Cooperative Flight Guidance of Autonomous Unmanned Aerial Vehicles

William Etter Paul Martin Rahul Mangharam
Department of Electrical and System Engineering
University of Pennsylvania
{etterw, pdmartin, rahulm}@seas.upenn.edu

***Abstract*—As robotic platforms and unmanned aerial vehicles (UAVs) increase in sophistication and complexity, the ability to determine the spatial orientation and placement of the platform in real time (localization) becomes an important issue. Detecting and extracting locations of objects, barriers, and openings is required to ensure the overall effectiveness of the device. Current methods to achieve localization for UAVs require expensive external equipment and limit the overall applicable range of the platform. The system described herein incorporates leader-follower unmanned aerial vehicles using vision processing, radio-frequency data transmission, and additional sensors to achieve flocking behavior. This system targets search and rescue environments, employing controls, vision processing, and embedded systems to allow for easy deployment of multiple quadrotor UAVs while requiring the control of only one. The system demonstrates a relative localization scheme for UAVs in a leader-follower configuration, allowing for predictive maneuvers including path following and estimation of the lead UAV in situations of limited or no line-of-sight.** [1]

I. Introduction

Unmanned aerial vehicles have recently become a viable platform for surveillance and exploration tasks where human presence is dangerous, impossible, or inadequate [1]. Several commercial quadrotor aircraft (a popular four-rotor vertical take-off and landing (VTOL) vehicle) have been successfully used as surveillance equipment by groups such as United States and Canadian police forces [2], and it is not difficult to imagine other applications for this burgeoning technology—exploration of radioactive/hazmat environments, naval search and rescue (SAR), or surveying a building on fire, to name a few. Despite the agility and speed of the quadrotor platform, current implementations of the VTOL configuration do not allow for extended flight times (usually upwards of twenty minutes) [2]. This short flight time limits the overall range as well as the amount of time available to thoroughly explore a specific area. To combat this drawback to quadrotor platforms, a system for intelligently controlling multiple quadrotor UAVs for concurrent exploration is proposed. This system uses a lead, human-controlled quadrotor and one or more follower quadrotors that track the lead unit autonomously. This system aims to improve the execution time required to complete missions and increase breadth of search and platform effectiveness.

When dispatching quadrotor units to a location, the challenge of determining the platform's placement in space with respect to surrounding objects and other platforms is increasingly important as the simplicity and familiarity of the environment diminish. Methods currently used require pre-installed sensor systems or a base station, limiting both range and effectiveness in a wide array of surroundings.

A cost-effective and manageable solution to this requirement is to use a single human operator to conduct the advanced requirements of simultaneous localization and mapping (SLAM), allowing any quadrotor(s) to effectively follow the human-controlled unit through the environment [3]. That is, the human operator is charged with the responsibility of determining the absolute location of the lead quadrotor at any given time. Thus, instead of performing absolute localization, quadrotors using this system will assume that the human operator is aware of his/her surroundings and follow the clear flight-path generated, localizing itself only with respect to the quadrotor designated as its leader. When the desired location is reached, the collection of environmental data can be achieved with a greater number of vehicles, decreasing mission time. Overall, this system eliminates the requirement of standard localization equipment or a human operator for each quadrotor, effectively managing resources during a critical mission and adding redundancy and robustness to an otherwise singular system.

A. Application Scenarios

It has been mentioned that quadrotor systems offer a viable platform for surveillance and search and rescue for a wide

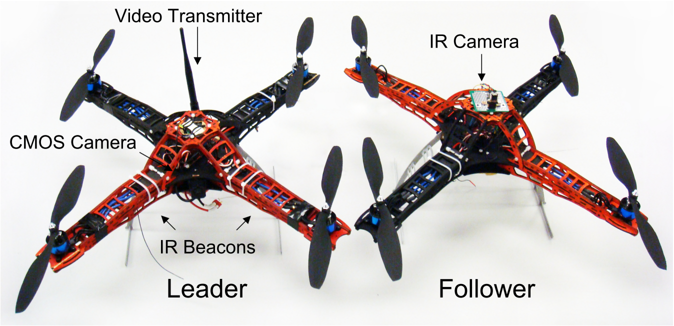

Figure 1. Experimental Quadrotor Platform (Leader and Follower Units)

[1] This research work was supported in part by the NSF MRI-0923518 grant.

variety of environments. A specific example can be seen in the potential for adopting cutting-edge target detection software. The Sentient Kestrel, for example, has recently been adapted to detect life jackets in a moving body of water from very far distances [4]. A quadrotor platform with adequate processing power could easily adopt such software to allow for quick deployment during time-critical operations.

In addition to search and rescue, quadrotor helicopters can be used for a variety of other collaborative tasks explored briefly here. Of great importance to the success of the system described herein, the low cost and small form factor of quadrotor helicopters allow for expandability to flocks of arbitrary size. In such a system, computations can be distributed across multiple platforms, decreasing processing time by operating as a parallel computing platform. This notion of parallel computing is closely tied to the equally important concept of redundancy mentioned previously, allowing for additional failsafe measures.

Finally, multiple and distributed quadrotor platforms create the potential for a dynamic wireless routing layer. That is, the system need not behave as a point-to-point network and can instead benefit from the increased throughput and extended range of a mesh network. And indeed, a leader-follower flocking system is almost ideally situated for use as a mesh network if one's aim is to increase effective range.

The effort presented here focuses on scalable leader-follower swarm systems and parallel autonomous search and rescue. This is accomplished via the development of a platform for autonomous aerial vehicles for SAR. The remainder of the paper is organized such that the system requirements and design are covered first and then are supported by the actual implementation and results. Section 2 begins with a survey of related work, Section 3 discusses the system/subsystem design requirements and choices, and Section 4 covers the system model completed in MATLAB. Sections 5, 6, and 7 provide details of the actual implementation for the system platform, communication and networking, and sensor fusion respectively. Section 8 contains data obtained from the model and sensors, and Sections 9 and 10 discuss the overall system design and expandability.

II. Related Work

Quadrotor platforms have found increasing use in academic settings. The maneuverability and easy access to the center of mass make such systems easily expandable with sensor arrays, microprocessors, and wireless communication hardware. The University of Pennsylvania's GRASP Laboratory has developed advanced controls for quadrotor platforms using a Vicon 3D motion capture camera system [5]. This system uses multiple stationary cameras to determine in real time the location of one or more quadrotor units. The accuracy and speed of the Vicon 3D system has allowed the GRASP laboratory to perform multiple flips, inverted perching, flying through hoops, and collaborative grasping. While this is a robust and accurate system for localization of UAVs and has allowed Penn's GRASP Lab to develop very intricate and innovative controls, it is limited to operation within a room where a motion capture system has been installed. Students at ETH Zurich have developed the PixHawk quadrotor—a system that uses cameras for SLAM as well as detection of objects. The PixHawk uses a downward-facing camera to track fiducials distributed on a mat below the unit. This method of SLAM offers potential in multi-quadrotor systems, however the PixHawk does not currently extend their vision detection to relative localization between quadrotors and movement is limited to the region above the mat [6]. The University of Essex has developed swarMAV, a multiple micro air vehicle (MAV) system designed for collaborative flocking and problem solving. To solve the problem of localization, the swarMAV team uses a separate base station unit with infrared ranging accurate to within 1 millimeter [7]. This, however, presents the same limitation inherent in the setup used by Penn's GRASP lab—operation is limited to within range of the base station.

Stanford's STARMAC quadrotor is not as strictly confined to a location as are the previously mentioned systems—Hoffmann et. al. use a commercial Draganflyer X4 platform with a GPS sensor to be able to command the aircraft through a sequence of waypoints [8]. Position data from GPS sensors have allowed Stanford and several other institutes to successfully fly autonomously in outdoor environments, but commercial GPS sensors often lack fast update rates and overall accuracy, preventing their use in close-quarter (or indoor) multi-unit swarms.

A third and growingly-popular technique for localization is by use of a scanning laser rangefinder. Research including [9] has successfully used laser rangefinders to map surroundings in 3D and respond to obstructions accordingly. This is a convenient and accurate, although computationally expensive, method of autonomy, but it is unnecessary if not cumbersome in determining relative location of units in a multi-unit system.

Wang et. al., noting that many natural locomotive systems as seen in honeybees and fish contain distinct leaders and followers, designed a ground robot swarm based on a consensus algorithm. As in [5], these robots receive positional feedback from a Vicon system and thus are limited in applicable range [10]. The same concept of leader-follower flocking can be easily applied to quadrotor systems, but in designing a system with practical applications it is necessary to determine relative locations using scalable, mobile hardware.

III. System Architecture

The proposed leader-follower system can be divided into three main subsystems. The first is the group of physical nodes interacting in the system (Figure 1). The second is the network that provides a communication layer between the nodes. The last subsystem is the sensor data acquisition and fusion to generate mission-related information. These subsystems are connected as shown in Figure 2.

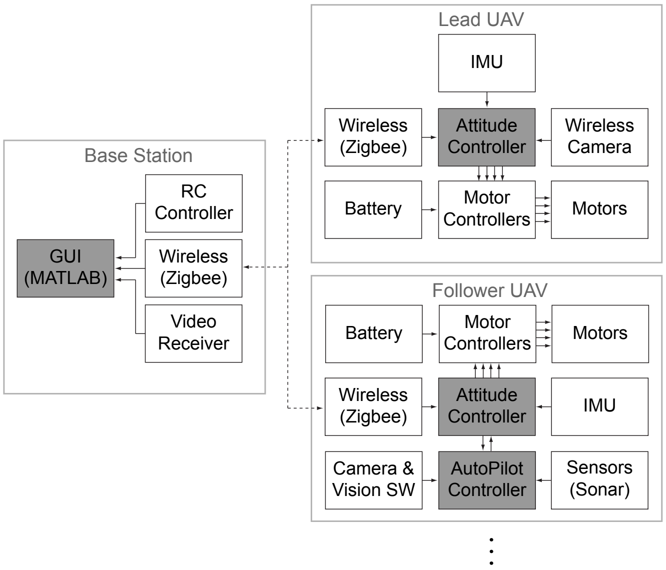

Figure 2. System overview showing a base station, leader, and one follower. Processing units are highlighted in gray, solid lines represent physical wire connections, and dotted lines represent wireless connections

A. *Node-Specific Architecture*

This subsystem is composed of three node types. The first is the lead unit, which is a human-controlled UAV that receives flight commands from a remote control (RC) transmitter. By flying this quadrotor the user effectively completes the calculation- and hardware-intensive process of localization, avoiding obstacles and maneuvering in ways that could only be achieved through the use of expensive and computationally powerful sensor systems. The lead unit has three critical components installed to assist in guidance, tracking, and data acquisition for itself and following units. First is a camera and video transmitter package. This provides a live feed of the quadrotor surroundings, allowing the system user to remotely operate the UAV even during times of limited or no line-of-sight (LOS). Next is a wireless transceiver with networking capability. This provides a link for data communication between the system nodes. Lastly, a pair of wide-angle 850 nm near-infrared (hereafter referred to as just infrared or IR) beacons are attached to the quadrotor. This provides a visual data source for flight tracking.

The second node is composed of the follower quadrotor(s). These units are autonomous in that they are not controlled directly by a human operator. In addition to a wireless transceiver, a visual tracking system composed of a near-infrared IR camera and microcontroller dedicated to sensor data is installed. This vision system is designed to detect the IR light emitted from the beacons installed on the lead quadrotor and provide IR point data. Additional sensors are also installed on the follower units to provide automated flight controls (such as obstacle avoidance) as well as environmental data of the mission zone.

The last node is the base station. This is made up of the real-time video display from the leader onboard camera as well as a graphical user interface (GUI) providing the user information regarding the system; this node is not required for system operation. The data provided by the GUI includes the current attitude information of the quadrotors, battery status, as well as any pertinent environmental information depending on the sensors installed on the quadrotors. This allows the user to quickly assess the system and mission sector being explored.

B. *Network Architecture*

The system network architecture is designed such that the individual components are insusceptible to common sources of interference and maintain a level of robustness against data loss. That is, the three main data communication channels (the remote control for the leader, the video transmission from the leader, and the inter-node data) are separated in channel and/or frequency. The RC and video transmissions are directional point-to-point, however the primary data setup is an ad-hoc network between the nodes. This provides system stability as the network is decentralized and not dependent on the operation of any individual node. This setup also allows for more complex configurations, such as packet routing back to the base station in situations when certain nodes are out of range.

C. *Sensor Fusion*

An integral component of the quadrotor leader-follower system is the fusion of data from the separate sensors on the dedicated microcontroller. This provides two main data types—node tracking and environmental data. The node tracking data is created from the combination of the IR camera and inertial measurement unit (IMU) data from each leader/follower pair. That is, the IR point data from the camera is transformed using the attitude data of both the follower and leader nodes. This provides the orientation-corrected IR point-width information which relates the relative spatial positioning of the quadrotors. The position of the IR points in relation to the major axes provides vertical and horizontal information (up/down and left/right) while the fixed width between the IR points provides the distance information (forward/backward). In this way, the data from three sensors (one of which is physically separated from the others) is combined to generate the three-dimensional tracking data required for flight following.

Environmental data is also processed by the sensor microcontroller. The range data from the ultrasonic sonar sensors directed along each axis of the quadrotor provides distance data to obstacles. When processed, this data provides flight feedback to prevent the nodes from maneuvering into blocked locations. Additionally, the suite of sensors installed on the nodes provide collaborative information to system users. For example when combined with the sonar data from multiple nodes the information from a thermopile array can provide location information for sources of heat such as people

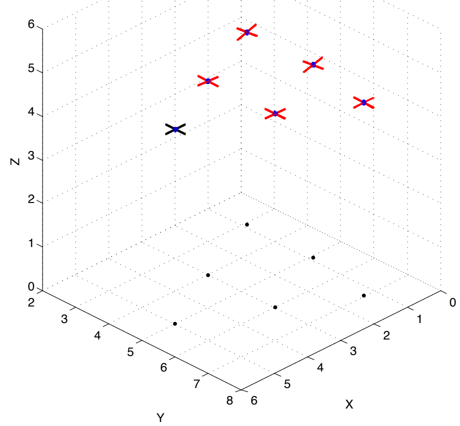

Figure 3. Recursively defined flocking model in MATLAB, showing 2 followers following one leader, and three followers following those two.

or fires and atmospheric sensors (such as carbon-monoxide sensors) can map locations presenting hazards. This results in the system assisting rescuers with mission-critical data that increases their effectiveness and efficiency.

IV. System Model

Before implementing vision tracking algorithms on an autonomous UAV with no user input, it is useful to model the system in software. Towards this end, a linear MATLAB model is used to simulate quadrotor physics, object-tracking cameras, RF communication, and proportional–integral–derivative (PID) controls. The model simulates individual quadrotor objects as a point mass, allowing easy creation of additional quadrotor objects along with arbitrary definition of leader-to-follower coordinates and thus an arbitrary flock formation as shown in Figure 3.

A. Controlling the Leader

The lead quadrotor is controlled by a user-provided event array, indicating desired attitude and thrust for a specified time. This event script, shown in Equation 1, simulates an RC controller and allows the trajectory of the lead quadrotor to be changed easily and realistically. The attitude is specified using Euler angles, with ψ, θ, and ϕ as yaw, pitch, and roll, respectively.

$$events = \begin{bmatrix} t_1 & thrust_1 & \psi_1 & \theta_1 & \phi_1 \\ t_2 & thrust_2 & \psi_2 & \theta_2 & \phi_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_n & thrust_n & \psi_n & \theta_n & \phi_n \end{bmatrix} \tag{1}$$

B. Virtual PID

To achieve the desired attitude commanded in the matrix in Equation 1, a virtual PID controller is used. With a time step δt of 0.1 seconds, the values for K_P, K_I, and K_D that resulted in a realistic transient response (decidedly 1 second rise time with less than 10% overshoot) were empirically determined to be 20, 0, and 0.6. These gains mold the response of the angular accelerations, $\dot{p}$, $\dot{q}$, and $\dot{r}$, which are then integrated to determine the angular velocities and the angles themselves.

C. Virtual Camera

To accurately simulate the conditions faced when implementing this system on physical hardware, it is useful to limit the information available to the follower quadrotor(s). To simulate an on-board camera, the leader's coordinates in world space are transformed via rotation about the unit's angles, as shown in Equation 2, where x_L and x_F indicate the x coordinate of the leader and follower, respectively.

$$\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \times \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \times \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L - x_F \\ y_L - x_F \\ z_L - x_F \end{bmatrix} \tag{2}$$

By simulating a virtual camera with the linear transformation given in Equation 2 we can determine whether or not a given quadrotor is in the field of view (FOV) for any of the follower quadrotors. This is trivial, and assuming a camera with aspect ratio 1:1 can be determined by Equation 3, where x is the forward-facing axis, y is the right-facing axis, and z is the up-facing axis.

$$\theta_{object} = tan^{-1}\frac{2z_{cam}}{x_{cam}} = tan^{-1}\frac{2y_{cam}}{x_{cam}} \tag{3}$$

By defining θ_{object} as the angle that the object seen by the camera makes with the forward-facing vector of the follower, we can determine if the object is within the camera's FOV. This allows the model to simulate the leader-follower tracking algorithm for various FOV values as well as handle cases in which line-of-sight is lost.

D. Follower Autonomy

Each follower quadrotor is given specific following coordinates with respect to the leader. In order to achieve these desired coordinates, a second PID loop is used on top of the stability PID. Gains are empirically chosen first to match altitude (z) by commanding a thrust and then to match x and y by commanding a pitch and a roll. Information about the leader yaw is not parsed in the camera transform, nor will it be on the experimental platform. Instead, the model creates a virtual RF link between the leader and the follower(s), allowing the follower access to the yaw information of the leader with some pseudo-random noise.

Values used for the tracking PID are: $K_P = 23$, $K_I = 0$, $K_D = 18$, $K_{P-THRUST} = 100$, $K_{I-THRUST} = 4$, and $K_{D-THRUST} = 6$. A non-zero $K_{I-THRUST}$ is needed to offset steady state error in altitude matching. A timestamped visualization of the model is shown in Figure 4.

V. Experimental Platform

The UAV platform used in this system is a custom-built quadrotor helicopter based on a commercial frame. Constructed of aircraft laminate and fiberglass, the frame is lightweight to allow for extended flight times and has an open design for sensor and equipment installation. The flight control system is shared by both the lead and follower quadrotor units. The system is based on an ATmega1280 microcontroller running at 16MHz and is responsible for stability and motor control. The microcontroller is pre-installed on a PCB designed specifically for aircraft systems with RC input/output. The main process on this microcontroller is the PID attitude controller that maintains the quadrotor at the desired angles. Communicating with this flight microcontroller over serial at a baud rate of 115,200 is a CHRobotics attitude and heading reference system (AHRS) IMU. This device uses a 32-bit ARM Cortex microprocessor to read the values from 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer sensors. After running an Extended Kalman Filter (EKF), the IMU outputs the absolute yaw, pitch, and roll Euler angles in reference to magnetic North and the horizontal plane as well as the rotational rates and accelerations. Also connected to the flight microcontroller is an XBee Pro wireless transceiver to transmit data between the units.

There are two types of vision systems installed on the units. On the leader is a CMOS camera connected to an 800mW video transmitter. The follower tracks IR beacons created using 850nm light emitting diodes (LEDs) arranged in a radiating pattern for the greatest viewing angle. This provides system accuracy even when the lead unit pitches/rolls. On the follower units is an IR camera connected to an Mbed microcontroller running at 100MHz. This microcontroller is used to collect data from the various sensors on the units and send the information back to the ATmega1280, where it is then used for flight or relayed to another unit.

The base station node consists of a portable LCD monitor and MATLAB GUI. The LCD monitor is attached to the video receiver and a battery pack, providing a live view for the user. The MATLAB GUI obtains data using an XBee Pro wireless transceiver attached through a USB converter and displays relevant system status information and environmental data.

The current version of the system has a total weight of approximately 900g for the lead quadrotor and 800g for the follower quadrotors with 2200mAh lithium-polymer batteries installed. Further details of the implementation and flight experiments may be found at the project website http://airhacks.org.

VI. Communication and Protocols

The platform created has three distinct wireless communication links. First, a 2.4 GHz commercial RC controller is used to command a total thrust as well as desired yaw, pitch, and roll angles. Second, a 1.2 GHz commercial wireless video camera and transmitter is used on the lead quadrotor unit alone for the purpose of maintaining a first-person point of view when flying at distance. Finally, a 900 MHz XBee Pro Module (IEEE 802.15.4) provides the communication link between quadrotor units and between each unit and the base station. Each wireless link operates at a different band to minimize interference, and the ZigBee communication layer was chosen to operate at 900 MHz to increase the wireless range of critical mission data. Broadcasting at 50mW, this communication link has a maximum LOS range of approximately 9,000 meters which provides adequate range in more constricted environments.

The 900 MHz ZigBee link is set up in API mode, allowing data packets to be sent to specific addresses and not as a broadcast. This link communicates using two distinct packet structures—one to relay attitude information to followers, providing necessary information needed to complete the tracking algorithms discussed, and one to relay (at a slower frequency) battery level, the relative locations of swarm units with respect to the leader, and any pertinent environmental data collected. The former transmits at roughly 50 Hz, while the latter transmits at 2 Hz.

VII. Sensor Fusion

A. Tracking Sensor Data

In addition to the flight control system installed on the units, at the core of the leader-follower system is the vision tracking system. On the lead quadrotor the CMOS camera is only used by the user for flight guidance (there is no input to the microcontrollers). However, the vision system on the follower quadrotor is essential to determining the spatial orientation and placement of the quadrotors while in flight. The sensor used on the follower units is a PixArt IR camera. This camera communicates with the sensor microcontroller on an I^2C (inter-integrated circuit) bus and is able to detect and track the locations of up to four IR points with a resolution of 1024x768 and a refresh rate of approximately 500Hz. To ensure that the lead quadrotor does not leave the FOV when the quadrotor pitches, the IR camera is attached to a micro-servo. Using the follower's attitude information the servo is actuated to offset the pitch and maintain the IR camera at a horizontal position. The sensor microcontroller is connected to both the IMU as well as the flight microcontroller using serial communication at a baud rate of 115,200. This provides the sensor microcontroller with the IMU (attitude) data of the follower node as well as the leader (which is transmitted over the XBee Pro wireless connection and sent through the flight microcontroller). This data is then combined with the IR camera data to provide the relative position data. This is done by first determining the camera projection based off of

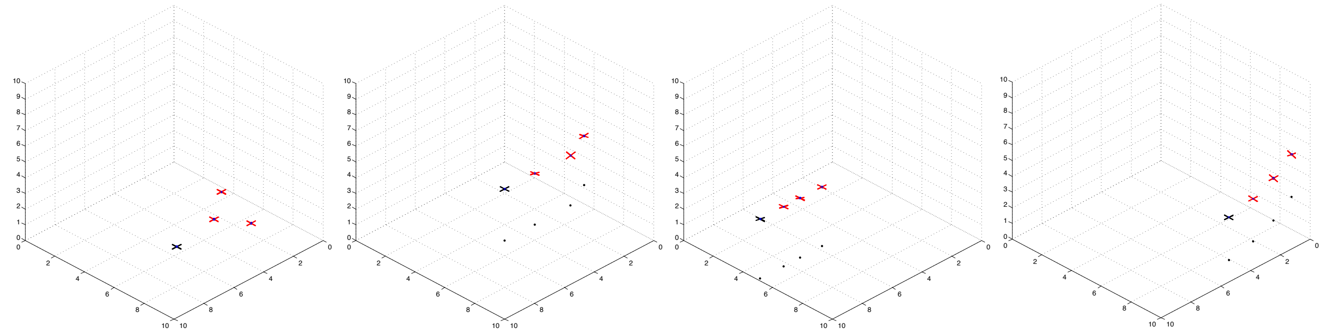

Figure 4. Snapshots of the MATLAB leader-follower model with three followers responding to initial thrust, a forward pitch of 10 degrees, a backwards pitch of 10 degrees, and then a roll left of 10 degrees. Times are (from left to right) 0, 3, 8, and 13 seconds.

the yaw, pitch, and roll IMU information from each of the units and the differences in attitude to obtain the correct view of the IR beacon points which are then projecting onto the world frame (represented by the the x-axis straight ahead of the follower and parallel to the ground, the y-axis straight out to the right of the follower and parallel to the ground, and the z-axis straight up from the follower and perpendicular to the ground). After completing this projection, the x value provides pitch control, the y value provides roll control, the z value provides thrust control, and the difference in the yaw values between leader and follower provides yaw control to follow the lead unit.

B. Environmental Sensor Data

The environmental sensor data is also combined to form more complete and effective information for the mission task. First, the ultrasonic rangefinder data is collected and aggregated to form a rough image of the area surrounding the units. When plotted in the GUI (Figure 5), this provides system users approximations to the dimensions of rooms and areas of interest. In addition, velocity data and/or GPS data (when possible) is combined with this range data to form a clearer SLAM estimate of the mission topology. This allows users and rescuers faster search times and improved situation response.

Other sensor payloads installed on the units provide a wide range of useful data to system users. For example when combined with the sonar sensor information and the SLAM data, results from a thermopile array return approximate locations of heat signatures in an area (possibly indicating people or other heat sources). When this information is combined with an atmospheric sensor (such as an airborne-particle sensor), these results can then be paired with sources of smoke that might indicate a fire. In this way additional sensor data is overlaid with existing data to form more relevant results to system users and improve mission success.

VIII. Leader-Follower Flight Experiments

Initial experiments were conducted to test the operation of the system. Out of these, two important experiments included the MATLAB simulation as well as the IR tracking experiment.

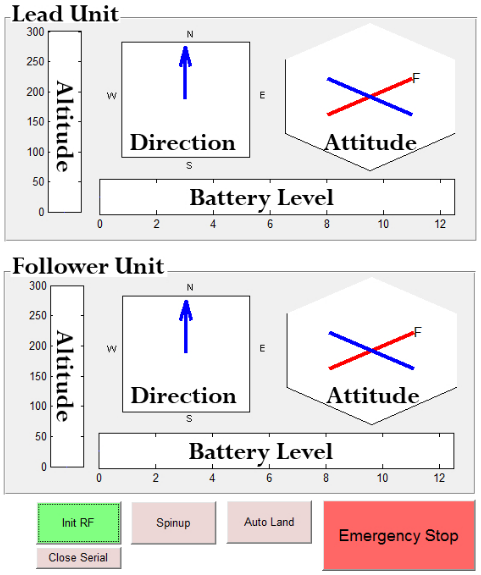

Figure 5. System GUI

These allowed the two major subsystems (the flight controls and the leader tracking) to be independently tested and verified prior to implementation and testing on the complete system.

A. MATLAB Simulations

With the MATLAB model, the system algorithms can be tested against specific real-world circumstances before they are implemented on the physical system. Specifically, potentially problematic non-idealities that exist in the real system include lost wireless packets from the leader to the follower containing IMU data, noisy IMU data, and noisy or unavailable camera data.

Noisy attitude information from the IMU will affect the stability of each individual quadrotor, but reducing or accounting for this noise is not directly tied with the success of the leader-follower algorithm used. However, the current infrared camera setup does not calculate the relative yaw of the leader and therefore must rely on accurate RF transmission of

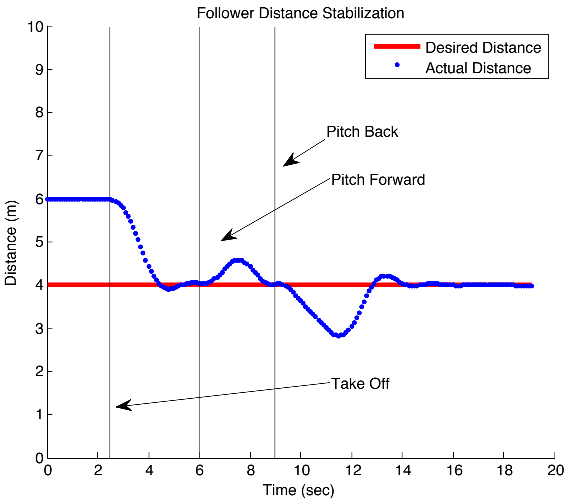

Figure 6. Accuracy of tracking algorithm along forward-facing vector with scripted ”leader” events

the leader’s yaw angle. Simulated lost packets and erroneous yaw data are lumped into a pseudo-random error between $-\alpha$ and $+\alpha$, where α is the maximum error (in degrees) in the calculation and transmission of yaw. Additionally, a pseudo-random error is introduced on all three axes of the infrared camera, allowing the user to specify in centimeters the maximum camera error.

For the following simulations, the event matrix (shown in Equation 1) is defined to give the lead quadrotor a 10° pitch forward, a 10° pitch backward, a 10° roll left, a 10° roll right, and finally a 90° yaw and 10° pitch forward. Only one follower is used, and the desired x, y, and z positions with respect to the leader are 4, 0, and 0 meters. The time step used for all simulations is $\delta t = 0.1$ seconds. Figure 6 shows the response of the follower’s tracking PID loop with the initial thrusts and pitches of the leader as defined by the event matrix.

Figure 7 shows the result of plotting versus time the total error (calculated as the sum at each time step of the instantaneous error, shown in Equation 4, where N is the total number of followers) for the cases of a ±10 degree yaw error, a ±10 centimeter camera error, and the accumulation of both errors. As shown, the success of the algorithm is affected more by a 10 cm camera error than a 10 degree yaw error, and naturally the aggregate of the two is worse still. However, though the camera sees the leader with only ±10 cm accuracy, the aggregate error over 30 seconds at $\delta t = 0.1$ intervals is only around 40 meters off of the ideal case, meaning that the worst case error is 40 meters / 10 steps = 4 meters total in all three axes, or roughly 1.3 meters in each of x, y, and z for the worst case.

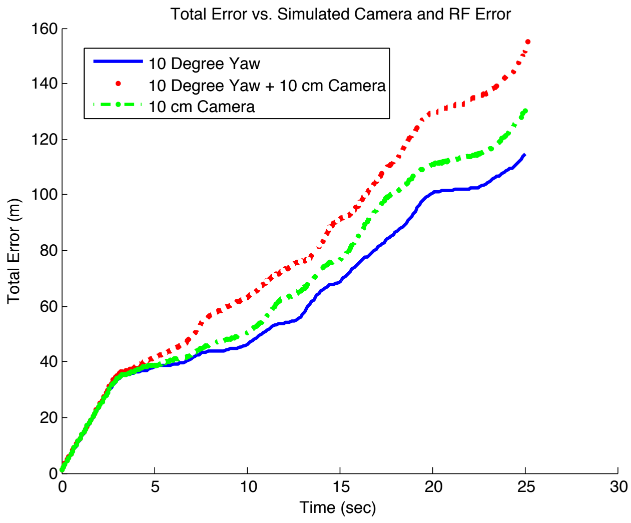

Figure 7. Accuracy (total error) vs. simulated pseudo-random errors

$$error = \sqrt{\sum_{i=1}^{i=N} [(x_{des} - x)^2 + (y_{des} - y)^2 + (z_{des} - z)^2]} \tag{4}$$

More dramatic than the effect of simulated errors on the system is the effect that the FOV of the follower’s camera has on the success of the system. For the same event matrix used for the previous test, the total error is plotted versus time for various simulated field of views. A quadrotor unit is commanded to hover if it loses sight of the leader, and it will continue following should line-of-sight return. Figure 8 shows the points at which the follower quadrotor loses sight for 20°, 30°, and 50° field of views. As shown, a follower with a 20° FOV already loses sight after the initial thrust and does not recover. A follower with a 30° FOV successfully completes thrust, pitch, and roll following (for the 10° event cases), but loses sight during the 90° yaw and does not recover. However, for a follower with a 50° FOV, the leader remains within line-of-sight. Many commercial cameras are within this FOV constraint, but lower FOV cameras may still be sufficient if the maximum yaw rate and angles are constrained.

B. Sensor Data

The PixArt IR camera was tested using multiple IR beacons moving in three-dimensional space. When running at full speed the camera was able to return IR tracking results over the I^{2}C bus at approximately 500Hz. Figure 9 illustrates a complex movement of the IR beacons in three-dimensional space (a sweeping motion away from the IR camera sensor combined with a lateral shift). This data and frequency is more than adequate for flight tracking, as this is almost twice as fast as the PID loop running on the ATmega1280 for stabilization. In addition, the FOV was experimentally determined to be approximately 45° horizontally and 32°

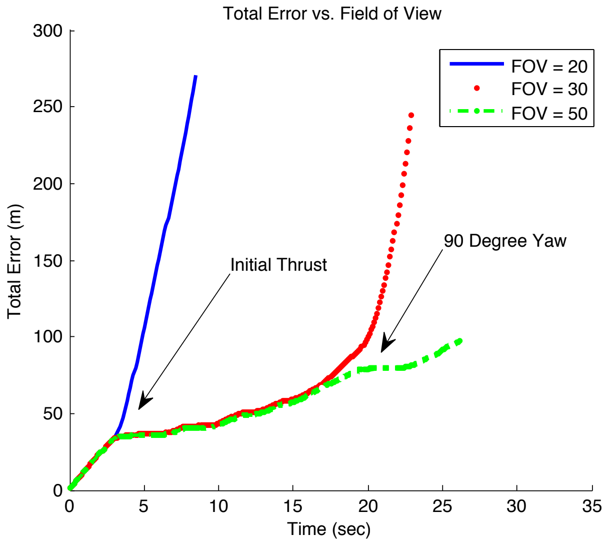

Figure 8. Accuracy (total error) vs. field of view of follower camera, given scripted "leader" events

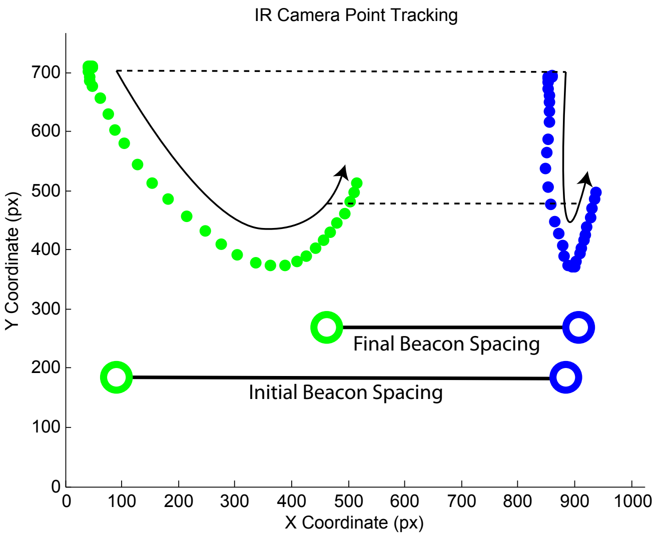

Figure 9. IR Camera Point Tracking with Simulated Unit Movement

vertically. This is close to the results found in MATLAB of a 50° FOV that resulted in limited error. However, since this is limited in comparison to the model additional changes would have to be made to improve this camera's effectiveness. This could include increasing the following distance as well as mounting the camera on a stabilizing mount (in this case a micro-servo supplied with controls based on the platform attitude) to offset changes in pitch.

IX. Discussion

A. System Limitations

Due to extensive system reliance on infrared sensors, there are limitations in this regard. The PixArt camera used can track up to 4 IR points at a time, allowing for up to 2 erroneous points (with the two real beacons on the leader for relative x, y, and z positions). However in sunny conditions or with other infrared sources, the IR cameras used will invariably have noisy signals. In addition, following quadrotors could move into the LOS of other followers, blocking the IR transmission and preventing lead unit tracking. This can be mitigated to some extent by using the new (now blocking) quadrotor as a leader, though this requires additional methods of localization in order to create the proper RF links to send yaw angle information. Other physical obstructions would require a similar response, in which the quadrotor would become a follower to another unit that still maintains visual contact with its leader.

Furthermore, transmission of attitude information via 900 MHz wireless will, given distance and obstructions, begin to suffer from suboptimal throughput. Because this system relies on yaw information to accurately parse the camera information, such errors would be very detrimental to the flocking algorithms.

Discussed earlier were various methods for achieving accurate position data outdoors (using GPS), indoors (using a Vicon or similar motion capture system), and the more robust method of laser rangefinders. Leader-follower configurations using dedicated cameras offer a decentralized method for controlling large flocks. As in the MATLAB simulation, the leader for the group need not be the leader of each follower. Instead, an autonomous follower can track the unit immediately in front of it. The aim is for such a system to support more than three—perhaps even tens—of similar rotorcrafts. Simulations show that the proposed configuration scales well, but indeed more experiments must be undertaken in order to fully understand how the performance of this system and the cumulative error will scale with the addition of agents—the simulation does not provide adequate insight in this regard.

B. System Safety

The primary failsafe of the quadrotor system is an auto-land feature that is called when the signal of the RC controller is lost for an extended period of time. This is an autonomously-controlled procedure and uses the on-board sonar sensors to avoid obstacles and land safely. Similarly, should the follower lose sight of the leader at any time, it will level out in an attempt to regain vision, and if the visual link is not reestablished within a set time period, the follower will auto-land as well. Finally, an auto-land and emergency stop packet can be sent from the base station in case of a system malfunction.

X. Conclusion

We present the design and evaluation of a leader-follower autonomous aerial quadrotor system for search and rescue operations. The system discussed provides a robust platform with the following key features:

1. Infrared three dimensional tracking for a decentralized leader-follower flocking algorithm. IR beacons installed on the platform provides accurate data regarding the spatial orientation and placement of the system units. This allows for precise unit movement analysis and flight tracking of the units directly ahead, resulting in a flocking design that is resilient to sensor noise and errors.

2. Sensor fusion through combining RF, onboard sensors, and vision data. The IR camera used in addition with the attitude and sonar information of the units transmitted wireless over a RF connection provides a complete system overview through the use of less sophisticated sensors. This allows for system expandability and feedback to the user.

3. Platform for parallel computing of arbitrary sensor data (including SLAM). The units act as a collaborative unit and process data onboard before transmitting over the wireless network to the other units and base station. In addition to the ultrasonic and IR vision sensors, the available I^{2}C, serial, SPI, and analog connections on the sensor microcontroller provide for a wide range of additional sensors (such as audio, thermal, temperature, etc.) to be installed. Combining the data from these sensors in parallel provides users with critical mission information that would otherwise required additional mission time or more specialized equipment.

The leader-follower system will be demonstrated in CPS Week 2011, showcasing flight-tracking and cooperative sensor capabilities.

References

[1] R. Voyles and H. Choset. Editorial: Search and Rescue Robots. *Journal of Field Robotics*, 25, 2008.

[2] DraganFly Innovations Inc. Draganfly government & military. http://www.draganfly.com/our-customers/government.php.

[3] J. Leonard and H. Durrant-Whyte. Simultaneous map building and localization for an autonmous mobile robot. *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 3(5), Nov 1991.

[4] C. Howard. Automated Life Jacket Detection Enhances Search and Rescue Operations. *MilitaryAerospace.com*, 2011.

[5] D. Mellinger, N. Michael, and V. Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *Int. Symposium on Experimental Robotics*, 2010.

[6] L. Meier, F. Fraundorfer, and M. Pollefreys. Onboard Object Recognition on the PixHawk Micro Air Vehicle. *Fourth International Conference on Cognitive Systems*, 2010.

[7] R. De Nardi, O. Holland, J. Woods, and A. Clark. SwarMAV: A Swarm of Miniature Aerial Vehicles. *Bristol UAV Systems Conference*, 2006.

[8] G. Hoffmann, D. Gorur Rajnarayan, S. L. Waslander, D. Dostalv, J. Soon Jang, and C. J. Tomlin. The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). *Digital Avionics Systems Conference*, 2, 2004.

[9] M. Achetelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments. *Association for Unmanned Vehicle Systems International (AUVSI) Symposium*, 2009.

[10] X. Li, Z. Cai, and J. Xiao. Biologically Inspired FLocking of Swarms with Dynamic Topology in Uniform Environments.