaluation of the ieee 802.15.4 mac for low-rate low-power wireless networks. In *Workshop on Energy-Efficient Wireless Communications and Networks (EWCN '04), held in conjunction with the IEEE International Performance Computing and Communications Conference (IPCCC)*, April 2004.

[20] M. Petrova, J. Riihijrvi, P. Mhnen, and S. Labella. Performance study of ieee 802.15.4 using measurements and simulations. In *Proceedings of IEEE WCNC*, April 2006.

[21] W. Pisano, D. Lawrence, and K. Mohseni. Concentration gradient and information energy for decentralized uav control. In *AIAA-2006-6459, AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, August 2006.

[22] A. Woo and D. Culler. Taming the challenges of reliable multihop routing in sensor networks. In *ACM SenSys*, November 2003.

[23] K. Xu, X. Hong, and M. Gerla. Landmark routing in ad hoc networks with mobile backbones. *Journal of Parallel and Distributed Computing (JPDC), Special Issues on Ad Hoc Networks*, 2002.

[24] K. Xu, X. Hong, M. Gerla, H. Ly, and D. Gu. Landmark routing in large wireless battlefield networks using UAVs. In *IEEE MILCOM*, pages 230–234, 2001.

[25] MAV SensorFlock experiments. http://www.cs.colorado.edu/ rhan/temp/MAV.pdf.

[26] R Project for Statistical Computing. http://www.r-project.org.

[27] Y. J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, 2003.

[28] J. Zheng and M. J. Lee. *A comprehensive performance study of IEEE 802.15.4*. IEEE Press, Wiley Interscience, 2006.

[29] G. Zhou, T. He, S. Krishnamurthy, and J. Stankovic. Impact of radio irregularity on wireless sensor networks. In *ACM Mobisys*, June 2004.