

Bootstrapping Multi-Agent Unmanned Aerial Vehicle (UAV) System Integration Using Ground-Based Assets: Lessons Learned

Matthew P. Vaughan* and Benjamin N. Kelley†
NASA Langley Research Center, Hampton, Virginia, 23666

In support of the Autonomy Teaming and TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) project, a fleet of unmanned ground vehicles (UGVs) was developed as a test and evaluation (T&E) platform to reduce system integration gaps between simulation and live flight hardware. While simulation and hardware-in-the-loop bench testing provide adequate environments for preliminary validation, differences in system deployment architecture, software interfaces, and hardware infrastructure increase the risks to safety, property, and the project. Given ATTRACTOR's goal of establishing a basis of certification of trust and trustworthiness in multi-agent autonomous systems, bridging these gaps was critical to successful project execution and feasibility assessment. In this paper we present the UGV fleet and its role in speeding up system integration, smoothing the transition from simulation to flight, and providing researchers an easy-to-use hardware test bed. An overview of the hardware and software on-board the vehicles is provided along with supporting infrastructure. The system integration process is documented including results in supporting both the overarching design reference mission (DRM) of ATTRACTOR and individual research efforts conducted since the creation of the fleet. Finally, we discuss the practical lessons learned regarding the testing, deployment, and operation of multi-agent autonomous systems.

I. Introduction

The highly dynamic nature of unmanned aerial vehicles (UAVs) imposes significant challenges when conducting initial testing ranging from safety risks posed by high-capacity lithium batteries and spinning propellers to rigorous timing demands on controllers and the consequences of failures mid-air. Flight testing of a single UAV is time and labor intensive due to these challenges and more, and the complexity increases exponentially with the number of vehicles. Simulations are an important early step towards validating complex systems before deployment. However, improving the fidelity of a simulation quickly becomes cost-prohibitive, leaving a gap between it and reality that must be carefully navigated. Traversing the gap during system integration and validation is one element of establishing trust and trustworthiness in cyber-physical systems, a problem explored by the Autonomy Teaming and TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) project.

ATTRACTOR [1][2] addressed the gap in the system integration process using two approaches: a novel mixed-reality simulation environment [3] and an intermediate hardware test platform. The simulation environment was engineered to blur the transition from simulation to flight hardware and leverages a hardware-agnostic software framework that allows any combination of real and virtual autonomous agents to interact with each other within a shared world model. This paper presents ATTRACTOR's intermediate hardware test platform, a fleet of unmanned ground vehicles (UGVs) developed by the Autonomous Integrated Systems Research Branch (AISRB) at NASA Langley. The UGV fleet offers several attractive benefits.

First, ground vehicles reduce or eliminate many of the risks and costs associated with operating aerial vehicles. The collection of small rovers avoids the logistical difficulty of coordinating multiple safety pilots allowing for faster iterations of testing and debugging. Since errors are less consequential on the ground, a more relaxed risk posture can be taken to permit earlier integration testing. Depending on the research, it may be acceptable for the UGVs to be used in lieu of airborne drones. In that case, less stringent size, weight, and power requirements translate into less expensive components, and the cost savings can be invested into additional vehicles or sensors.

*Research Computer Scientist, Autonomous Integrated Systems Research Branch (AISRB)

†Research Computer Scientist, Autonomous Integrated Systems Research Branch (AISRB)

The second benefit of using ground vehicles as an intermediate test platform [4][5] is that the hardware inherently captures certain properties of physical multi-agent autonomous systems through embodiment rather than explicit simulation. If the ground and air vehicles use a similar computation environment (e.g. an ARM-based processor running a Linux distribution), the effects of constraints such as network bandwidth and latency, processor speed, and available memory can be evaluated simply by running the software on the rover rather than modeling the constraints in a simulation. A UGV can also be equipped with the same sensor payload as a multirotor to evaluate the hardware and processing requirements for the sensors in a live albeit less dynamic environment.

Along with the benefits of utilizing the rover fleet, there are inevitably some drawbacks to consider. Even though the rovers are generally less expensive than similarly equipped drones, they still represent a nontrivial investment of resources to construct, write software for, and test. In the AISRB, the UGVs are valuable not only as an intermediate test analogue but also as a suitable target platform for applications such as human-machine teaming and swarm operations. The other consideration is applicability of test results from the rovers to the multirotors. To use the UGV fleet as an effective intermediate test bed, the amount of platform-agnostic software should be maximized. The software framework developed by the AISRB, which is summarized in section II.B and described in detail in [3], enables programmers to easily create modular capabilities for autonomous systems independent of any particular hardware solution. With the pros and cons in mind, we go on to discuss the actual fleet.

Section II describes the hardware used on each vehicle, the software architecture, and additional key infrastructure for operating the fleet. Section III discusses several research projects in which the rovers have been used in some capacity. Finally, section IV concludes with lessons learned in operating multi-agent autonomous systems and future work for the UGV fleet.

II. UGV Fleet Overview

A. Rover Hardware

Several criteria factored into the hardware selection for the individual rovers comprising the UGV fleet. First, the on-board computing resources needed to be easily programmable in order to deploy and run custom autonomous algorithms with varying sensor suites. Second, the vehicles were required to connect to a wireless network for communication with each other and a ground station operator. Finally, low level control algorithms for complex ground vehicles were not the intended use case for the fleet, so a simple and easy-to-control hardware solution was preferred. The GoPiGo3™ by Dexter Industries was chosen as the base platform for the UGV fleet. The GoPiGo3™ is a differential drive robot that uses a Raspberry Pi™ single board computer (SBC) to interface with its motors and other peripherals. The robot kit includes open source driver libraries in several programming languages allowing developers to create custom applications.

Because the GoPiGo3™ kit includes motors with wheel encoders as well as the electronics needed to drive them, the only hardware changes to the nominal vehicle configuration were using a lithium polymer battery, attaching spherical Vicon™ motion tracking markers, and mounting a forward-facing Raspberry Pi™ camera. Twenty vehicles were constructed using this configuration shown in Fig. 1 with five reserved for spares and additional modifications to the sensor payload. To date, two alternate configurations have been created to satisfy significantly more demanding sensing requirements. Figure 2a shows a GoPiGo3™ outfitted with a Hokuyo™ lidar and VectorNav™ inertial measurement unit (IMU) to test sensor hardware interfaces and a state estimation pipeline before deploying them on a UAV. In Fig. 2b, an Intel Realsense™ depth camera and Phidgets™ IMU were mounted on the rover as a much less expensive sensor payload to explore visual-inertial localization and mapping.

B. Command and Control Architecture

The software for the rover fleet builds on the Autonomous Entity Operations Network (AEON) framework. As described in [3], the AEON framework facilitates software development for autonomous agents through a collection of utilities for writing concise, modular capabilities and connecting them through well-defined interfaces using the Data Distribution Service (DDS) [6] as the inter-process communication protocol. Figure 3 illustrates the core processes used to operate a single rover and how they instantiate the AEON design principles of modularity and layered functionality. Each rounded rectangle represents a standalone DDS-connected process (or node) and ovals represent hardware elements. The hardware interface layer exposes sensors and actuators to the DDS network, publishing data and receiving commands to and from higher layers. The next layer encompasses low level control, which turns target waypoints into steering and

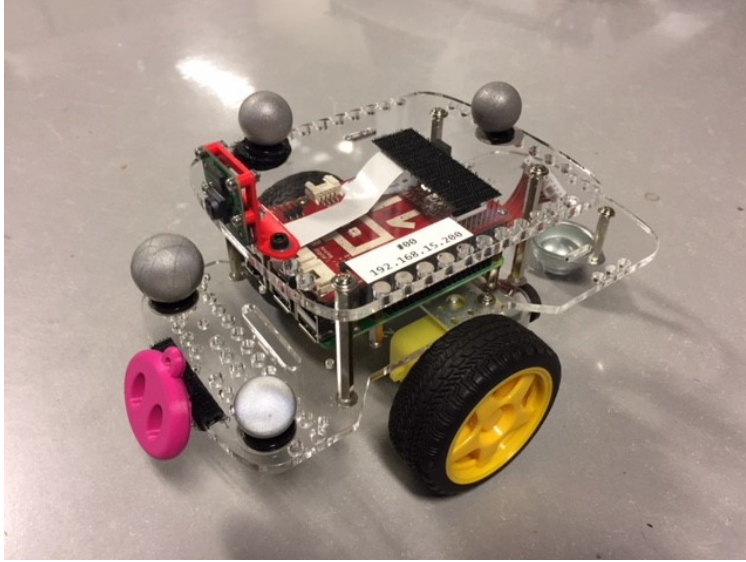
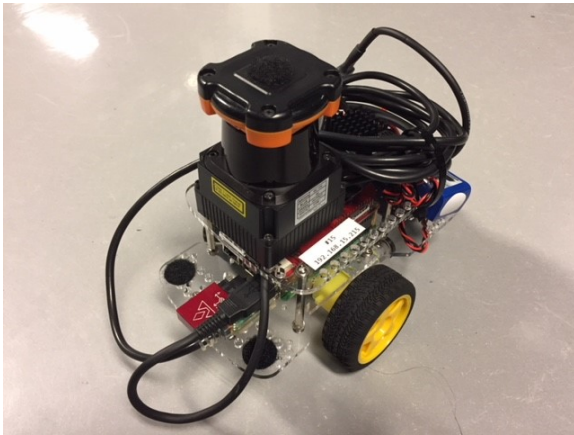
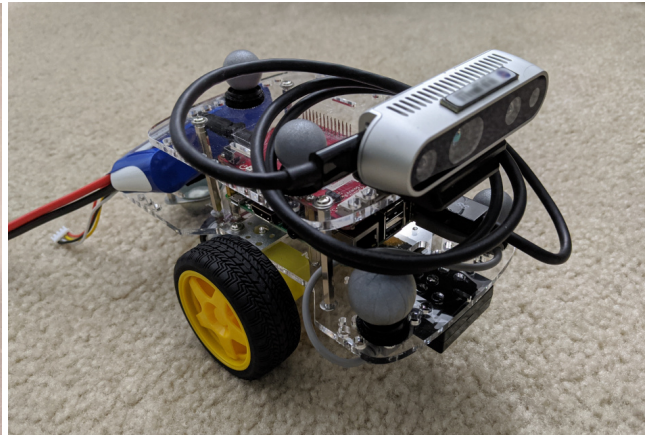


Fig. 1 The nominal GoPiGo3™ configuration with Vicon™ markers and camera



(a)



(b)

Fig. 2 Alternate sensor payloads: (a) lidar and IMU, (b) depth camera and IMU

drive commands, and sensor processing such as feeding a video stream into an object detection algorithm. For the purposes of the use cases described in section III, the top layer of autonomy for the fleet is some form of waypoint generation and dispatch.

The GoPiGo3™ DDS interface node is the key piece of software enabling the rover to be used as an autonomous agent within the AEON framework. The library provided by Dexter Industries allows a program running on the Raspberry Pi™ to command the speed of the two drive motors, read their encoder values, and communicate with any other attached peripherals. The DDS interface wraps the library allowing other nodes on the network to access its functionality. The node accepts steering and drive commands and converts them into target wheel speeds using the geometry of the differential drive robot. Encoder state is continuously published for use in sensor fusion algorithms, and configuration parameters can be queried and set dynamically such as the maximum speed and the red-green-blue values of the two light emitting diodes (LEDs) on top of the GoPiGo3™.

The waypoint navigation node produces steering commands for the GoPiGo3™ interface to consume. The node accepts DDS messages containing one or more goal waypoints and implements two very simple proportional-integral-derivative (PID) controllers, one for the heading error and one for the forward velocity. The current pose of the rover is provided over DDS by either Vicon™ or a state estimation algorithm. Controller parameters include the gains for the

PID controllers, the radius inside which a waypoint is considered to have been reached, the maximum heading error above which the rover stops and turns in place, and the maximum forward and turn rates. These parameters can be set dynamically using DDS to tune the controller and saved to a configuration file, making the navigation node usable on other rovers than the GoPiGo3™.

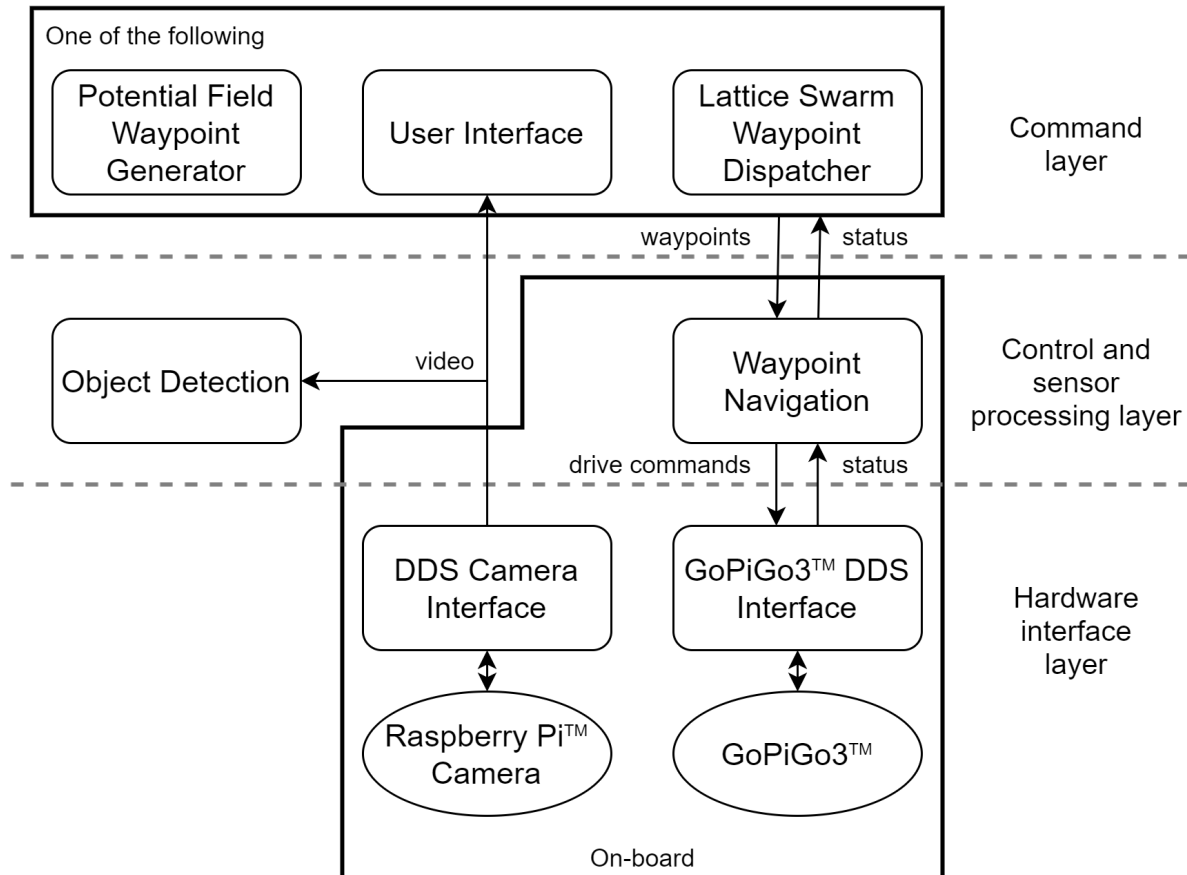


Fig. 3 Hardware and software diagram for commanding a single rover

C. Supporting Infrastructure

1. Software Utilities

Prior to the UGV fleet, autonomous systems operations in the AISRB only utilized up to four simultaneously running agents. Simple tasks such as updating software or starting DDS nodes that used to be done manually quickly became tedious with twenty vehicles to manage. An existing utility script for deploying libraries and DDS nodes to a vehicle was upgraded to support multiple destinations. Another script allows users to quickly reboot or shut down many vehicles simultaneously. To alleviate the burden of typing dozens of commands to spin up demonstrations involving many agents, process management code from an existing project was broken out into a standalone node to start, stop, and query processes using messages sent over the DDS network. The process manager was installed on each vehicle and configured to start on boot, and a utility script was created to read extended markup language (XML) files containing process definitions and send them to the process manager to start or stop.

In addition to the infrastructure automation, a graphical user interface (GUI) was created to test and debug the rover hardware interface and control layers shown in Fig. 3. The GUI, shown in Fig. 4, allows the user to select any vehicle connected to the DDS network, view and tweak controller parameters, send waypoint commands by clicking on the central plot, and view sensor data including vehicle pose (red chevron), lidar scans (black dots), and video.

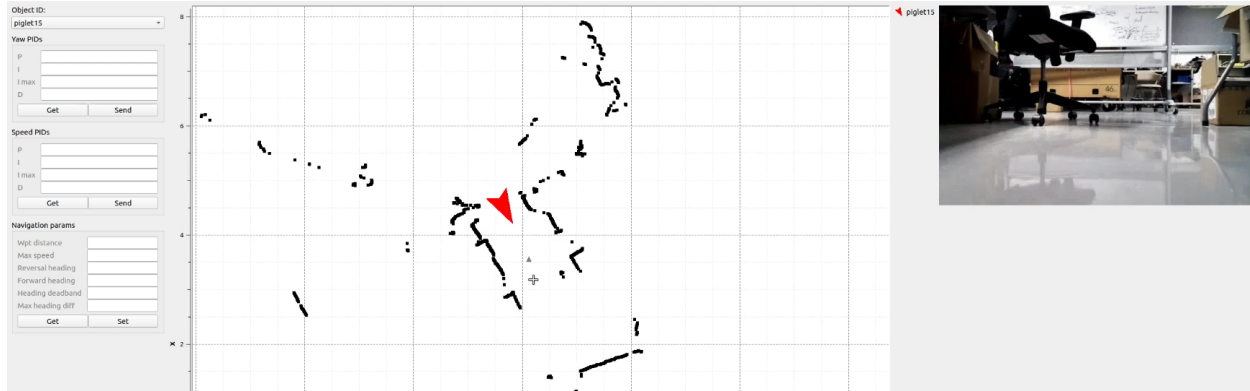


Fig. 4 Rover engineering GUI showing controller parameter fields, a lidar scan, and a video feed

2. Vicon

The indoor flight range at the AISRB is outfitted with a ViconTM motion capture system. The accompanying tracking software generates high quality, high speed position and orientation information for all registered objects in the range. Depending on the goals of a specific research project, the ViconTM data may serve either as the primary source of object pose information or as a placeholder for early testing of downstream consumers of that information. ViconTM data can also be used as ground truth for analyzing the performance of controllers and state estimators. The UGV fleet posed a unique challenge for the ViconTM system due to the vehicles' size and number. The ViconTM object tracker requires unique 3D marker configurations to define and distinguish objects in the environment, but it was impractical to come up with twenty clearly distinguishable patterns on the small rovers.

To track the large number of small robots, a custom tracker was developed inspired by the approach in [7]. Each rover was given an identical marker configuration, and the Vicon server was configured to publish raw marker positions instead of object tracks. After an initialization procedure, incremental tracking was performed using Iterative Closest Points (ICP) [8] on the position data. Unlike [7] which performed the track initialization using a configuration file with fixed starting locations, we developed a GUI to manage and visualize the tracking. New marker configurations can be specified either by manually adding them to a dictionary of templates or by selecting a set of markers in the visualization and saving them to the file. Initialization can then take place at any time by selecting a template from the dictionary and clicking a cluster of markers to register the template against. After a successful registration, object pose data from the incremental ICP tracker is streamed out over DDS. Figure 7b shows the interface being used to track and monitor the UGVs performing the coordinated swarm demonstration described in section III.B.1.

3. Auxiliary Computing Resources

Operating the fleet currently requires at least two additional computers: a ground station computer and a server running the ViconTM system. The ground station is used to start the requisite DDS nodes on each robot and runs one of the command generation nodes at the top of Fig. 3. The nominal rover configuration is not yet capable of producing its own state estimate, so the ground station also runs the custom ViconTM tracker described above. The tracker listens to the server that runs the ViconTM system and broadcasts marker positions. Depending on the demonstration, additional computing resources may also be running on the network. For example, the deep neural network (DNN) performing object detection for the ATTRACTOR design reference mission (DRM) is too large and computationally intensive to run on the Raspberry PiTM, so the video feed is sent over DDS to a desktop that runs the DNN and sends back the results. Running the mixed-reality simulation is another example that pulls in additional machines for users to interact with the environment. Figure 5 shows a typical deployment of computing resources when operating the UGV fleet.

III. Use Cases

The UGV fleet has proven to be a versatile platform for multi-agent autonomous systems research. While the original goal of the fleet was to create a safer test bed for validating new algorithms before flight, the following use cases integrated the rovers into the final demonstrations either as the sole vehicle type or as part of a heterogeneous network of multirotors and ground assets.

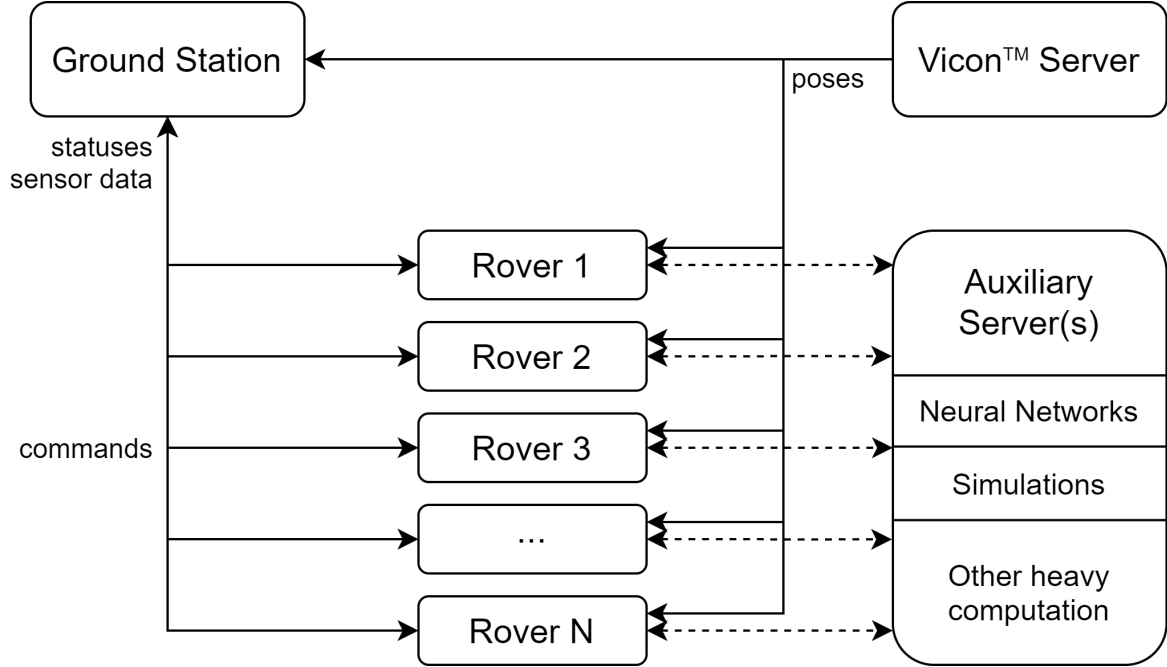


Fig. 5 Deployment diagram for operating the rover fleet

A. ATTRACTOR DRM: Search and Rescue

The design reference mission for ATTRACTOR was a search and rescue task where a human operator deploys multiple autonomous agents to explore various locations. Aerial vehicles would survey outdoors and inside a large open room and then hand off the search to ground robots to continue in a more constricted indoor space. The goal of the mission was to integrate many of the capabilities developed throughout the project including the simulation environment, task management, trajectory generation, state estimation, and object detection. Of particular interest was integrating into the AEON framework the lidar-inertial state estimation pipeline developed in [9] as part of a previous collaboration between NASA Langley and MIT. An accurate state estimator that runs entirely on-board a vehicle would greatly enhance the AISRB research portfolio by extending its capabilities into GPS-denied areas and beyond the Vicon™ range.

The DRM required both aerial and ground vehicles to utilize the state estimator. The rover shown in Fig. 2a was equipped with the same model of lidar and IMU used on the quadrotors in [9]. Working first with the UGV permitted faster and more aggressive testing of the estimation pipeline because the risk to the expensive sensor payload was very low. The scene in Fig. 4 shows the rover using the state estimate to navigate to waypoints in a cluttered office environment with the latest lidar scan overlaid on the plot and the view shown from the forward facing camera. Figure 6 highlights how the modular capabilities created using the AEON framework enable a substantial amount of code sharing across platforms. Only the GoPiGo3™ hardware interface and the waypoint controller are platform-specific, meaning that most of the autonomous software that would run on the multirotor can undergo integration testing in a real environment before the first takeoff.

B. Other Projects

1. Lattice Swarm

An early use case for the ground vehicle fleet was to perform a hardware demonstration of the work in [10]. The distributed algorithm selects members of a swarm of agents in a lattice formation to break off from the main group to perform some task. The algorithm preserves connectivity of the swarm by creating a trail of agents from the main group to the subgroup and by continuously filling in gaps formed as the subgroup departs. Because the initial work used a 2D point simulation for the swarm, the goal of the hardware demonstration was to show preliminary feasibility of the approach on real robots. To bridge the gap between the rovers and the MATLAB™ simulation running the algorithm,

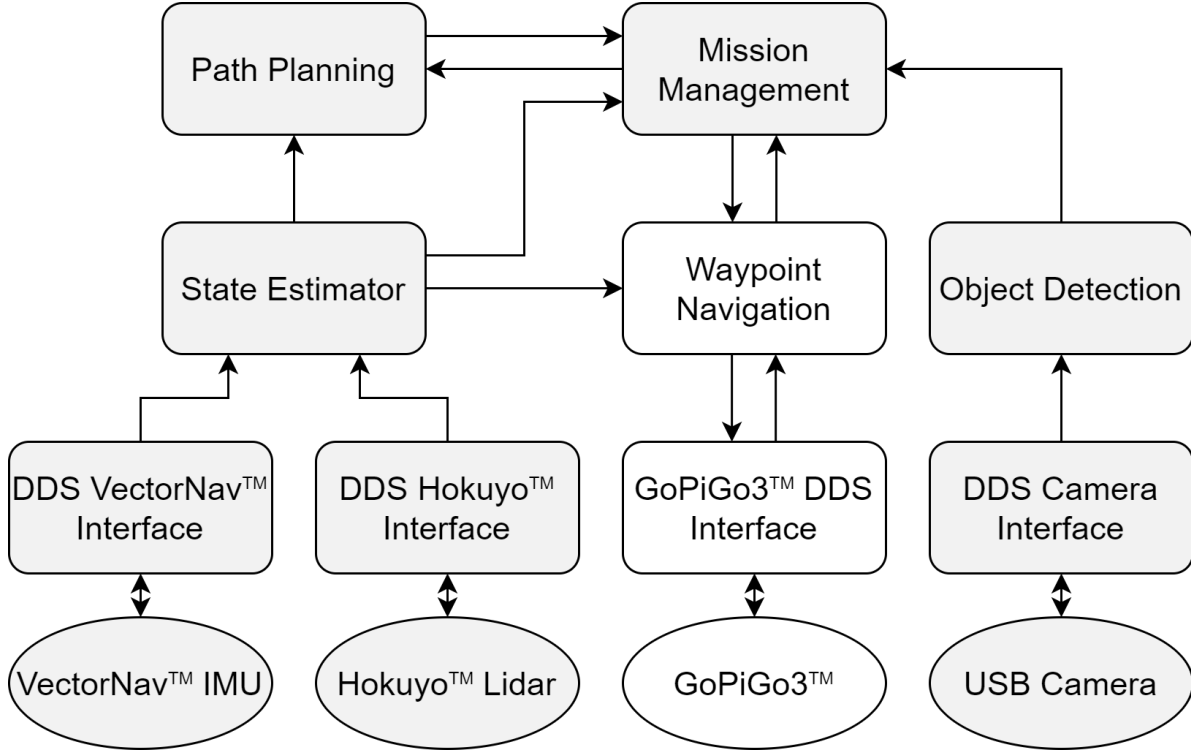


Fig. 6 Software diagram for the ATTRACTOR DRM. Shaded boxes are shared by the UAV and UGV.

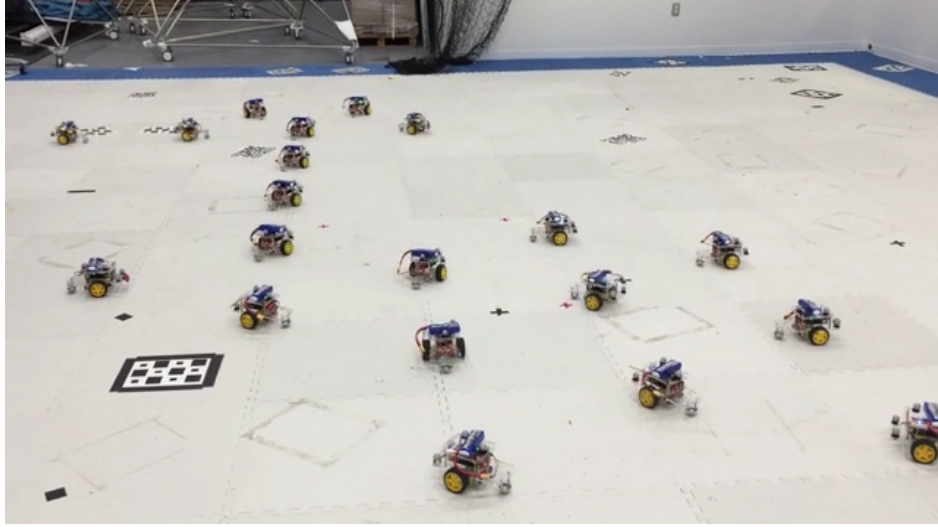
the latter was modified to output a file containing waypoints for each member of the swarm at each time step. A DDS node was written to consume the file and dispatch the waypoints to each vehicle, sending the next time step's waypoints as soon as every vehicle reached their current goal. As shown in the top right corner of Fig. 3, the waypoint dispatcher node sits at the command generation layer and feeds waypoint commands to each rover's waypoint-following controller. Figure 7a shows the demonstration with a sub-swarm breaking off in the top left and a line of rovers connecting it to the main group.

2. Potential Field Exploration

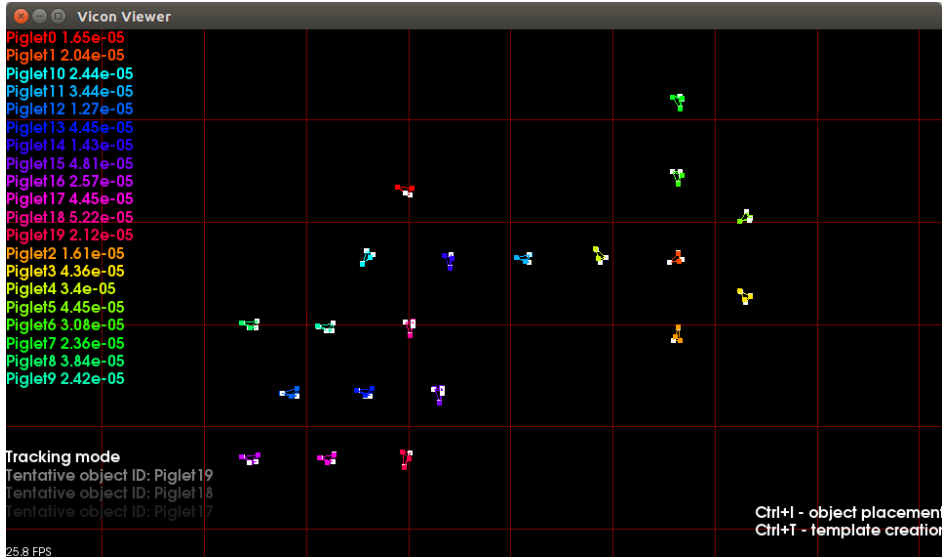
The next use case for the fleet was another hardware demonstration, this time for the exploration algorithm presented in [11]. The algorithm uses a potential field to steer the vehicles away from each other and towards under-explored regions of a target area. The potential field is constructed from two sets of terms. First, a grid of interest points is overlaid on the area, and each point increasingly attracts agents based on the time since the last visit by any agent according to a simple dynamic model. The second set of terms repels agents from each other to avoid collisions and spread out the swarm. Figure 8 shows the live demonstration of eight robots with the interest points visualized in simulation. Green circles indicate sufficiently visited areas, and the robots can be seen moving towards redder regions. Like the lattice swarm demonstration, the rover fleet sufficed to show the algorithm in action, eliminating the overhead and risk associated with flying eight drones simultaneously.

3. Simultaneous Localization and Mapping (SLAM)

The vehicles used by the AISRB generally rely on external sources of state information whether it is GPS outdoors or Vicon™ indoors. The lidar-inertial state estimation pipeline used in the ATTRACTOR DRM performs well and offers a viable on-board solution for larger vehicles, but the cost, weight, and power requirements of the sensor payload prevent it from being deployed on smaller multirotors and the full rover fleet. In contrast, vision-based SLAM algorithms are alluring due to relatively modest sensor hardware requirements that can be met on smaller, less expensive vehicles. The tradeoff in this case is far more demanding sensor processing, but advances in SBC performance mean that previously nonviable algorithms can be revisited. In [12], results were presented from field testing two open source, off-the-shelf



(a)



(b)

Fig. 7 Rover fleet performing the lattice swarm demonstration (a) with associated object tracking (b)

monocular visual odometry solutions. Initial exploration is underway to update the work with more recent algorithms and expand the evaluation to include stereo vision and IMU fusion approaches. The Intel Realsense™ camera mounted on the rover in Fig. 2b can produce monocular, stereo, or depth images, making the vehicle a useful test platform for a variety of visual SLAM algorithms.

IV. Conclusion and Future Work

For ATTRACTOR, a new fleet of unmanned ground vehicles was used to advance research in multi-agent autonomous systems. The rovers serve as an effective test analogue in the transition between simulation and the target aerial platform during the system integration and validation process. As long as the software architecture is designed to be sufficiently modular and hardware agnostic, autonomous operations can be tested on ground robots at much lower risk with far less time and manpower to operate. For autonomous systems research in both the aeronautics and space domains, the rovers can also serve as the target platform to show initial feasibility of the work on real systems faster and at lower cost than using aerial vehicles.

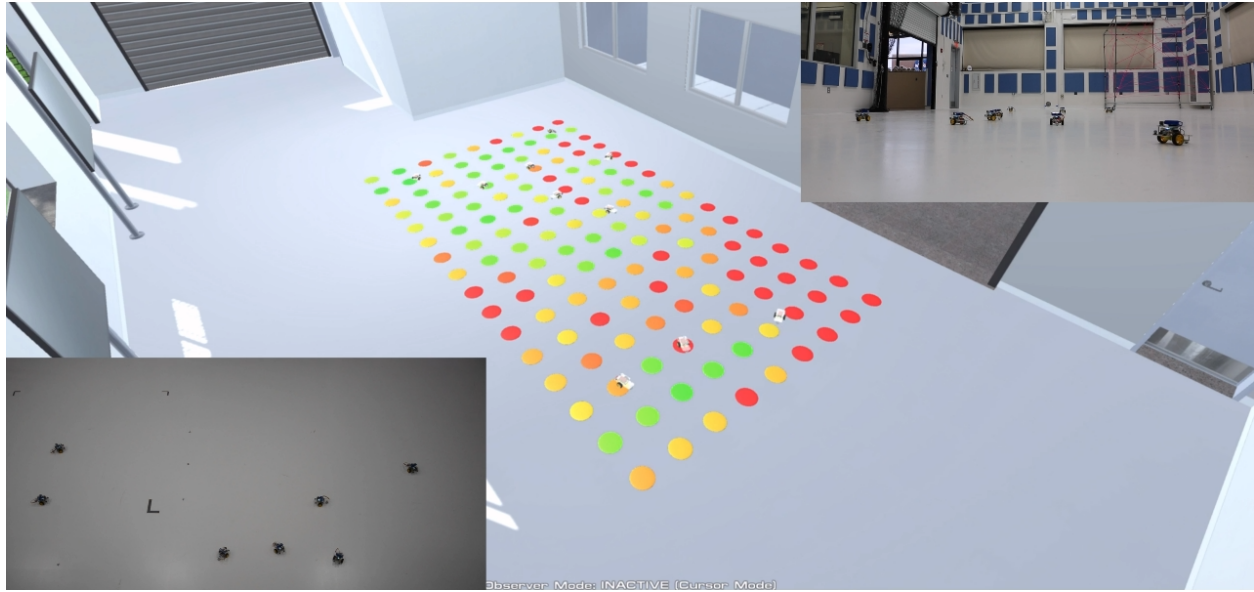


Fig. 8 Potential field demonstration with grid point visualization in simulation

The robot hardware itself is very simple and only consists of minor additions to an inexpensive but reliable educational kit. As a result, a sizable fleet could be constructed for both multi-agent and swarming research applications. The key enabling technology is the AEON software framework used by the AISRB to build up increasingly more sophisticated autonomous systems, but the progress flows both ways. Just as AEON facilitated the creation of the UGV fleet, lessons learned while operating the fleet have been incorporated into the software framework to improve future operations. Various utility scripts were written to speed up repetitive tasks such as code deployment and rebooting computers. The process manager provides hooks to remotely start, stop, and poll processes using DDS messages, easing the burden of initializing and shutting down a large fleet of vehicles. Libraries for sharing coordinate frames and configuration parameters over DDS were developed to assist with monitoring, debugging, and maintaining a consistent world model across agents. Even the unidirectional publish-subscribe DDS protocol was extended to provide two-way communication channels, improving the reliability of the fleet by enabling application-level acknowledgement of messages.

Two natural threads of future work are upgrading the rover hardware and adding new software capabilities. Continuing to explore alternate sensor payloads will not only increase the diversity of possible missions but also create more opportunities for research such as task planning for large groups of heterogeneous assets. As mentioned in section III.B.3, empowering vehicles to operate outside of the Vicon™ range will be one of the top priorities for future additions to the AEON framework. Another avenue of future work will be to decentralize the command structure away from a single hub (the ground station) with many spokes towards a local mesh communication protocol. Finally, the highest level of autonomy for most of the use cases described above was executing a script that produced waypoints for the vehicles to follow. We will increase the autonomy of the rovers by expanding the command layer in Fig. 3 with behaviors such as searching an area or following a target and adding a planning layer to turn high level objectives into series of behaviors to execute.

Acknowledgments

The authors would like to thank the entire ATTRACTOR team, led by Dr. B. Danette Allen and Dr. Natalia Alexandrov, and the Convergent Aeronautics Solutions (CAS) Project of the NASA ARMD's Transformative Aeronautics Concepts Program (TACP) for supporting this work via ATTRACTOR.

References

- [1] Harrington, J. D., “NASA Selects Three Aeronautics Teams to Explore ‘Ambitious’ Ideas,” NASA, 2017. URL <https://www.nasa.gov/press-release/nasa-selects-three-aeronautics-teams-to-explore-ambitious-ideas>.
- [2] Allen, B. D., “Real-World and Mixed-Reality Applications of Trustworthiness and Trust in Autonomous Cyber-Physical-Human Systems,” *AIAA Scitech 2021 Forum*, (submitted for publication).
- [3] Kelley, B. N., and Vaughan, M. P., “A Distributed Simulation-to-Flight Framework to Support Investigating Trust/Trustworthiness in Multi-Agent Systems,” *AIAA Scitech 2021 Forum*, (submitted for publication).
- [4] Michael, N., Fink, J., and Kumar, V., “Experimental Testbed for Large Multirobot Teams,” *IEEE Robotics Automation Magazine*, Vol. 15, No. 1, 2008, pp. 53–61. <https://doi.org/10.1109/M-RA.2007.914924>.
- [5] Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M., “The Robotarium: A remotely accessible swarm robotics research testbed,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1699–1706. <https://doi.org/10.1109/ICRA.2017.7989200>.
- [6] Pardo-Castellote, G., “Omg data-distribution service: Architectural overview,” *23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.*, IEEE, 2003, pp. 200–206.
- [7] Preiss, J. A., Honig, W., Sukhatme, G. S., and Ayanian, N., “Crazyswarm: A large nano-quadcopter swarm,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3299–3304. <https://doi.org/10.1109/ICRA.2017.7989376>.
- [8] Arun, K. S., Huang, T. S., and Blostein, S. D., “Least-Squares Fitting of Two 3-D Point Sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, 1987, pp. 698–700. <https://doi.org/10.1109/TPAMI.1987.4767965>.
- [9] Tian, Y., Liu, K., Ok, K., Tran, L., Allen, D., Roy, N., and How, J. P., “Search and rescue under the forest canopy using multiple UAVs,” *The International Journal of Robotics Research*, Vol. 39, No. 10-11, 2020, pp. 1201–1221. <https://doi.org/10.1177/0278364920929398>, URL <https://doi.org/10.1177/0278364920929398>.
- [10] Chandarana, M., Luo, W., Lewis, M., Sycara, K., and Scherer, S., “Decentralized Method for Sub-Swarm Deployment and Rejoining,” *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 1209–1214. <https://doi.org/10.1109/SMC.2018.00212>.
- [11] Cooper, J. R., “Optimal Multi-Agent Search and Rescue Using Potential Field Theory,” *AIAA Scitech 2020 Forum*, 2020. <https://doi.org/10.2514/6.2020-0879>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2020-0879>.
- [12] Neilan, J. H., Eddy, J., Tran, L., Kelley, B., McQuarry, A. K., Vaughan, M., Williams, R., and Allen, B. D., “Field Testing Visual Odometry: Results from Benchtop to Flight for Autonomous Science Mission Needs,” *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3989.