Autonomous Coupling of a UAV and UGV

By

Chris Breaux

Undergraduate Honors Thesis

November 8, 2019

APPROVED:

Panagiotis Artemiadis, ____, Director

Hyunglae Lee, ____, Second Reader

ACCEPTED:

____

Dean, Barrett, The Honors College

Abstract

A heterogeneous team of robots working in symbiosis can maximize their strengths while complementing each other's weaknesses. These simple robots can achieve more working together than they could on their own but cost less than a single robot with the same combination of capabilities. This project aims to validate the symbiotic relationship of an Unmanned Aerial Vehicle (UAV) and an Unmanned Ground Vehicle (UGV) with a physical implementation of a heterogenous team of robots and a demonstration of their capabilities. This paper details the selection of robots, the design of the physical coupling mechanism, and the design of the autonomous controls. An experiment was performed to assess the capabilities of the robots according to four performance criteria. The UGV must navigate a space while the UAV follows. The UAV must couple with the UGV. The UAV must lift the UGV over an obstacle. The UGV must navigate the space while carrying the UAV.

Contents

Introduction

Motivation

In disaster situations, victims may become trapped by fallen debris or immobilized by their injuries. First responders must react immediately to find survivors before the situation worsens and before the victims succumb to their injuries. First responders risk their own lives by entering buildings without fully understanding the dangers at play. Oftentimes, the workers become victims of the disaster themselves.

To mitigate these risks, teams of robots can enter a compromised structure before, alongside, or in place of human workers. A wave of robots entering a building before workers can navigate the space, identify points of interest, and plot a direct path for the human workers to follow. A mixed team of humans and robots working concurrently can cover more ground than a smaller team of human workers. More complex robots may completely replace human workers and provide aid directly to the victims.

Every task that a robot can do reduces the time that a human worker spends in harm’s way; however, search and rescue teams encounter many types of obstacles and must perform many types of tasks. A search and rescue robot must be able to traverse through a hallway, survey the surroundings, avoid or move fallen debris, climb stairs, open doors, deliver supplies, and administer care and transportation. Simple robots are cheap and expendable but will not be able to address many of these tasks. Increasingly complex robots can address more of these tasks but become more expensive and less expendable.

In addition to being more expensive, increasing the complexity of a robot to address its weaknesses can detract from its strengths. Unmanned aerial vehicles like the quadrotor platform are highly maneuverable and have high scope to survey the surroundings, but they have low payload capacity and low range. A larger UAV can accommodate a greater payload and battery capacity but will be less maneuverable in a tight hallway. Unmanned ground vehicles like the two-wheel drive and tracked robot platforms are less maneuverable and have a limited field of view from the ground, but they have greater payload capacities and greater range for extended missions. An ambulatory UGV can maneuver over obstacles, but the additional actuators consume more battery and add weight, decreasing its range and payload.

A heterogeneous team of robots working in symbiosis can maximize their strengths while complementing each other’s weaknesses. For example, a UAV can maximize its high mobility and scope and share those qualities with the UGV, and the UGV can maximize its high range and payload and share those qualities with the UAV. These simple robots can achieve more working together than they could on their own but cost less than a single robot with the same combination of capabilities.

Previous Work

A team from the University of Zagreb used the simulation shown in Figure 1 to demonstrate the symbiotic relationship of a UAV and UGV (Petrovic, Haus, Arbanas, Bogdan, & Orsag, 2015). In their simulation, the robots cooperated to maneuver around obstacles and close a valve. The team recognized that a UAV can lift a UGV over an obstacle, and the UGV can carry the UAV to conserve battery life. The team focused on developing high level task scheduling and mission planning algorithms but not the hardware required to do so. With a focus on algorithms over hardware, the team was able to use a simulation to demonstrate their results.

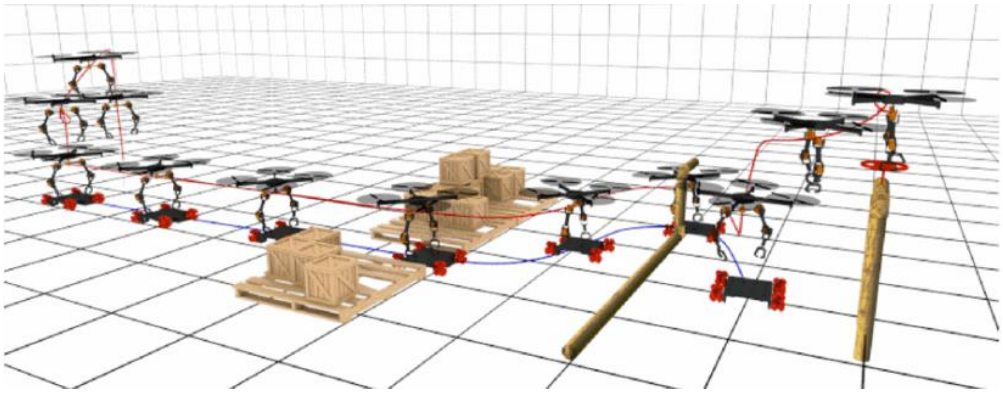


Figure 1: Simulation of a Symbiotic UAV and UGV Relationship

Problem Statement

This project aims to validate the symbiotic relationship first described and simulated by the Zagreb team with a physical implementation of a heterogenous team of robots and a demonstration of their capabilities. This paper details the selection of robots, the design of the physical coupling mechanism, and the design of the autonomous controls.

A successful implementation of this relationship must include a UAV and UGV team capable of the following:

1. The UGV must navigate a space while the UAV follows.
2. The UAV must couple with the UGV.
3. The UAV must lift the UGV over an obstacle.
4. The UGV must navigate the space while carrying the UAV.

Method

Robot Selection

The selection of robots primarily depends on the weight of the UGV and the payload capacity of the UAV. The UGV must be lighter than the UAV's payload capacity but large enough to serve as a stable platform upon which the UAV can land and ride. The UAV must be large enough to lift the UGV but small enough to maneuver through a hallway and land on the UGV.

The two-wheel drive UGV platform shown in Figure 2 was selected for its light weight and stability. The acrylic chassis has a 140-millimeter diameter, making for a light platform weighing only 330 grams. The two wheels and two casters create four points of contact with the ground, making the platform very stable. Additionally, the mounting of the motors and casters underneath the platform allow enough room inside of the platform to house the battery and computer. This leaves the flat upper deck open for the coupling mechanism to be mounted.

The Intel Aero quadrotor platform shown in Figure 3 was selected for its payload and size. The dry weight of the UAV is 865 grams, but the 1900-gram maximum takeoff weight allows up to a kilogram of additional payload. The 360-millimeter diagonal hub-to-hub length makes the UAV maneuverable in tight spaces, and the open frame presents multiple mounting points for the coupling mechanism.

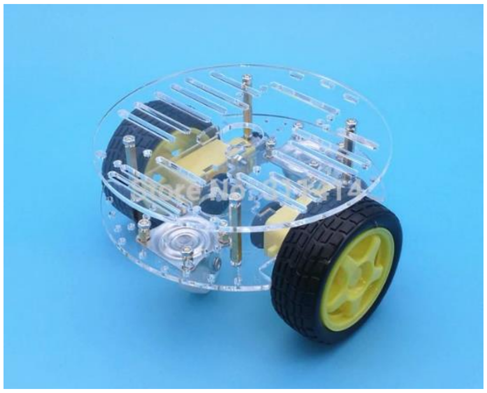


Figure 2: 2WD UGV Platform

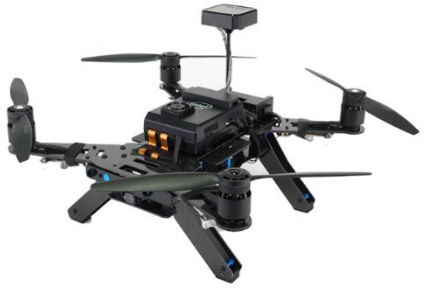


Figure 3: Intel Aero UAV Platform

Coupling Mechanism Design Requirements

There are two primary design requirements of the coupling mechanism. First, the mechanism must allow the UAV to lift the UGV and remain coupled during flight. Specifically, the final assembly of the robots and their mechanisms must not weigh more than the UAV's maximum takeoff weight, and the bond must sustain the weight of the UGV and the accelerations during flight. Second, the mechanism must allow the UGV to carry the UAV and remain coupled while driving. Specifically, the robots must remain coupled when the UAV runs out of battery, and the platform must sustain the weight of the UAV and the accelerations during driving.

There are three secondary design requirements of the coupling mechanism. First, the coupling mechanism must account for the robots' resolution of control. Inaccuracies of the robot's localization and control create a space around the desired position that contains the actual position. The coupling interface must be larger than this space to guarantee coupling from any point in the space. This can be accomplished by making the mechanism larger or by applying forces to guide the coupling points together. Second, the maneuver to reach the coupled position and to engage and disengage the coupling mechanism must be simple. A UAV's control outputs are not independent, so certain maneuvers will make coupling more difficult. For example, horizontal velocity depends on the pitch and roll of the vehicle. A maneuver that requires the UAV to pass through a point with a horizontal velocity will create undesired pitch or roll at the coupling interface. All of these errors are magnified when a sequence of maneuvers is required. Any errors that occur during the coupling process propagate through the rest of the sequence, increasing the likelihood that the coupling points miss their target. A simple coupling maneuver must occur in a single step and must preserve the other control outputs of the UAV. Third, the coupling mechanism must not affect the control of the robots. Introducing a swinging load or changing the location of the center of mass may alter the dynamics of the system beyond the robots' capabilities. The coupling mechanism must connect the robots in a way that preserves the dynamics of the system.

Coupling Mechanism Design

The final coupling mechanism design consists of an electromagnet interface and a ferromagnet interface shown in Figure 4. The EM interface mounts to the upper deck of the UGV and consists of a circular ring with an electromagnet at its center. The FM interface mounts to the bottom deck of the UAV and consists of the same circular ring with a steel plate at its center. Both of these parts were designed in SolidWorks.

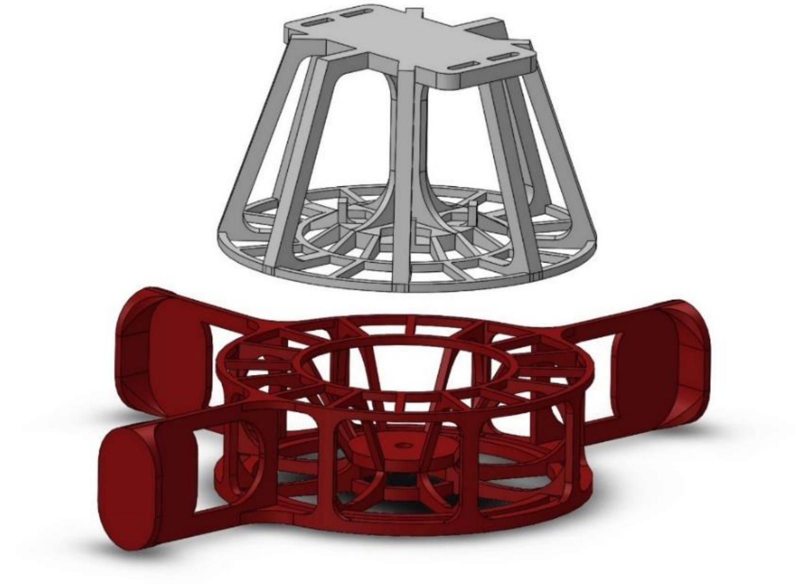


Figure 4: EM Interface (Red) and FM Interface (White)

The primary function of the EM interface is to attach the electromagnet to the UGV. A slotted ring at its base creates a variable mounting point, allowing the electromagnet to be centered over the UGV’s center of mass. The specific electromagnet, shown in Figure 5, was selected for its 5-kilogram holding force and its compatibility with a Raspberry Pi. The electromagnet is 20 millimeters tall with a 25-millimeter diameter, and it draws 0.3 amps at 5 volts. The electromagnet protrudes approximately 1 millimeter out of the plane of the interface so that it can be the first point of contact with the UAV. The electromagnet mounts to the UGV so that the connection does not depend on the UAV’s power. The UGV powers the connection even when the UAV is out of battery, and the connection does not draw power away from the UAV’s motors during lifting. At 30 millimeters tall, the EM interface is tall enough to accommodate the height of the electromagnet but short enough to sit below the propellers if the UAV were to land nearby. Three wings extend from the circular base to provide mounting points for the reflective tracking markers. This ensures that the markers remain visible while the robots are coupled.

The primary function of the FM interface is to attach the steel plate to the UAV. Two slots at its base mount to two similar slots in the UAV’s frame below its center of mass. The circular steel plate, shown in Figure 6, has a 35-millimeter diameter and four tabs that extend behind the coupling plane and fasten to the base. The plate protrudes approximately 1 millimeter out of the plane of the interface so that it can be the first point of contact with the UGV. A ferromagnetic material was chosen over a permanent magnet because the plate is only magnetized when the electromagnet is engaged. A permanent magnet would attract the electromagnet even when it is disengaged. This residual pull may disrupt the decoupling process. At 50 millimeters tall, the FM interface is nearly level with the UAV’s landing gear. This prevents the landing gear from hitting the UGV before the interface does. The vertical separation of the interface and the UAV limits interference with the UGV’s reflective tracking markers.

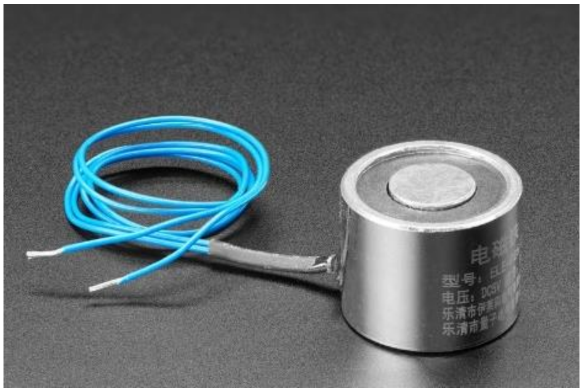


Figure 5: Electromagnet

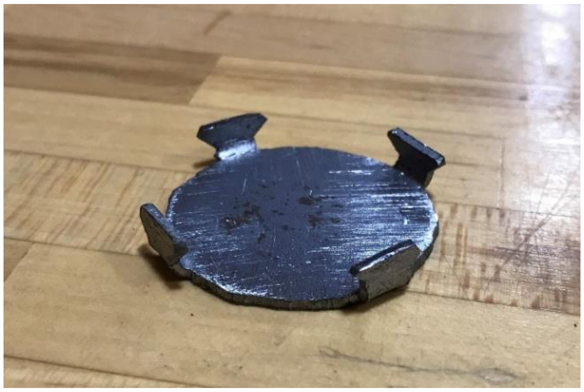


Figure 6: Steel Plate

Both the EM interface and the FM interface have a flat, circular ring with a 90-millimeter diameter that stabilizes the connection. During flight, the electromagnet applies a normal holding force. By design, this tension will always be greater than the force required to lift the UGV. This normal force also creates friction that prevents shearing mode separation of the magnets. The magnets are easiest to separate by applying a bending moment at the connection. The circular rings provide support away from the neutral axis to prevent opening mode separation. During driving, the circular rings serve as a stable platform upon which the UAV can land and ride.

The maneuver to reach the coupling position and to engage and disengage the coupling mechanism is simple. The UAV hovers above the UGV and matches the UGV's stationary ground position. The UGV engages the electromagnet, and the UAV descends towards the UGV, as if to land. The vertical descent has no effect on the other control outputs of the UAV. When the UAV is near the UGV, the magnetic field guides the coupling points to the coupled position. The first points that contact are the electromagnet and the steel plate, and the coupling is secure as soon as this contact is made because the bond is magnetic and not mechanical. The electromagnet remains engaged while the UAV lifts the UGV or while the UGV drives with the UAV. To decouple, the UGV disengages the electromagnet, and the UAV ascends again.

Hardware and Software Setup

The UGV's computer consists of a Raspberry Pi with a Motor HAT. The Raspberry Pi runs an Ubuntu MATE 16.04 operating system with ROS Kinetic installed. The Raspberry Pi uses the Adafruit_MotorHAT library to control the motors and the electromagnet (Fried, 2017).

The UAV's computer consists of an Intel Aero Compute Board and an Intel Aero Flight Controller. The Compute Board runs an Ubuntu 16.04 operating system with ROS Kinetic installed (Guermonprez, 2018). The Flight Controller runs Dronecode's PX4 Autopilot to control the motors. The Compute Board receives tracking data using the vrpn_client_ros package (Bovbel, 2017) and relays transformed data to the Flight Controller using the mavros package. The Flight Controller receives the offboard pose data, and the LPE position estimator fuses it with data from an onboard IMU (Willee, 2019).

A laptop running an Ubuntu 16.04 operating system remotely accesses the robots via secure shell protocol to activate ROS nodes and to execute Python scripts. The laptop uses Dronecode's QGroundControl application to calibrate the UAV and to adjust control parameters. QGroundControl is used to arm the motors and to switch the UAV to its offboard control mode.

Localization

This project was performed in Arizona State University's motion capture space, shown in Figure 7. In this space, an array of OptiTrack motion capture cameras track the positions of small reflective markers. The camera array is operated from a single desktop computer running OptiTrack's Motive software. A cluster of markers attached to a robot can be defined as a rigid

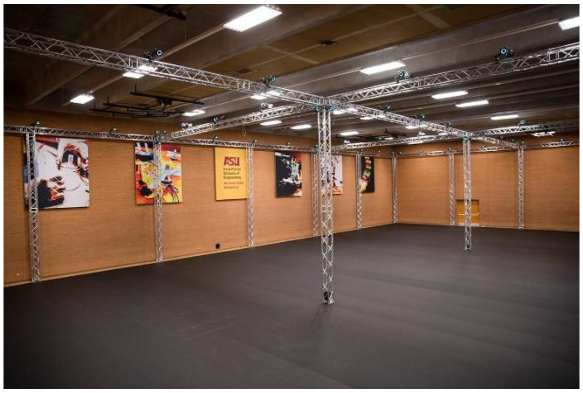


Figure 7: Motion Capture Space

body in Motive. This particular setup can track objects with an accuracy of less than a millimeter in position and less than a degree in orientation. Motive hosts a VRPN server that publishes the real-time pose data of the robots to the Wi-Fi network. The UAV launches roscore and the vrpn_client_node which subscribes to the tracked objects published by the desktop. The UGV communicates with the ROS master running on the UAV, so all of the pose topics are automatically available to the UGV without launching a master or VRPN client of its own.

With the up-axis set as Z in Motive, ROS outputs pose data in the Forward-Left-Up and East-North-Up local and world frames of reference shown in Figure 8. The UGV uses this frame of reference for its controls. However, the UAV's PX4 Autopilot must receive input pose data in the Forward-Right-Down and North-East-Down local and world frames of reference shown in Figure 9. To make this transformation, the UAV launches a mavros node and leaves it running while the VRPN client remaps the UAV's pose topic to /mavros/mocap/pose. A mavros plugin automatically performs the transformation and relays it to the PX4 (Willee, 2019).

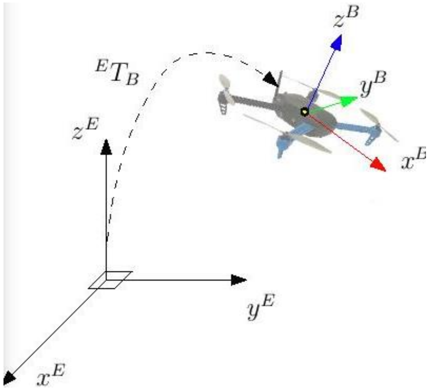


Figure 8: ROS frame: FLU/ENU

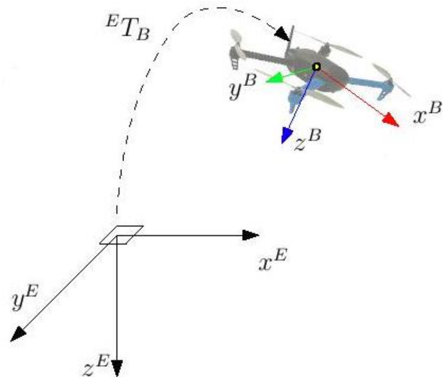


Figure 9: PX4 frame: FRD/NED

Offboard Position Control

The UGV runs a Python script that subscribes to its pose topic and drives the vehicle through a list of predefined waypoints (Singhvi, Biyani, & Artemiadis, 2019). The UGV simulates a differential in the motors to steer itself to the waypoints. The UGV calculates the desired steering angle based on its position relative to the waypoint and compares it to the actual angle it receives from the pose data. A proportional-derivative controller calculates the steering differential required to drive the actual angle to the desired steering angle. When the UGV reaches an acceptable distance to the waypoint, the UGV will target the next waypoint on its list or stop if it is at the end of its path. The UGV engages the electromagnet with the same commands that drive the motors. The magnitude of the motor velocity corresponds to the strength of the electromagnetic force. Reversing the current direction reverses the poles of the electromagnet.

The UAV runs a Python script that subscribes to its pose topic and drives the vehicle through a list of predefined waypoints (Antener, 2015). Offboard position control of the UAV is performed using the setpoint package in mavros. The setpoint commands are accompanied by an acceptable distance and a delay. The UAV flies within the acceptable distance to the waypoint and maintains that position for the duration of the delay. After the delay, the UAV performs the next setpoint command. Every sequence must end with a setpoint command to a landed position.

Final Assembly

The final assembly of the UGV, shown in Figure 10, includes all of the components required for localization and control. Two motors and two casters are mounted below the lower deck of the UGV. A Li-Po battery and Raspberry Pi with a Motor HAT are housed inside of the platform. The 3D printed EM interface is mounted on the upper deck, and reflective tracking markers are mounted to its wings. The rearward mounting of the motors and battery shift the center of mass behind the UGV's geometric center. The top view, shown in Figure 11, shows the EM interface's mounting position over the UGV's center of mass behind its geometric center. The final assembly of the UGV weighs 639 grams.

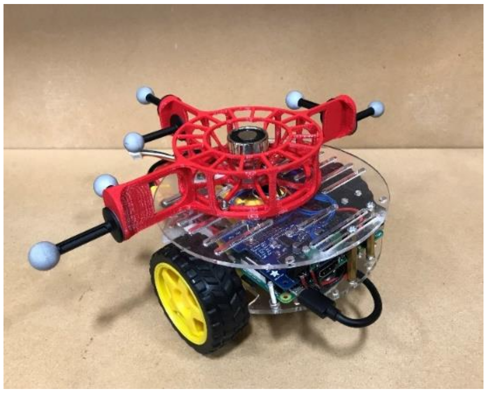


Figure 10: UGV Final Assembly

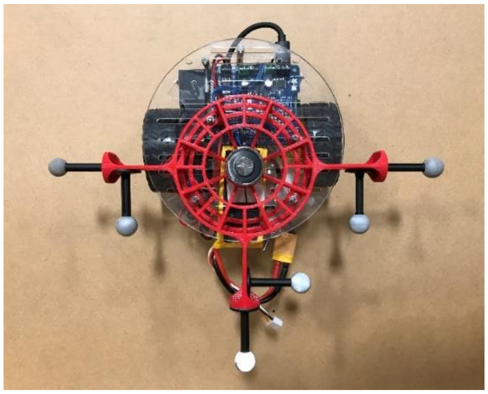


Figure 11: UGV Top View

The final assembly of the UAV, shown in Figure 12, includes all of the components required for localization and control. The 3D printed FM interface is mounted below the lower deck of the UAV, and a Li-Po battery is housed inside of the platform. The rest of the UAV is mostly original, but the odometry sensors were secured to the frame, and the Wi-Fi antenna and the propellers were replaced after a crash. The cameras were removed because they were not used. The reflective tracking markers are mounted to the upper deck of the UAV. The bottom view, shown in Figure 13, shows the FM interface's mounting position below the UAV's center of mass at its geometric center. The final assembly of the UAV weighs 1015 grams.

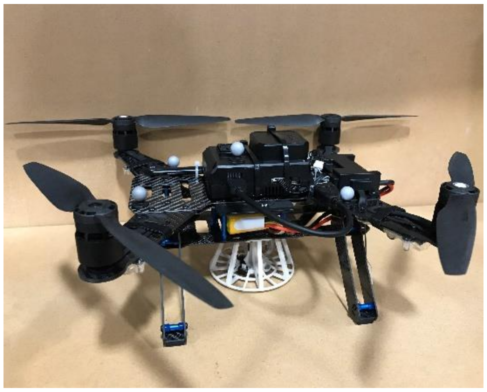


Figure 12: UAV Final Assembly

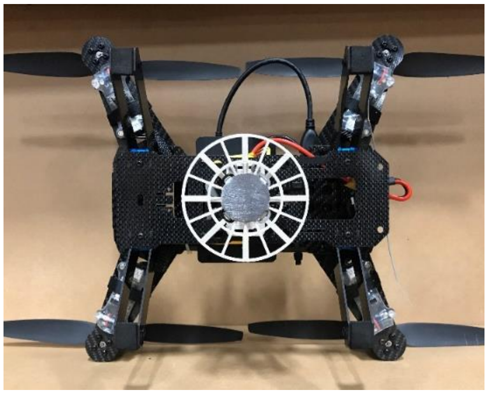


Figure 13: UAV Bottom View

The final assembly of the coupled system, shown in Figure 14, includes both robots connected by their magnetic interfaces. The reflective tracking markers mounted to the wings of the EM interface extend beyond the landing gear of the UAV. The landing gear of the UAV does not interfere with the coupling mechanism. The total height of the coupling mechanism that separates the upper deck of the UGV and the lower deck of the UAV is 8 centimeters. The final assembly of the coupled system weighs 1654 grams which is 246 grams below the UAV's maximum takeoff weight or 87% of the UAV's maximum takeoff weight. The UGV experiences a 159% increase in load during coupled driving. The UAV experiences a 63% increase in load during coupled flight.

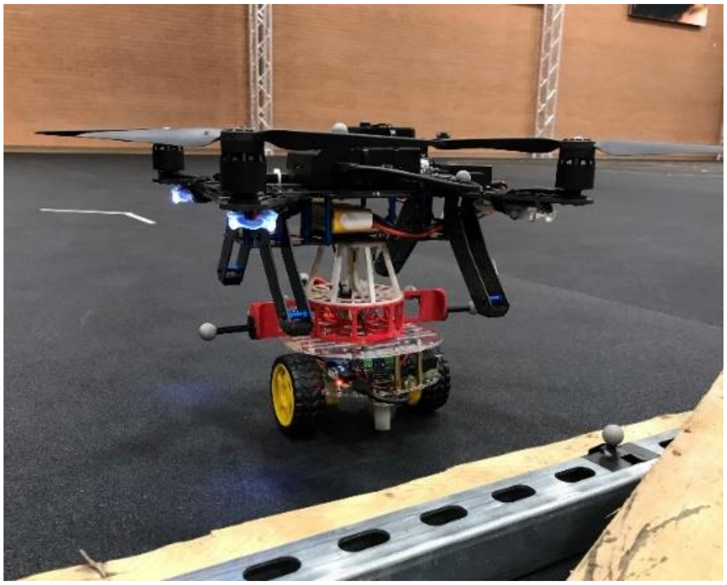


Figure 14: Coupled System Final Assembly

Experiment

Objective

The objective of this experiment is to assess the capabilities of the robots and their fulfillment of the performance requirements described in the problem statement. Specifically, the robots must demonstrate their ability to navigate a space, to couple, to fly over an obstacle, and to drive while coupled. A test bed was constructed in the motion capture space to demonstrate the results. The experiment was performed in this space because it isolates the performance of the hardware. The controlled environment limits disturbances to the system, and the tracking capabilities provide accurate localization. Therefore, the results of the experiment depend solely on the control of the robots and the performance of the coupling mechanism. The motion capture space can also store the tracking data to analyze the robots' performance.

In the experiment, the robots progress through a course defined by six waypoints interrupted by an obstacle, shown in Figure 15. The robots begin the experiment uncoupled, making two 45-degree turns through the first waypoints before they stop at the fallen debris. The UAV then couples with the UGV and lifts it over the obstacle. The team then decouples and proceeds briefly before stopping at the third waypoint in the straight section and coupling again to simulate the low battery life of the UAV. The UAV disarms its motors while the UGV drives through two more 45-degree turns to the final position.

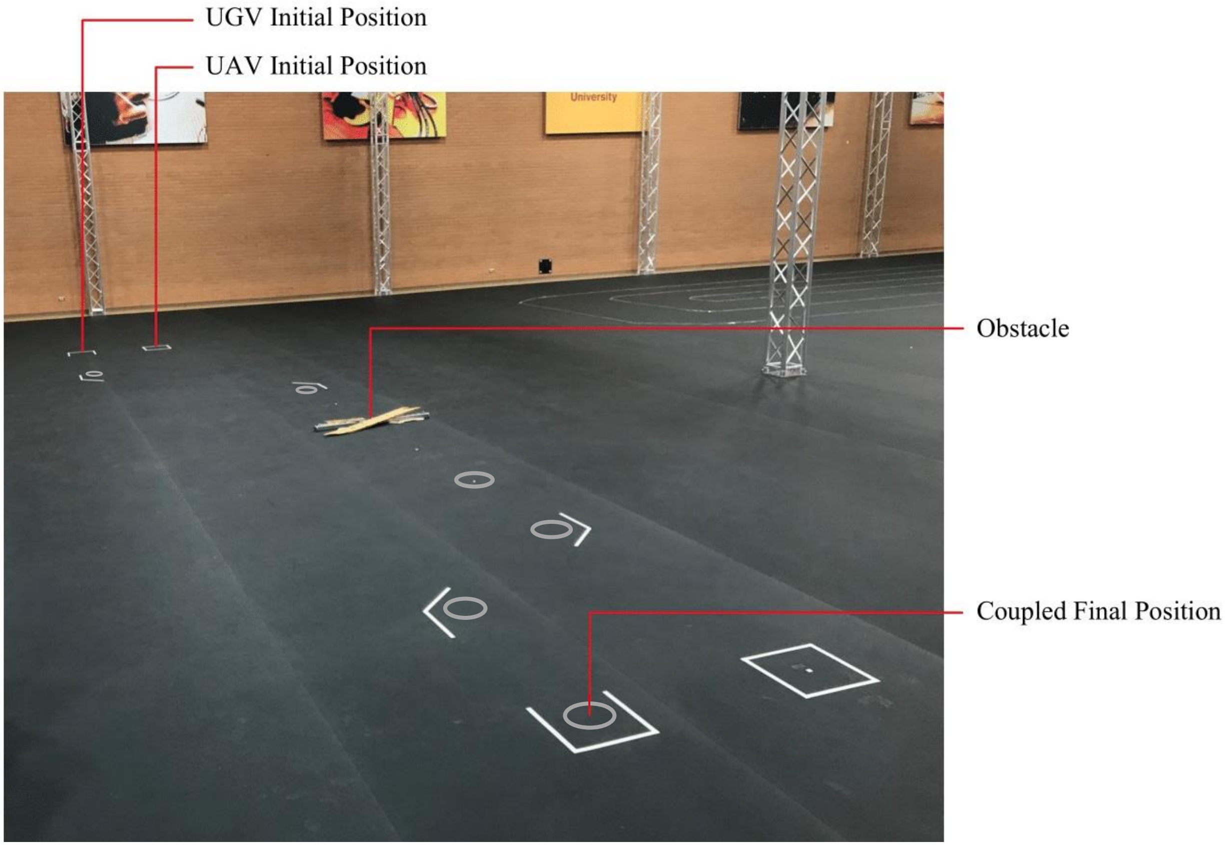


Figure 15: Experimental Test Bed

Autonomous Controller Design

At the start of the sequence, the UGV subscribes to pose topics for both robots and the obstacle and a “fly mode” topic that the UAV publishes to communicate its status. Every update to the UGV’s pose topic invokes its main operating function, and a “drive mode” variable decides which action it will perform. The UGV begins the experiment in a “standby” mode until the UAV takes off and matches the UGV’s stationary ground position. During this mode, the UGV continually evaluates its distance to the UAV. When the UAV flies within 10 centimeters of the UGV’s ground position, the UGV switches to “drive” mode and proceeds to the first waypoint. As it drives, the UGV continually evaluates its distances to the waypoint and the obstacle. When the UGV is within 7 centimeters of the waypoint, it targets the next waypoint and continues driving. The 7-centimeter threshold is equal to the radius of the robot itself and maximizes accuracy while limiting oscillations near the target. When the UGV is within 30 centimeters of the obstacle, it stops, switches to “engage” mode, and waits for the UAV’s instruction. The UAV updates its fly mode to begin the coupling procedure. The UGV activates its electromagnet and leaves its motors off while the UAV lifts it up and over the obstacle. When the UAV lands, it updates the fly mode again to begin the decoupling procedure. The UGV disengages the electromagnet but leaves its motors off while the UAV ascends to the following position. When the UAV is overhead, it updates its fly mode. The UGV switches back to drive mode and continues on to the waypoints. At the third waypoint, in the straight section, the robots couple again. Instead of lifting, the UAV switches to a “resting” fly mode which instructs the UGV to switch back to drive mode but to leave the electromagnet engaged. The UGV continues through the last two turns and stops at the final position.

At the start of the sequence, the UAV subscribes to pose topics for both robots and the obstacle and starts a publisher for its fly mode. The UAV targets a waypoint 1.5 meters above its initial position to take off and waits 2 seconds to settle before proceeding. As it flies, the UAV continually evaluates the UGV’s distance to the obstacle. While the UGV drives and is not near the obstacle, the UAV continually targets waypoints 1.5 meters above the UGV’s ground position with 10-centimeter accuracy to quickly follow it through the course. When the UGV reaches the obstacle, the UAV updates its fly mode to begin the coupling procedure. To prevent tracking interference during the coupling procedure, the UAV saves the UGV’s stationary ground position and uses these coordinates to set waypoints for its descent. The UAV targets waypoints over the coupling position with decreasing altitude and increasing accuracy. When the UAV is within 1 centimeter of the coupling position, it waits 2 seconds to settle before proceeding. The UAV targets a waypoint 0.6 meters above the coupling position with 20-centimeter accuracy to account for the hindered control of the UAV under load. The UAV targets another waypoint on the far side of the obstacle, and one more waypoint in the landed position. When the UAV lands, it waits 1 second to settle, updates the fly mode to decouple, waits 1 second for the electromagnet to disengage, then targets a waypoint 1.5 meters above its landed position to take off. Once in the air, the UAV updates the fly mode, and continues following the UGV. At the third waypoint, in the straight section, the robots couple again. Instead of lifting, the UAV updates the fly mode indicating a “resting” state and disarms its motors while the UGV carries it through the last two turns and stops at the final position.

Results and Discussion

Following

During the following phase of the experiment, the UGV drives through two 45-degree turns while the UAV follows. The plot shown in Figure 16 shows a top-down view of the robots' paths during this phase, starting from the moment the UAV takes off to the moment the UGV reaches the boundary of the obstacle. The plot was projected onto the x-y plane because the top-down view best illustrates the robots' behavior. The robots' paths varied least in the z-direction because the UGV drives on the ground plane, and the UAV targets waypoints at a constant altitude of 1.5 meters.

The UAV's path is indicated by a red line with a circle at the initial position and a cross at the final position. The UGV's path is indicated by a blue line with a circle at the initial position

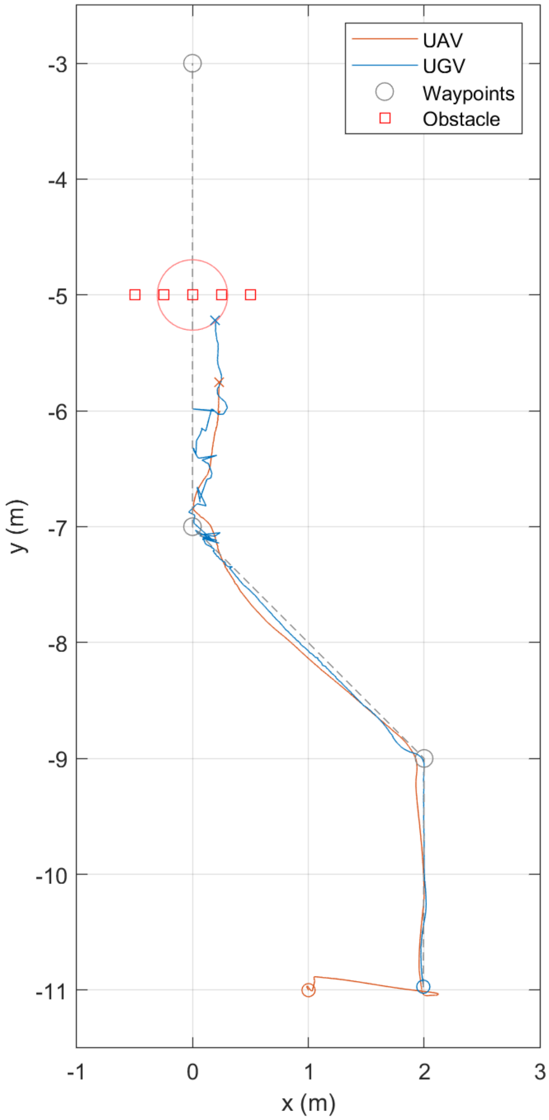


Figure 16: Following Phase Tracking Map

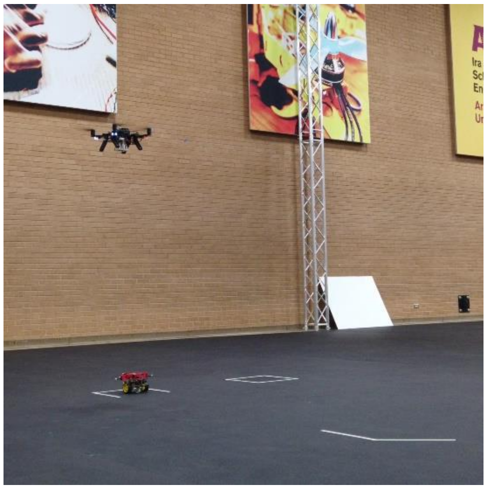


Figure 17: Start Position

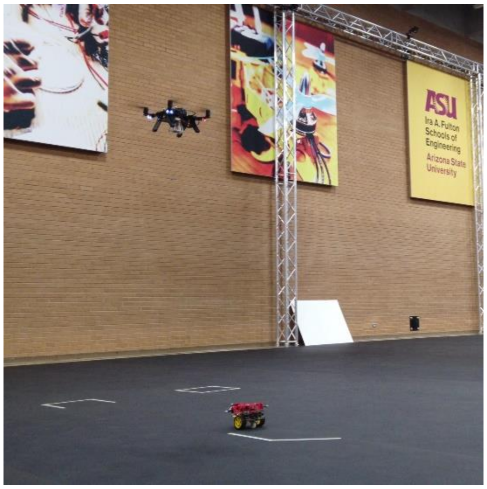


Figure 18: UAV Following UGV

and a cross at the final position. Waypoints are indicated by gray circles whose radii correspond to the thresholds set in the UGV's code. Red squares indicate the obstacle's fixed position, and a red circle indicates the boundary around it.

From its initial position, the UGV targets the first waypoint at the first turn and begins driving. Once the UGV reaches the boundary of the first waypoint, it targets the second waypoint at the second turn and changes direction. Once the UGV reaches the boundary of the second waypoint, it targets the third waypoint on the far side of the obstacle and changes direction. The UGV temporarily loses control when the UAV interrupts the camera array's view of the UGV. The UGV regains control and continues toward the third waypoint. When the UGV reaches the boundary of the obstacle, it stops. This concludes the following phase.

From its initial position, the UAV takes off and meets the UGV at the start position shown in Figure 17. While the UGV drives, the UAV continually targets waypoints above the UGV's ground position to follow it through the two turns. When the UGV reaches the boundary of the obstacle, the UAV begins the coupling phase.

The following phase was successful because the UAV was able to follow the UGV as it drove through two 45-degree turns. Besides the tracking interference, the UGV was able to drive to its waypoints and change direction with minimal overshoot. Despite the tracking interference, the UAV was able to follow the UGV's path very closely. The following positions shown in Figure 18 and the final positions shown in Figure 16 show that at each point along the path, the UAV is approximately 0.5 meters or 1 second behind the UGV.

Coupling

During the coupling phase of the experiment, the UAV matches the UGV's ground position and descends to the coupling position. The plot shown in Figure 19 shows a side view of the UAV's path during this phase, starting from the moment the UGV reaches the boundary of the obstacle to the moment the robots are coupled. The plot was projected onto the y-z plane because the side view best illustrates the UAV's behavior. The UAV's paths varied least in the x-direction because the UAV targets waypoints at a constant x-coordinate. The side view of the plot matches the views in Figure 20 and Figure 21.

The UAV's path is indicated by a red line with a circle at the initial position and a cross at the final position. The UGV's stationary position is indicated by a blue circle. Waypoints are indicated by gray circles whose radii correspond to the thresholds set in the UAV's code. Red squares indicate the obstacle's fixed position.

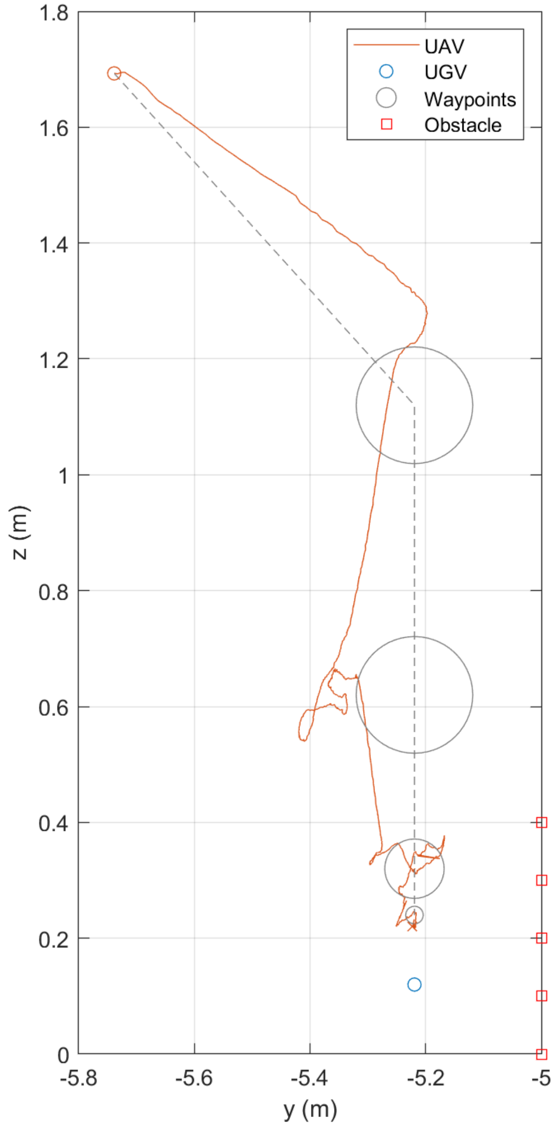


Figure 19: Coupling Phase Tracking Map

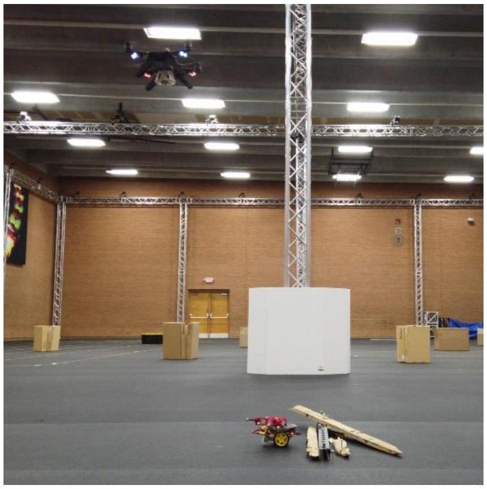


Figure 20: Initial Position

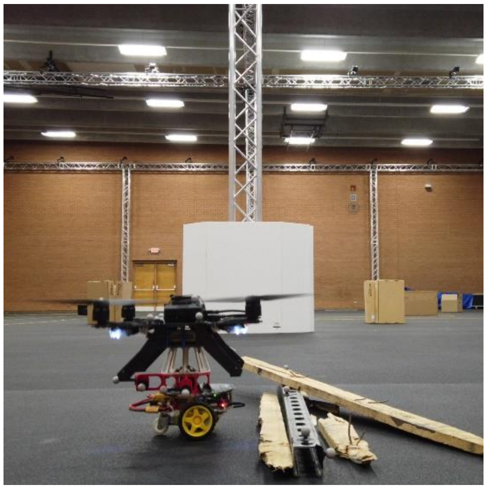


Figure 21: Coupled Position

From its initial position, the UAV targets a waypoint 1 meter above the UGV's stationary ground position with an accuracy of 10 centimeters. Once the UAV reaches the boundary of the first waypoint, it targets the next waypoint 0.5 meters above the UGV's stationary ground position with an accuracy of 10 centimeters. Once the UAV reaches the boundary of the second waypoint, it targets the next waypoint 0.2 meters above the UGV with an accuracy of 5 centimeters. Then it targets the second to the last waypoint 0.12 meters above the UGV with an accuracy of 1 centimeter. At this point, the FM interface is about 2 centimeters above the EM interface. The UAV targets the last waypoint, and the magnets guide it to the coupling position and engage. This concludes the coupling phase.

The coupling phase was successful because the UAV was able to meet the UGV at the coupling position and to engage the coupling mechanism. The UAV was able to fly to its waypoints with minimal overshoot. The overshoot and error decrease as the accuracy increases. The last waypoint brings the UAV within range of the electromagnetic force that corrects the small error and guide it to the coupling position.

Lifting

During the lifting phase of the experiment, the UAV lifts the UGV over the obstacle and lands on the other side. The plot shown in Figure 22 shows a side view of the UAV's path during this phase, starting from the moment it takes off to the moment it lands on the other side of the obstacle. The plot was projected onto the y-z plane because the side view best illustrates the UAV's behavior. The UAV's paths varied least in the x-direction because the UAV targets waypoints at a constant x-coordinate. The side view of the plot matches the views in Figure 23 and Figure 24.

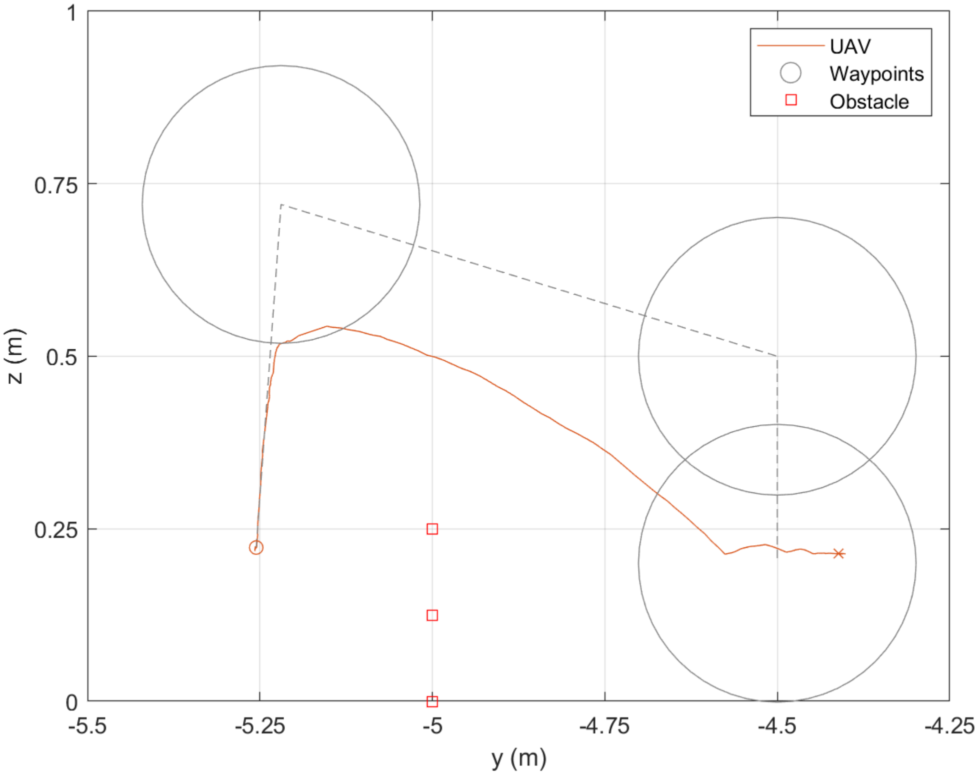
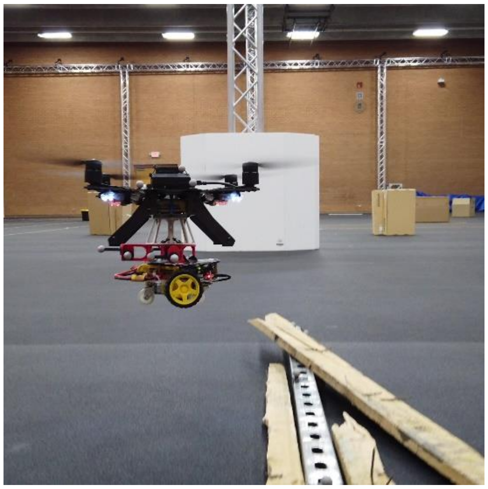


Figure 22: Lifting Phase Tracking Map



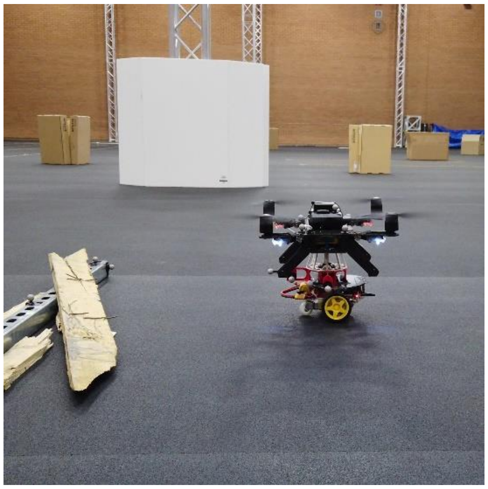


Figure 23: UAV Lifting UGV

Figure 24: Landed Position

The UAV’s path is indicated by a red line with a circle at the first position and a cross at the final position. Waypoints are indicated by gray circles whose radii correspond to the thresholds set in the UAV’s code. Red squares indicate the obstacle’s fixed position.

From its initial position, the UAV targets a waypoint 0.6 meters above the coupling position with an accuracy of 20 centimeters. Once the UAV reaches the boundary of the first waypoint, it targets the next waypoint 0.5 meters past the obstacle and 0.5 meters above the ground with an accuracy of 20 centimeters. Then, the UAV targets a waypoint at the landed position. This concludes the lifting phase.

The lifting phase was successful because the UAV was able to lift the UGV and land on the other side. The UAV had enough thrust to lift the UGV to its first waypoint but applied too great of a roll angle when changing direction and was not able to keep enough elevation to fly to its second waypoint. The UAV’s code should not have advanced past the second setpoint command without reaching the boundary of the waypoint. The code was paused after the UAV landed, and the error was not realized until the plots were generated, so the cause of the error is uncertain. A localization error might make the UAV appear to be within the boundary, causing the code to advance to the third setpoint command. The UAV might have continued to target the second waypoint after landing if the code had not been paused. The UAV was still able to land on the other side of the obstacle, and it would probably be able to reach the waypoints with less aggressive controls.

Resting

During the resting phase of the experiment, the UGV drives through two 45-degree turns while carrying the UAV. The plot shown in Figure 25 shows a top-down view of the UGV's path during this phase, starting from the moment the UGV begins driving to the moment it reaches the boundary of the final waypoint. The plot was projected onto the x-y plane because the top-down view best illustrates the UGV's behavior. The UGV's paths varied least in the z-direction because the UGV drives on the ground plane.

The UGV's path is indicated by a blue line with a circle at the first position and a cross at the final position. Waypoints are indicated by gray circles whose radii correspond to the thresholds set in the UGV's code.

From its initial position, the UGV targets the first waypoint at the first turn and begins driving. Once the UGV reaches the boundary of the first waypoint, it targets the second waypoint

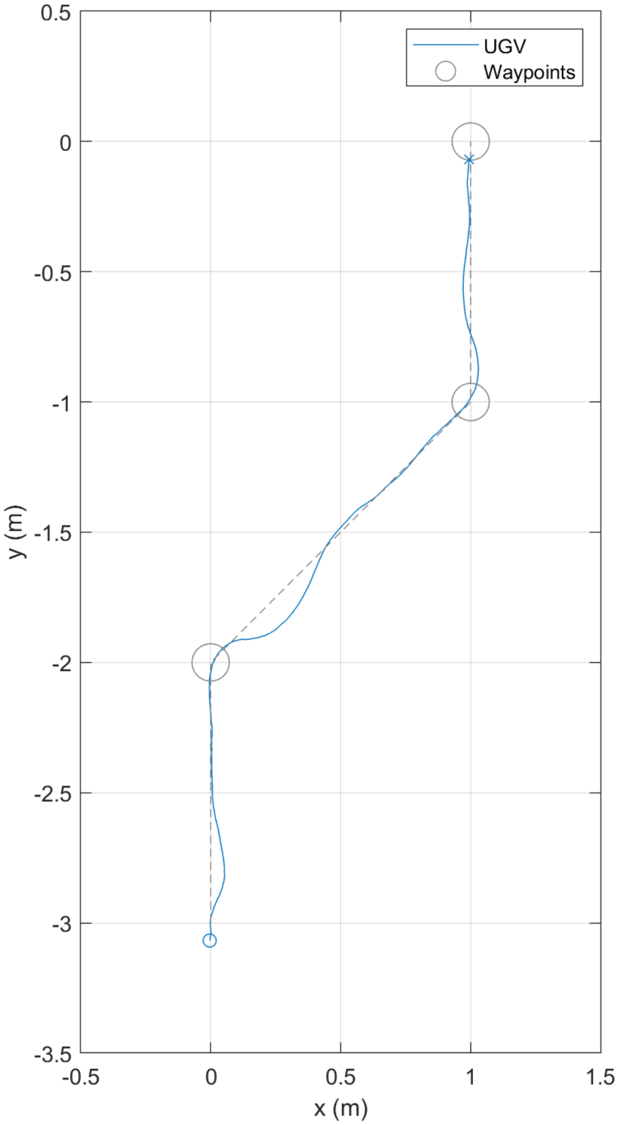


Figure 25: Resting Phase Tracking Map

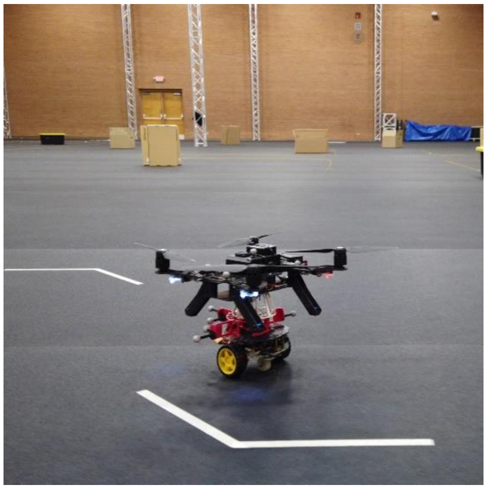


Figure 26: UGV Carrying UAV

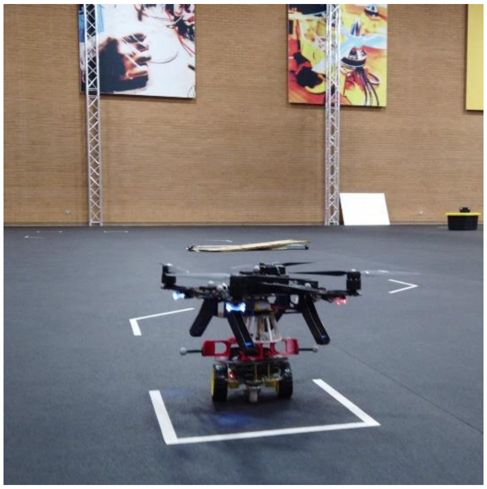


Figure 27: Final Position

at the second turn and changes direction. Once the UGV reaches the boundary of the second waypoint, it targets the last waypoint at the final position. When the UGV reaches the boundary of the last waypoint, it stops. This concludes the resting phase.

The resting phase was successful because the UGV was able to drive through two 45-degree turns while carrying the UAV. When changing direction, the UGV experienced a greater overshoot with longer settling time during this phase than it did during uncoupled driving, but it was still able to drive to its waypoints at approximately the same speed.

Conclusion

Summary

The selection of robots meets the design requirements of the heterogenous team of robots. The UGV's small, acrylic frame make it light and easy to lift. The UGV's four points of contact with the ground make it stable to land and ride on. The UAV's payload capacity allows it to lift the UGV. The UAV's small size makes it maneuverable in a hallway.

The EM interface, the FM interface, and the coupling maneuver meet the design requirements of the coupling mechanism. The EM interface is fixed over the UGV's center of mass to preserve the driving dynamics of the UGV. The electromagnetic force guides the coupling points to the coupling position. The electromagnet's 5-kilogram holding force creates a bond strong enough to lift the UGV. The EM interface's circular ring stabilizes the connection. The EM interface's low profile prevents interference with the UAV's rotors. The FM interface is fixed below the UAV's center of mass to preserve the flying dynamics of the UAV. The non-permanent magnet causes no residual pull when the electromagnet is disengaged. The FM interface's tall profile prevents physical interference with the landing gear and prevents tracking interference in the coupled position. The maneuver to reach the coupling position occurs in a single step and preserves the other control outputs of the UAV. The process to engage and disengage the coupling mechanism is non-mechanical and occurs instantaneously.

The offboard position control allows the robots to autonomously navigate through a list of waypoints. The communication through publishers and subscribers allows the robots to coordinate and perform complex procedures. The motion capture space provides a controlled environment and accurate localization to isolate the performance of the hardware during the experiment.

The results of the experiment conclude that the heterogeneous team of robots fulfills the performance requirements described in the problem statement. The UAV successfully followed the UGV through two 45-degree turns. The UAV successfully coupled with the UGV. The UAV successfully lifted the UGV over the obstacle. The UGV successfully carried the UAV through two 45-degree turns.

The fulfillment of the design requirements and the demonstration of the performance validate the symbiotic UAV and UGV relationship. The UAV and UGV shared their strengths to perform a complex task. The robots achieved more working together than they could on their own but cost less than a single robot with the same combination of capabilities. The robots' capabilities can be applied to search and rescue missions in disaster situations to reduce the time that human workers spend in harm's way.

Future Work

Improvements could be made to this system to address the diminished control of the robots under load. Before lifting the UGV, the UAV could adjust its controller to account for the dynamics of the weight of the UGV and its lower center of mass. Before carrying the UAV, the UGV could adjust its controller to account for the dynamics of the weight of the UAV and its higher center of mass. Stabilizing the control of the robots under load will improve energy efficiency and enable more complex maneuvers.

The heterogenous team of robots could adapt to perform similar tasks. With a more stable controller for coupled flight, the UAV could lift the UGV up a flight of stairs to navigate buildings with multiple floors.

This coupling mechanism could be modified to enable power sharing between the robots. The UGV could share battery capacity with the UAV when resting to increase the UAV’s flight time. The UGV could share power with the UAV when lifting to increase the UAV’s thrust.

The heterogeneous team of robots could be expanded to perform different tasks. A large suite of sensors could be distributed across many small robots. Some robots could be outfitted with end effectors to open doors or move debris. Other robots could carry supplies to the victims.

The concept of using a heterogenous team to perform complex tasks could be applied to other systems. Nanorobots are limited by their size and complexity. A heterogeneous team of nanorobots could perform complex tasks while adhering to those constraints.

References

Antener, A. (2015, December 3). *setpoint_position_demo.* Retrieved from GitHub.com: https://github.com/mavlink/mavros/blob/master/test_mavros/scripts/setpoint_position_demo

Bovbel, P. (2017, November 29). *vrpn_client_ros.* Retrieved from GitHub.com: https://github.com/ros-drivers/vrpn_client_ros

Fried, L. (2017, May 31). *Adafruit_MotorHAT 1.4.0.* Retrieved from pypi.org: https://pypi.org/project/Adafruit-MotorHAT/#data

Guermonprez, P. (2018, December 12). *Autonomous drone programming with ROS*. Retrieved from GitHub.com: https://github.com/intel-aero/meta-intel-aero/wiki/05-Autonomous-drone-programming-with-ROS

Petrovic, T., Haus, T., Arbanas, B., Bogdan, S., & Orsag, M. (2015). Can UAV and UGV be best buddies? Towards heterogeneous aerial-ground cooperative robot system for complex aerial manipulation tasks. *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 238-245.

Singhvi, A., Biyani, S., & Artemiadis, P. (2019). *Multi Robot Formation Control using Motion Capture System.* Human Oriented Robotics and Control Lab.

Willee, H. (2019, May 9). *Using Vision or Motion Capture Systems for Position Estimation*. Retrieved from dev.px4.io: https://dev.px4.io/v1.9.0/en/ros/external_position_estimation.html