**1. Project description**

**1.1 Purpose.**

Users of Unmanned Surface Vehicles (USVs), such as first responders in a marine mass casualty search and rescue scenario, can better take advantage of the robots with an overhead view. As the USVs increase in distance from the operators, the depth perception makes it difficult to discern the situation. It has been demonstrated in [2] that a quadcopter Unmanned Aerial Vehicle (UAV) positioned to keep the USVs in camera view can provide an effective elevated view for improved decision making and situation awareness. This project aims to demonstrate that using communication, rather than image processing, can support path prediction to better keep the targets in the field of view. The target robot system that is a group of EMILYs (USVs) and a DJI Phantom (UAV). The project will be demonstrated in the Morse simulator but with easy transfer to hardware in mind. In the future, it could either be implemented as a script for a ground control station (GCS), such as Mission Planner, or in a companion computer directly attached to a UAV.

**1.1 Implementation Overview.**

The project is focused around a core Follow\_USVs behavior that keeps the USVs in view by following the centroid of the USVs, but periodically adjusting the standoff distance, vehicle yaw, camera tilt and camera zoom to keep the entire group in camera view. The algorithm makes the assumption that the USVs are traveling and working as a group and thus are close enough to each other that "following" the group is feasible. Widely dispersed USVs would be better monitored by a behavior that cyclically visits each subgroup. Follow\_USVs would be a good fit for a system where a specific tracking behavior is selected depending on mission characteristics and spread of the USVs. Rather than extending a single existing algorithm, Follow\_USVs take inspiration from two sources. The motivating scenario is described in [2], where a UAV-UGVs team aided human operators with an overheard view of the USVs. However, in [2] the UAV went directly to a target location where the USVs are programmed to end up. In this project, the UAV autonomously repositions to keep USVs in view. In [1], a UAV followed the group of surface vehicles by predicting the path of the group's centroid and calculating a standoff distance that allowed constant sensor coverage of the targets. However, the UAV was a fixed-wing and is subject to movement constraints that do not apply to the quadcopter. The concepts of centroid path modeling and standoff distance selecting is to be adapted for an algorithm that takes advantage of a quadcopter’s maneuverability freedom.

The Follow\_USVs behavior is based around predicting and following the path of the USV group's centroid. The UAV is listening for periodic updates of position and trajectory (waypoints) that are broadcast by the USVs. This information can be used to model the speed of the centroid and construct a set of waypoints that corresponds to the centroid's predicted trajectory. By following this path, the UAV's camera captures ahead of the overall group direction. This can help first responders see where the USVs are going and help better guide them.

Determining the standoff distance from the centroid is critical to keeping the USVs in view and promoting quality imagery. When the USVs are close together, the standoff distance can be reduced and the UAV can capture closer imagery and a less distorted perspective. However, when the USVs are further apart, the distance must increase and the camera must tilt upward to keep USVs in view. This component of the algorithm will borrow from [1], which demonstrated choosing an appropriate distance based on sensor angle, overall speed of targets, target spread, and error margins.

The predicted path can also reduce the amount of UAV movements in order to reduce the energy consumption. Instead of constantly repositioning the UAV based on the current state of the USVs, the UAV can select a position that is expected to keep the USVs in view for some duration. While the UAV is in position, it can continue to track the USVs by tilting and zooming the camera while adjusting vehicle yaw.

The actual task of following the centroid is performed by setting waypoints and adjusting the speed of the UAV using a simple potential field approach to maintain a target standoff distance. If the UAV's distance from the centroid is less than the target standoff distance, then the UAV reverses with speed proportional to its distance from the standoff distance. If the UAV is further than the standoff distance, it speeds toward the standoff distance, proportional to the distance. In either case, the speed is limited by a maximum speed parameter.

By making use of USV trajectories, the UAV will be able to plan ahead for smoother behavior. For example, if the USVs reach their destination and begin tending to victims, the UAV will be expecting this and will not fly off even though it has not yet obtained a position notification that tells it the USVs are now behind. The UAV should slow down and halt, while keeping the USVs in view. To deal with disparities between predicted and actual USV locations, an extended Kalman filter should be applied to construct a more accurate prediction over time.

The ability to plan ahead also impacts the amount of network communication required for stable operation. In order to keep bandwidth usage low, it was decided that the UAV should not send messages to the USVs. While two-way communication could potentially aid path planning, this approach requires less system requirements (for portability) while reducing the network traffic. Waypoints should be sent less often than position updates, but waypoints still require periodic retransmission since no handshaking ensures that anything was listening when the waypoint was initially announced.

In order to not interfere with other possible constraints such as FAA regulations or a desire for smoother imagery, Follow\_USVs do not modify the UAV altitude. However, other components could be in place to alter the altitude based on the output of Follow\_USVs.

**1.3 Programming Language.**

The Python language was chosen because it is well suited for working with the Morse simulator and is supported in Mission Planner GCS. Morse is a very flexible simulation environment that is rendered in a 3D Blender world. Prototyping is well supported as the Morse system allows the creating of customized robots by selecting a base platform and adding sensors and actuators such as "RGB Camera", "Pose", and "Waypoint". Aqueous environments are not readily available, so UGVs are being used as analogue. Environmental factors such as wind and waves impact the path of USVs. These factors are not explicitly modeled in simulation, but an error margin will be used to represent uncertainty when considering whether a UGV is in view.

**1.4 Equipment.**

**1.4.1 Equipment for Development.**

Because of the number of robots required, the project is being done entirely in simulation (for now). The team members will be using their personal computers for development. The required development tools are the Morse simulator, python3 and Git for version control.

**1.4.1 Equipment for Execution.**

The Morse simulator and python3 are required in order to execute the simulation. The hardware port could be done in a number of ways, but at minimum would require multiple (smart) EMILYs and a DJI Phantom. Technically, a single EMILY could be used, but it simply would not demonstrate the full extent of the algorithm.

**1. Inputs.**

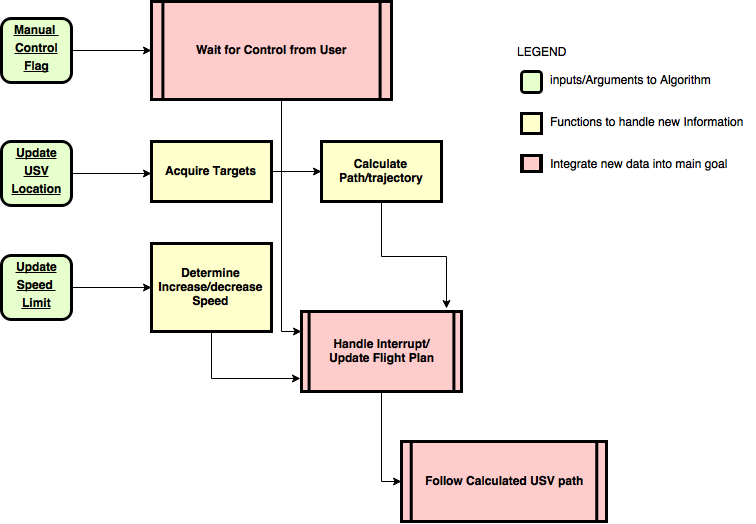
The algorithm depends on certain states and parameters of the UAV as well as periodic updates on the positions and trajectories of the target USVs. It is expected that the UAV's current position, altitude, and speed are accessible. The current position and waypoint messages are periodically sent to the UAV. This could be done through Mission Planner or a direct radio transmission, depending on whether the hardware setup is based on GCS scripting or an onboard companion computer. In simulation, these fields are provided through the Morse environment. The maximum speed is a parameter to the algorithm; it should be able to be explicitly input by the user (or another system component) so that the robot's actual limit can be restricted. When simulating the messages broadcast from the USVs, periodic transmissions will be enforced even though the UAV could "cheat" and check the values straight from the simulator environment.

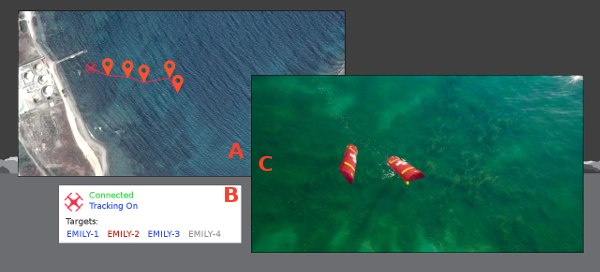
**3. Outputs.**

**3.1 Outputs.**

Follow\_USVs should periodically report the system state, such as the current UAV waypoint, standoff distance, predicted centroid trajectory, and which targets are in the camera field of view. The UAV's speed, heading, and camera angle could be used to support a subsumption architecture where other system components modify, for example, the standoff distance to get additional environment targets in view. While not an output of the algorithm itself, the project is centered around the transmission of imagery from the onboard camera. Each periodic report should have the same fields, even if those fields have not changed since last time. This allows for converting each message to a row in CSV file that can be used to perform diagnostic analysis in the plotting environment of the user's choice such as Excel or R.

**3.1 Interfaces and Mockups.**

[](NULL)

[](https://raw.githubusercontent.com/ekrell/AI_Robots_multiAgent/master/documentation/mockup.png)

This project is to develop an algorithm that would be a component of a system. Rather than show the entire mission planning interface, this mockup shows a few specific aspects of an interface that would be relevant the project.

**Figure A:** Because Follow\_USVs predicts the path of the USV group's centroid, that path can be displayed as a set of waypoint which can be displayed on a map. This easily communicates to the users what the quadcopter is "thinking"; the operators can visually inspect if the path is reasonable.

**Figure B:** Deployment of the quadcopter should be very fast. The operators should be able to enable and disable the tracking with a single click and have a corresponding visual indicator. The list of targets is color coded to easily see what targets are in view (blue), not in view (red), and which are not being tracked (grey). An operator may have a specific reason for a USV to leave the the group, and can click its name to toggle between tracking and not tracking that target. An untracked target is not considered when calculating the group centroid and generating a path prediction.

**Figure C:** The streaming imagery from the quadcopter camera. The goal of this project is that the USVs will always be in view.

**4. Use case(s).**

**Scenario: Marine mass casualty**

The capsized vessel has come into view and number of people can be seen struggling to hold onto its rapidly sinking hull. Others are nearby on floating cargo and other structures. The first responders on board are able to deploy the EMILYs, but are far enough that their view is at a low angle. Also, some parts of the disaster scene are occluded by the floating structures. The team decides to send out a group of EMILYs, but also have the quadcopter follow them so they can better assess the situation. While the EMILYs are being let out into the water, the quadcopter operator opens his laptop where Mission Planner is already running, and the operator selects the Follow\_EMILYs script. The maximum speed is set from a previously setup configuration file, and the operator is prompted to select which EMILYs are part of the target group. The operator hits 0 as code for 'all' and the quadcopter arms, elevates and accelerates toward its first calculated waypoint. The EMILYs begin making their way toward the victims, and the quadcopter periodically repositions to maintain them in the field of view. The first responders can watch the EMILYs travel to the site, and since the quadcopter is following from behind and looking ahead, they have a much better sense of the disaster scenario. Better informed, the team selects new waypoints to aid those most in dire need of assistance. All the while, the quadcopter provides a view with clear depth perception and no significant occlusions.

**Consideration:** Energy efficiency

In order to extend the possible flight time of the UAV, an important goal is to conserve the USV's battery through strategic positioning. Instead of constantly "tail-chasing" the group of USVs through frequent repositioning, the UAV hovers at a position where it predicts the USVs will be in view for some duration. Once in position, it maintains the USVs in view by tilting and zooming the camera, as well as turning the UAV. When the USVs are positioned more densely, the camera zooms in for a better view. Meanwhile, the camera tilt and UAV yaw follow the estimated centroid position.

**Exception: lost/delayed updates**

While undergoing a mission such as in Scenario A, one of the EMILYs is damaged and unable to broadcast. Using the periodic status updates, the UAV is able to alert the operators that it has lost communication with a particular target. After some interval, the UAV can focus on following the USVs that it is still receiving from but still listening for the lost USV unless the operators explicitly disable following that USV. If the USV has lost all communication with USVs, but still has communication with the base, then the operators will be notified and can direct the UAV accordingly. For example, they may choose to have the UAV return to base or control manually.

**Flexibility: Dynamic parameter adjustment**

During a mission, multiple EMILYs have reached their destination and halted while victims grab hold of the floatation. The UAV currently has a zoomed view of the USVs but does not provide enough of the surrounding environment of interest. The operator sends a command to enlarge the minimum field of view so that the UAV zooms out to reveal the desired scene. Minimum and maximum values for the following parameters can be adjusted while in flight to get the desired view because the algorithm re-checks the parameters at each cycle, not just at initialization.

**Contention: UAV given other tasks**

The proposed algorithm is very focused on the following behavior but is designed to support a subsumption architecture. At any time, other system components such as manual intervention or additional autonomous features could override this behavior. However, the path prediction is still being performed even if the outputs are intercepted and modified or discarded before reaching the motor components. Thus, once control is returned to the follow behavior, the UAV can easily get back on track since it has a next waypoint ready. Also, the subsumming components could use modify the outputs to maintain the view while performing other tasks.

**5. Test Plan.**

The goal is to determine if the quadcopter will be able to constantly keep a group of USVs within camera view by following the group through repositioning and camera tilting/zooming, using position and trajectory messages broadcast by the USVs. The hypothesis is that the trajectory messages will allow the UAV to make effective path predictions that keep the UGVs in view using less movements than if only current positions were communicated. Using less movements is important because it conserves energy and thus increases the available flight time. Also, fewer movements would result in a more stable image, which reduces the disorienting effect of a shaky image, and potential image processing applications would benefit from the improved image quality.

The algorithm will be tested in the Morse simulator, using a single quadcopter and variable number of ground vehicles. The quadcopter will have a speed actuator and an abstract 'Waypoint' actuator. The ground vehicles will also have speed and waypoint control. Both robot types will have a Pose sensor for localization, and the quadcopter will have an RGB camera with tilt and zoom control.

A constant starting location will be set as the "base", as if this is the ship or shore from which the USVs are deployed. Each run has a specific USV message broadcast rate for the current position and trajectory messages. The UGVs will be initialized from this location, but they will be positioned three meters apart from each other and the base. Each UGV will be loaded with a set of waypoints which represent its trajectory. The waypoints will be selected to be within 200 yards from the base, based on the expected maximum distance between base and victims. Also, the UGVs are assumed to be working as a group and staying close enough to each other to be within the camera's maximum FOV. The waypoints will be assigned such that the UGVs do not leave the maximum camera footprint, given the constant altitude. The UGVs are deployed, followed by the UAV. The experiment will run until all vehicles have reached their final waypoint or until the mission duration has gone overtime by 5% of the expected mission duration. Expected mission duration is calculated based on considering the total distance from each waypoint and the USV speeds.

The dependent variables of interest are the percentage of total time that all the USVs are kept in view and the amount of time that the UAV spends adjusting its position. The main independent variables are the position and trajectory message broadcast rates which relate to the hypotheses. Other independent variables are the number of UGVs, their trajectories, and speeds. To get meaningful results despite several independent variables, runs consist of two levels for combinatorial experiments. At level one, the "scenario" is set, which varies the waypoints, speeds, and group size. For each "scenario," a common set of message broadcast rates are tested.

**Dependent Variables:**

* time that all UGVs are kept in view
* time spent moving UAV

**Independent Variables:**

* Rate at which UGVs broadcast position
* Rate at which UGVs broadcast trajectory (to account for trajectory modifications)
* Number of USGs
* UGV trajectories
* UGV speeds

**Constants:**

* Base location
* UAV altitude

This experimental setup should demonstrate the effect of varying the message update rates. In order to compare the effect of trajectory information to using only UAV positions, one run will be performed for each scenario in which no trajectory information is sent. In addition to comparing against a position-only broadcast system, it models a "best case" of visual tracking in which no trajectory information is sent. In addition to comparing against a position-only broadcast system, it compares against the "best case" of visual tracking if the update rate is comparable to the identification rate. Within each scenario, statistical significance of mean time spent moving UAV while varying the broadcast rates would be obtained using a paired T test. The same would be done to test significance of time that all USVs are kept in view.

**6. References.**

[1] He, Z., Xu, J. X., Yang, S., Ren, Q., & Deng, X. (2014, June). On trackability of a moving target by fixed-wing UAV using geometric approach. In Industrial Electronics (ISIE), 2014 IEEE 13rd International Symposium on (pp. 1571-1577). IEEE.

[2] Xiao, X., Dufek, J., Woodbury, T., & Murphy, R. (2017). UAV Assisted USV Visual Navigation for Marine Mass Casualty Incident Response. International Conference on Intelligent Robots and Systems.

**Image Sources**

* Quadcopter symbol used in mockup: <https://thenounproject.com/term/quadcopter/22061/>
* Waypoint symbol used in mockup: <https://waypointbhs.com/>
* Background of fake UAV camera imagery: <http://www.redbull.com/en/adventure/stories/1331670732010/15-incredible-shots-of-action-and-adventure-with-a-drone-cam>
* EMILY robots placed into the fake UAV camera imagery: <http://emilyrobot.com/product-category/accessories/>

Map used for "Mission Planner" view: <https://maps.google.com/>