

# Formation Flying for Satellites and UAVs

Garrick Merrill

Marshall Space Flight Center  
NASA  
Huntsville, AL  
Garrick.Merrill@nasa.gov

Chris Becker

Marshall Space Flight Center  
NASA  
Huntsville, AL  
Christopher.M.Becker@nasa.gov

**Abstract**—A formation monitoring and control system was developed utilizing mesh networking and decentralized control. Highlights of this system include low latency, seamless addition and removal of vehicles, network relay functionality, and the ability to run on a variety of hardware.

## I. INTRODUCTION

The shrinking size of Unmanned Aerial Vehicles (UAVs) is enabling lower cost missions. As sensors and electronics continue to downsize, the next step is multiple vehicles providing different perspectives or variations for more precise measurements. While flying a single UAV autonomously is becoming commonplace, flying multiple vehicles in a precise formation is still a challenge.

Our approach to a formation system was to develop a scalable mesh network between vehicles to share real-time position data and maintain formations autonomously. The mesh network is a custom designed Global Position System (GPS) synced Time Division Multiplexed Architecture (TDMA). This architecture allows radios to only be listening during specific time slices which helps to keep power usage low. External syncing via GPS allows all network nodes to be peers without any master node. A major cycle of at least once a second guarantees that each vehicle transmits its position with that frequency. Data relaying is built into the network architecture so that all vehicles do not need to be in direct communication with each other.

Formation modes were established to allow various formation types and control methods. The primary formation modes are designed to accommodate static and dynamic formations. Other modes include the ability to remove formation control and a failsafe mode that places vehicle(s) in non-interference altitudes to avoid collisions. Formation shapes can be preprogrammed before flight and then selected during flight along with new positions. Manual control of the formation is also possible by taking manual flight control of a single vehicle with all the other vehicles following in a dynamic formation. The combination of all the formation modes allow for flexibility as well as safety in testing and deployment.

In this paper we will go over the description of our formation system including the first and second generation designs. Following this we describe the formation operations and modes. Finally we will go over the vehicles used for testing and testing performed.

## II. DESCRIPTION

In this section we will go over the system architecture, our first and second generation systems, and the details of the node operation.

A key feature of the system architecture for this formation system is to not have a single point of failure for the entire system. The ability to handle vehicles being added and dropped from the network during operation was also desired. A peer to peer network topology was needed that operated without a central switch or master. This led us to a mesh network topology. In our first generation system we used a Digi XBee point-to-multipoint network architecture since it provided most of the features we needed in a commercially available product. In our second generation system we designed our own system based around a GPS synced Time Division Multiplexed Architecture (TDMA) so that we were not reliant on any specific radio.

### A. First Generation System

The first generation communication system employs a mesh network architecture to provide communication between all vehicles in the formation. Each deployed element of the system is referred to as a node. Every node in the system communicates with every other node in the system. The system does not require the use of a master or other coordinator to establish the network, and it will continue to function as designed regardless of how many or which nodes are currently present in the mesh network.

The first generation node hardware consists of a Beaglebone Black (BBB) single-board computer, two XBee Pro 2.4GHz radios, and a custom interface board (“cape”) (Figure 1). The XBee radios are attached to headers on the cape which in turn mate to the headers on the BBB. Each radio is used to communicate on a separate mesh network (operating on a different fixed frequency). The networks are completely independent and are used to provide redundancy for the communication system.

The mesh network control software consists of custom Python scripts running on the BBB. The software interfaces to the radios and the flight computer of the host platform via serial Universal Asynchronous Receiver/Transmitter (UART) devices. All data to be transmitted by a node is sent over both redundant mesh networks. The proprietary XBee radio

firmware is responsible for coordinating communication timing on the wireless networks to prevent collisions.

All nodes on the system broadcast the required status data and commands from their vehicle over the wireless networks. The first generation system does not employ message relaying functionality, but instead uses direct communication between all nodes thus requiring all nodes to be within range of all other nodes to able to receive data from all vehicles.

While the system was initially developed to function separately with its own independent hardware and software for modularity purposes, the system software could also be deployed to run directly on the host vehicle's flight computer. The system could also make use of existing wireless radios on the host assuming the necessary bandwidth was available and the radios provided the required coordination and timing.

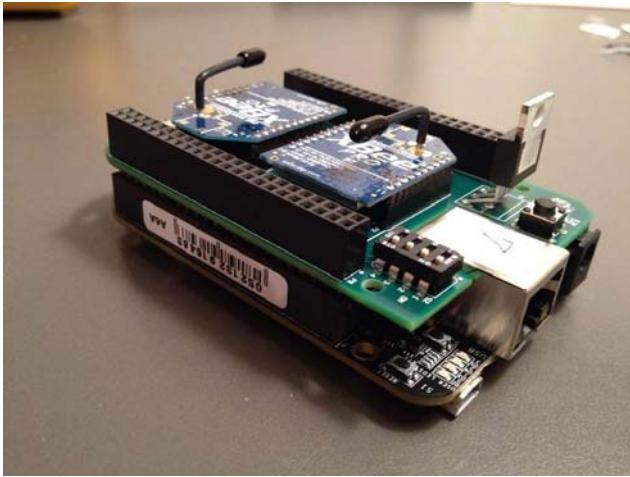


Figure 1 (First Generation Node)

#### B. Second Generation System

The second generation system was designed to make the communication system hardware independent, i.e. the system is not dependent on a particular model or brand of radio to function. To enable this, a custom TDMA scheme was developed to control the sequencing of communication on the network (Figure 2). This contrasts with the first generation system which did not have any software-based communication control scheme but instead relied on the XBee radios to provide this function. By moving this function into software, the system is not only hardware-independent, but this also helps cut down on power requirements by allowing the radio receiver and transmitter to be powered off when not in use.

To further reduce power requirements as well as mass, only one network is employed therefore only requiring one radio. To showcase the capabilities of the system and to demonstrate deployment on a wide range of hardware, a relatively simple radio with minimum complexity was chosen. The radio control logic is minimized by moving the collision and other communication control logic into the mesh network communication system software.

The 2<sup>nd</sup> generation system also employs relaying to allow nodes to communicate and pass data and commands between all vehicles without requiring direct communication between all nodes. Commands received by a node will be retransmitted by that node. This will allow commands to propagate along the mesh network to the desired destination node. The critical vehicle status data from all nodes will also be collected by each node and rebroadcast. In this way, a node that has no communication path to a particular vehicle can still receive that vehicle's data via other nodes.

Because the TDMA scheme requires precise timing, some method must be provided to synchronize the clocks of all nodes in the system. Because of its existing widespread use as a vehicle navigation source by many vehicle types, GPS was chosen as the time synchronization source. The time broadcast by the GPS constellation and a pulse per second (PPS) signal from a GPS receiver is used to provide time synchronization within 1 millisecond or less amongst the network nodes. However the communication system is not dependent on this particular time source, so any other external time synchronization method that is implemented by the host platform would be sufficient provided it meets the time accuracy requirements.

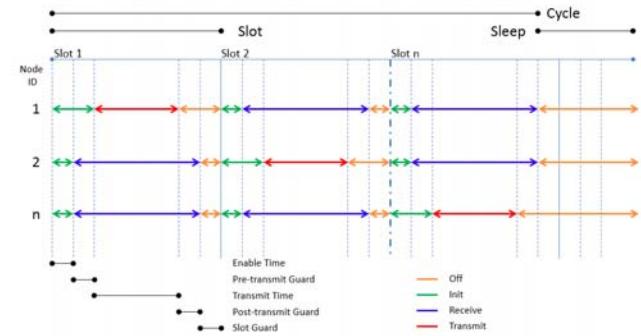


Figure 2 (TDMA Timing)

#### C. UAV Formation Node Operation

To test and demonstrate the capabilities of the mesh communication architecture, we used a formation of quadcopter small unmanned aerial systems (sUAS). We chose to use multicopters because they allow for precise movements, hover capability, and a minimum test range area requirement. The multicopters used a slightly modified version of the ArduCopter software running on Pixhawk flight computers. We modified the standard ArduCopter "Guided" mode to accept and implement position update commands sent from the formation computer. Guided mode is the only flight control mode that accepts commands from the formation system. This allows us to quickly discontinue formation control and safe the UAVs in the event of a failure.

Each formation node is responsible for determining the appropriate position commands to send to its host vehicle depending on the current operating mode. The formation control software is divided into the following operating modes:

-Passive: Upon formation control software startup, all nodes initially enter Passive mode. In this mode, the communication network is initialized and the nodes begin exchanging data among each other and communicating with the flight computer on their host vehicle. A formation node does not send position commands to its flight computer in this mode.

-Active: In Active mode, the formation nodes send position commands to their host vehicles to create a formation with the desired shape and at the latitude, longitude, and altitude. The parameters of the desired formation type can be changed via command sent to the formation nodes.

To prevent collisions between vehicles, all movements in active mode are coordinated between the vehicles. Before transitioning to a new horizontal position, each formation mode will transition to a “failsafe” non-interference altitude before making horizontal movements. This altitude is different for each node, so that nodes can move horizontally without fear of collision. Once a node reaches the desired latitude/longitude, the node will then command its vehicle back to the desired altitude. This phased implementation of new formation position commands is coordinated between all nodes, i.e. nodes will wait for all other nodes to finish the current phase before moving to the next.

-Leader/Follower: In the Leader/Follower modes, one node is set as the “leader”. This node can then be commanded by a Remote Control (RC) pilot to a new position/altitude. The other nodes will then autonomously “follow” the leader’s movements to maintain the desired formation. All nodes in Active mode will automatically transition to Follower mode when a leader is detected and will transition back to Active when the leader is no longer present.

-Failsafe: Failsafe mode is used to move a node to a non-interference altitude. Upon entering Failsafe, a node will maintain its current latitude and longitude, but change to failsafe altitude. This mode is entered autonomously based on certain failsafe conditions or can be commanded at any time from the ground control system.

### III. TESTING

In this section we’ll go over our test setup, the vehicles and operations of those vehicles, our software and ground control system, and the tests performed.

#### A. Test Setup

The formation nodes were mounted in an enclosure and each mounted to the bottom of a quadcopter (Figure 3). The node received power from the quadcopter and had a serial link to the flight computer. This data link provided current position data to the node and accepted position commands from the node.

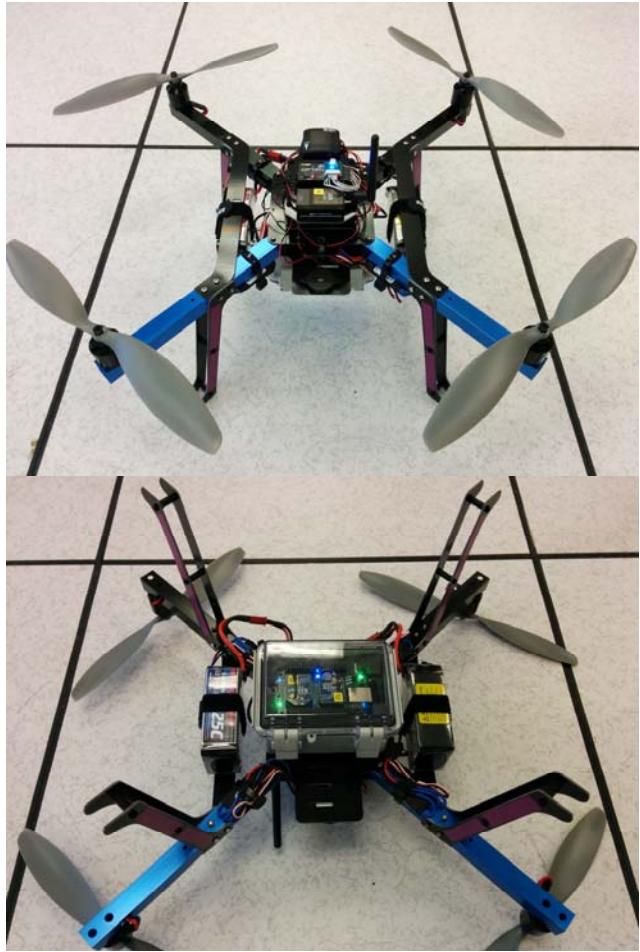


Figure 3 (Quadcopter Configuration)

#### 1) Vehicles/Operations

The UAVs chosen for testing the formation nodes are modified 3D Robotics Quadcopters. Larger motors and 12” propellers along with three cell Lithium Polymer batteries gave good performance and efficiency. Hitec Aurora 2.4GHz radios with Adaptive Frequency Hopping Spread Spectrum were used for manual flight control and configured to avoid the Xbee radio frequencies being used. 3D Robotics 915MHz telemetry radios were used for autonomous control. Other modifications made to the vehicles included relocating batteries and adding longer landing legs to accommodate the formation node enclosure. Flight time with the formation payload is around 15 minutes with two 60Wh batteries.

Flight tests were performed with a minimum of four people, a RC pilot, a ground control system (GCS) operator, a formation control system (FCS) operator, and an observer. The RC pilot had manual flight controls of any of all vehicles if problems occurred and had line of sight to all vehicles. The GCS operator was in primary control of all vehicles and control the autonomous flight controls of each vehicle. The FCS operator controlled and monitored the

formation system. The observer also maintained line of sight to all vehicles and kept watch over the flight area for other traffic.

## 2) Software/GCS

The ground control software used for flight testing is divided into two primary functions: ground control and formation control. These two functions are operated independently using separate laptop computers. Ground control is responsible for monitoring and controlling the individual UAVs. This control includes arming/disarming the vehicles, commanding vehicle flight mode changes, and monitoring all telemetry received from the vehicles' flight control avionics. Formation control consists of interfacing and commanding the formation nodes. This division of responsibilities allows the vehicle flight control systems to be operated independently of the formation system. This helps in the execution of flights tests by allowing for individual takeoff/landing of vehicles prior to formation flight tests and after test completion as well as enabling failsafe actions in the event of a formation system error.

The ground control system (GCS) consists of a software application that displays all vehicle telemetry and commanding functions in a single graphical user interface (GUI) (Figure 4). While there are several commercial off the shelf UAV ground control software options, their primary purpose is usually for the control of a single vehicle. To allow one operator to control multiple vehicles, the necessary telemetry and command functions need to be presented completely but concisely and use minimal screen real estate. The GUI shown in Figure X displays the required data and command widgets for multiple vehicles in a compact form. Necessary telemetry data such as current vehicle position/altitude/heading, power system status, and radio signal quality are displayed along with vehicle control command buttons, all contained in a small interface that is replicated for each vehicle. The GUI allows a single operator to maintain a comparatively large set of vehicles thereby helping to minimize ground personnel requirements.



Figure 4 (GCS GUI)

The formation control system is divided into two separate GUIs. One of these applications (Figure 5) provides status data from each formation node and command buttons to issue commands independently to a particular node or to the entire formation. This GUI can be used for formation mode changes, position command updates, and to reconfigure formation and mesh network settings.

The other formation control GUI presents a basic map that displays the current three-dimensional position of all the vehicles in the formation. This is helpful in providing better situational awareness of the vehicles' positions relative to each other. A map view is also useful to monitor and visualize whether the formation is maintaining the commanded formation shape and alignment. Like the GCS, the formation control system is manned by a single ground operator.

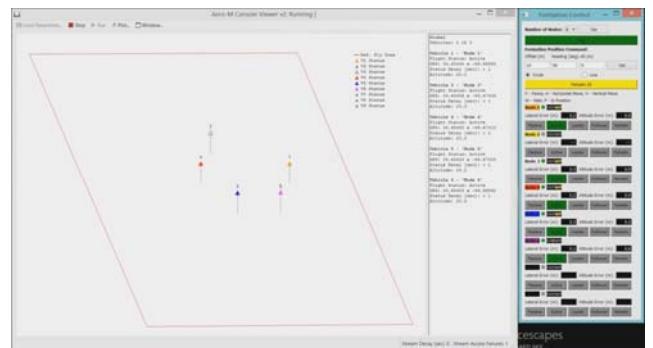


Figure 5 (Formation GUIs)

### *B. Tests Performed*

Testing of the formation system was done in three phases. The first phase was ground testing inside with simulated flight computers. A computer was used to simulate the Pixhawk flight computers, and all flight interfaces and GUIs could be used. The second phase was flight testing and involved six quadcopters in an open field. The final phase was satellite simulations and involved a computer to simulate a satellite flight computer.

Flight testing was done incrementally starting with a single vehicle and working up to six vehicles. This allowed us to uncover problems with fewer vehicles in the air. For each flight test we tested all formation modes. The vehicles took off autonomously in passive mode. Each vehicle flight computer was programmed with a simple waypoint mission to takeoff and ascend vertically to that vehicle's failsafe altitude. Since each vehicle's failsafe altitude was different, vehicles could drift at their altitudes and not be in danger of colliding. Once all vehicles were at their failsafe altitudes the nodes were commanded from passive to active mode. Each vehicle's flight computer was commanded to Guided mode which allowed the flight computers to start accepting formation commands from the nodes. The vehicles then began moving to their initial static formation position. Once all vehicles reached the initial position, formation changes including formation type, position, and altitude were commanded from the FCS operator (Figure 6).

To test the dynamic formation capability, a single vehicle was commanded to the leader mode, and all other vehicles were observed as they changed to the follower mode automatically. The leader vehicle was then changed to a semi-manual flight mode and the RC pilot began flying that vehicle. The other vehicles were observed as they followed the leader both in position and altitude. At any time during the test the failsafe mode was tested by commands from the FCS operator. A vehicle in failsafe stopped horizontal movement and changed to its failsafe altitude. Following this test the vehicle could be commanded back to the active mode to rejoin the formation. Once all tests were complete the vehicles were commanded into land or return to launch mode by the GCS operator and landed autonomously. A video of a five vehicle formation flight can be found at this link:

<https://www.youtube.com/watch?v=oH9C43To3Dk>



Figure 6 (UAVs in Formation)

The third phase of testing was focused on simulating multiple satellites in formation. This simulation used the same hardware as the initial phase, but with a computer simulating a satellite flight computer instead of a Pixhawk flight computer. We added a new formation type that was focused on keeping propellant usage to a minimum. This formation type had one satellite in the middle that did not use propellant with the other satellites circling around it using a minimum of propellant. This would allow satellites to stay in proximity to each other without a large impact to their propellant stores. We simulated up to four satellites as part of this phase.

#### IV. CONCLUSION

We believe we have developed and demonstrated a formation system that allows multiple UAVs or satellites to maintain that formation autonomously. With our first generation system we demonstrated that the concept was viable and that the mesh network could allow data exchanges between vehicles with low enough latency to allow for precise control. With our second generation system we demonstrated that this technology can be hardware independent and more robust with relay functionality. This type of system is applicable to satellites as well as current and future generation UAVs.

#### REFERENCES

[1] APM Autopilot Suite: <http://www.ardupilot.com>, 2015.