Environmental Science: Atmospheres

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Environ. Sci.: Atmos.*, 2024, **4**, 214

# The mobility virtual environment (MoVE): an open source framework for gathering and visualizing atmospheric observations using multiple vehicle-based sensors

Marc D. Compere, *[a] Kevin A. Adkins, [b] Avinash Muthu Krishnan, [a] Ronny Schroeder [c] and Curtis N. James [c]

Uncrewed Aircraft Systems (UAS) are becoming prevalent in a wide variety of meteorological investigations. UAS fill an important atmospheric observational gap, namely observations between ground-based sensors and higher altitudes where manned aircraft can safely operate. This paper explores the hardware and software design used for a multi-vehicle atmospheric data collection campaign. The Mobility Virtual Environment (MoVE) is a software framework designed specifically to collect data from multiple vehicles and present a coherent, summary view of a complex scenario. Using both a 2D map and a live updating table, multiple vehicles can be monitored simultaneously to make real-time decisions and quickly assess the mission's effectiveness. MoVE is the software framework used to gather live telemetry inputs before, during, and after flight. MoVE is also the set of tools used to post-process multiple data logs from days of flight experiments into 3D and 4D visualizations over the surrounding terrain. The results are visualizations of otherwise invisible quantities like T, P, RH, and especially vector wind velocities, $\vec{V}_{\text{wind}}$, captured during flight with drone-based sensors. The open-source software and procedures described here can help the atmospheric research, and broader scientific community, achieve greater understanding when using drone-based sensors.

Received 16th August 2022
Accepted 20th December 2023

DOI: 10.1039/d2ea00106c

rsc.li/esatmospheres

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

### Environmental significance

This paper provides experimental methods and results from Uncrewed Aircraft (UA), or drones, sampling the atmosphere over different spatial and temporal scales suitable to the scientific objectives. Atmospheric sensors include temperature, pressure, relative humidity, and sonic anemometers for wind speed, plus GPS for time and location. The paper's focus is software design and methods for gathering multiple time-series data records from different computers on-board individual mobile aircraft. The challenge to aggregating multi-vehicle data includes experiment planning, execution, and post-processing. The solution uses on-board computers to collect data using a common timestamp and a common coordinate frame. This allows scientists, engineers, and pilots to observe otherwise invisible phenomena like temperature profiles or wind speed distribution over complex terrain.

## 1 Introduction

Atmospheric phenomena are composed of features and gradients at a variety of spatial scales. A consensus of atmospheric scientists call these scales, from largest to smallest: global, synoptic, mesoscale, and microscale.[1] Consequently, investigations of atmospheric phenomena often need analysis at a variety of spatial scales. There are multiple considerations needed to achieve the proper scale for a given atmospheric investigation.

First, the right choice of aircraft is critical. Instrumented fixed-wing UAS offer an opportunity to sample multi-kilometer horizontal and vertical distances in a continuous manner with high spatial resolution. Instrumented multirotor UAS possess the ability to fly at slow airspeeds, hover, accomplish vertical sampling profiles, and probe obstacle laden environments, all while making spatially dense observations. Using both fixed-wing and multirotor aircraft simultaneously enables high resolution spatial and temporal sampling of the atmosphere not previously possible. This combination also increases operational and data collection complexity.

Second, when point observations are undertaken, then discrete but concurrent observations are taken at appropriate spatial intervals. Fixed point observations are typically easily

[a]*Department of Mechanical Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA. E-mail: comperem@erau.edu*
[b]*Department of Aeronautical Science, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA*
[c]*Department of Applied Aviation Sciences, Embry-Riddle Aeronautical University, Prescott, AZ, USA*


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

brought together using time with no need to consider a changing geographical position. However, when mobile observations are made, especially concurrent mobile observations, it can be challenging to merge these unique observational datasets together. This task typically requires great effort dealing with datasets after the observations are gathered and the team has returned from the field. This is called post-processing and is non-trivial. An additional task that exacerbates this effort is portraying the resulting merged dataset in an effective manner that lends itself to ready interpretation. Multiple data sources with different formats, different sample rates, and different elevation datum, or time references are common.

### 1.1 The data collection challenge of outdoor field campaigns

The data collection challenge for a field campaign involves scientific, engineering, logistical, personnel, electronics, software, and regulatory considerations. The multi-vehicle field campaign data collection task is to gather incoming data from multiple computer processes, possibly running on separate vehicles, and aggregate these messages with a common timestamp in a common coordinate frame. Because field campaigns can be expensive and timely, it is important to ensure all necessary data is captured in the field, while conditions are correct.

### 1.2 The mobility virtual environment

The Mobility Virtual Environment (MoVE) is designed to aid in all parts of a field campaign: planning, execution, and post processing. In the planning stage, the software can use simulated vehicles to rehearse a test scenario prior to being in the field. A 2D or 3D representation with vehicle icons moving across satellite imagery provides realistic placement and route visualization. Simulations allow test engineers to better design the experiment and convey the plan to pilots and other scientists. This visualization also assists with routing and predicting separation anomalies such as vehicle sequencing and separation for a multi-vehicle scenario. Flight planning prior to arrival at the test site is helpful as a start, but in the field, sometimes conditions change and often it is not simple to understand multiple vehicle trajectories and timings to ensure safe vehicle separation. Multiple flight strategies can be evaluated prior to field execution. For example, simulated scenarios can include lawnmower style flight plans[2] for large area coverage, vertical profiles with multirotor aircraft or helical profiles with fixed-wing aircraft, and horizontal transects. Any single uncrewed aircraft (UA) can have its motion and observational route planned to meet the atmospheric sampling needs, address platform battery and performance limitations, deconflict multiple UA, and meet UA piloting and FAA regulatory requirements. Once this planning is complete, the observational strategy must be conveyed to multiple UA pilots, an Air Boss, Safety Officer, and Visual Observers (VO) for the real flight experiment.

### 1.3 Example multi-vehicle scenario

An example MoVE scenario with real vehicles and pilots is illustrated in Fig. 1. Each vehicle has GPS and onboard sensors

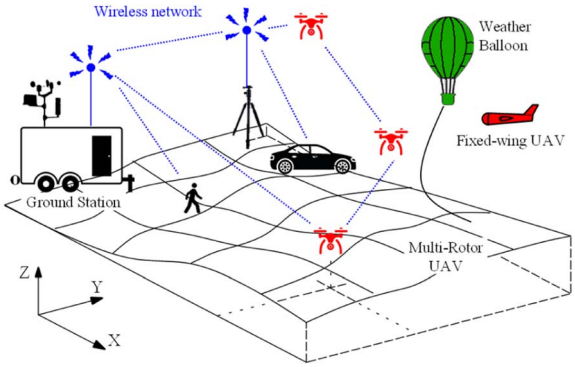


Fig. 1 Example MoVE experiment with moving aircraft (red), ground vehicles (black), pedestrians (black), and a weather balloon (green) sending telemetry data over a wireless network (blue) to a PC located in a nearby ground station.

to measure parameters such as temperature, pressure, humidity, and wind speed. A range of other lightweight, low power sensors, such as particulate matter, methane ($CH_4$), or carbon dioxide ($CO_2$) sensors can also be integrated.

During flight execution with real pilots and real vehicles, MoVE gathers telemetry data from each vehicle to display each vehicle's location and sensor data in real time on the same updating map with icons showing each vehicle's location. A live updating data table shows sensor data in real time from each vehicle. The onboard data collection process writes all flight and sensor data to a comma-separated variable (csv) file with higher frequency than the telemetry link will allow. This means the onboard file is the best data record of the entire flight, but the telemetry updates provide certainty on the sensor values during the experiment execution. MoVE's telemetry receiver also writes a csv file with all vehicles' information logged with a common timestamp and coordinate frame. This improves speed with which sensor data becomes usable by creating a plot of all vehicle positions on a 2D or 3D map while the experiment is conducted, or immediately afterward. Comparing sensor data from each vehicle is straightforward because a Real-Time Clock (RTC) on board ensures all vehicles use the same time reference to timestamp every datapoint. Each vehicle logs data locally, but sometimes field experiments have unexpected failures. The MoVE telemetry offers a secondary record of this data if the onboard logged file is lost or corrupted.

This paper focuses on atmospheric sensing, but MoVE has already been used with ground, air, and pedestrian scenarios documented in.[3–5]

### 1.4 It takes scientists, engineers, and pilots

Effectively exploring the atmosphere with uncrewed aircraft using specialized sensors has both high science potential and high complexity. Data engineers providing live sensor telemetry data from airborne vehicles to atmospheric scientists is key and enables atmospheric scientists to provide guidance to pilots to fly toward areas of interest. Fig. 2 illustrates how pilots, engineers, and scientists can collaborate to more effectively investigate important, but typically invisible, phenomena occurring

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

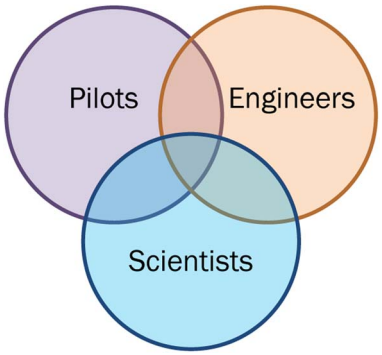


Fig. 2 Collecting atmospheric data using drones takes collaboration among pilots, scientists, and engineers.

in the sky. This teaming allows scientists to readily identify features in real time, such as moisture plumes, updrafts or downdrafts, temperature inversions, flow channelling, or other atmospheric phenomena.

### 1.5 Paper organization

The paper is organized as follows. In Section 2 we provide a literature review of meteorological field campaigns that have utilized multiple mobile sensing platforms and a literature review of multi-vehicle simulation environments. In Section 3 we describe the MoVE design, architecture, and features. Section 4 describes an ideal field campaign with multiple instrumented drones. Section 5 presents the experiment execution and Section 6 presents post-processed results from the MoVE system's data collection. Section 7 summarizes lessons learned and Section 8 contains conclusions and future work.

## 2 Literature review

This section summarizes literature in two categories. First, in Section 2.1 presents relevant literature describing vehicle types for measuring different atmospheric phenomena. Then, Section 2.2 presents literature related to multi-vehicle sensing.

### 2.1 Suitability of vehicle types for atmospheric phenomena

Field campaigns typically generate observations distributed across a geographical area, and several atmospheric investigations exist in the literature that utilize concurrent mobile observations, sometimes with a variety of sensors. Urban areas often possess the infrastructure required for surface-based vehicles to make observations at sufficient spatial resolution. Consequently, both automobiles[6] and bicycles[7–10] and other surface vehicles[11] have been used to investigate spatial temperature variations across urban areas. Instrumented automobiles have also been used to examine the spatio-temporal variability of moisture in the urban canopy layer[12] (UCL) and the spatial variability of atmospheric state parameters in complex terrain.[13] The zero emission nature of various mobile sensing platforms has also been exploited to obtain a variety of atmospheric measurements. Examples include general weather observations using a wind sled[14] and air quality measurements using bicycles.[15] Analogously, aerial measurements have been obtained by hot air balloons to study atmospheric boundary layer (ABL) winds[16] and cloud properties have been explored *via* cable cars.[17]

Uncrewed aircraft systems (UAS) have become increasingly employed in atmospheric investigations in recent years. With much of the atmosphere not accessible to *in situ* measurements, UAS afford a high-potential new option for filling the observational gap between the reach of surface-based sensors and the altitudes that conventional instrumented aircraft can safely operate. UAS also provide an observational platform that is reusable, operationally flexible, and requires minimal supporting infrastructure. Fixed-wing UAS can cover extensive horizontal and vertical distances and multirotor UA possess the ability to launch and recover in confined areas, hover, accomplish non-skewed vertical profiles and survey obstacle laden environments. Swarms of meteorologically instrumented UA provide an opportunity to further capitalize on these advantages. Concurrent UAS operations, with a wide variety of sensors, have been used for intercomparison of lower atmospheric observations.[18] Similarly, an investigation of the lower troposphere has taken place with multiple UAS, in conjunction with traditional meteorological balloons and radar.[19] A combination of well-established observational methods and a variety of UAS has also been used to gain additional insight into the stable ABL.[20] This portion of the atmosphere has also been monitored for trace gases using simultaneous UAS flight operations.[21] Vertically stacked UA have made concurrent observations below, in and above clouds.[22,23] Convective initiation has also been investigated using vertically stacked UA[24] and coordinated multirotor UA undertaking vertical profiles while fixed-wing UA transected the preconvective boundary layer between these vertical profiles.[25] The evolving thermodynamic and kinematic state of the ABL over complex terrain has also been investigated by an assortment of UA operating at two different sites simultaneously.[26–28] The General Urban area Microclimate Predictions (GUMP) tool forecasts urban flow and has been validated using simultaneous observations from multiple meteorologically instrumented UA.[3,29]

Other mobile platforms include ship-based measurements,[30] crewed (manned) aircraft measurements,[31] or dropping radiosondes or launching radiosondes,[32] tethered balloon systems,[33] or pedestrian measurements.[34] All these observation systems need to geo-tag and timestamp each atmospheric sample and aggregate the results into a coherent picture. Each of these vehicles and contexts can benefit from MoVE's data aggregation and real time telemetry for experiment monitoring. The limiting factor for ship-based or crewed aircraft scenarios is network communication. In both scenarios, the vehicle platforms likely already have satellite communications for internet connection globally. In the case of weather balloons or radiosondes, custom telemetry systems using 400 MHz, 1676 MHz, or similar frequencies provide telemetry ranging in 10's-of-km to 300 km, which is likely adequate for updating time and geo-tagged data to a MoVE listener nearby. These satellite or radiosonde


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

communication links have not been incorporated into MoVE but existing Xbee, Lora, ADS-B, 802.11 Wi-Fi, and cellular networks serve as good examples for adding new networks.

### 2.2 Other types of multi-vehicle sensing

Recently, data collection campaigns involving multiple vehicles and over larger areas are becoming more common. With the rise in popularity of autonomous systems, MoVE can aid with communication and data collection between multiple sources. MoVE is applicable to distributed sensing like an intricate system of underwater sensors designed to collect oceanic data and transmit it up to the surface[35] or a wireless sensor network for data collection with UA.[36] By displaying real time data updates, scientists monitoring the data can react quickly to anomalies.

The applicability and societal impact of drones on the atmospheric science and weather community has been clearly documented.[37] Frazier and collaborators present a compelling case and associated list of challenges including multi-vehicle and sensor coordination, communication and data update rates that vary widely among platforms, network link budgets, and even location. This paper directly addresses some of the issues they describe, including spatial sampling scales relevant to atmospheric phenomena in vertical and horizontal dimensions and the challenge of real time data collection and communication in the sensor network.

The geoPebble system is a ground-based wireless sensor network for collecting ice sheet data where a drone is used to hover over each ground-mounted sensor, connect wirelessly, and gather data.[38] The GlacierHawk uncrewed aircraft that travels between geoPebble sensors is described in more detail in ref. 39. Similarities exist between MoVE and this multi-sensor system with a traveling vehicle that collects data from each sensor, but the primary difference is that geoPebble and GlacierHawk is a drone-based approach to ground-based sensors whereas the work presented here is about multiple mobile aerial sensors.

Other methods of vehicle tracking exist, like video image processing systems (VIPS) that produce predictive models based on image tracking and processing.[40,41] VIPS combines multi-vehicle traffic data with cameras and newer VIPS also track vehicles using a two degree-of-freedom mobility model to offer real-time updates. VIPS requires significant computational load and may be difficult to implement internationally. MoVE represents a different, simpler vehicle tracking approach that can address a large area and be used internationally.

Conventional single-vehicle simulators for ground vehicles, like carSIM,[42] AdamsCar[43] or SimVehicle[44] have potential for multi-vehicle interfaces but were not designed for multi-vehicle interactions or vehicle-to-vehicle communications. Other simulators like Microsoft FlightSim[45] focus on a single high-fidelity vehicle model for realistic motion, and include multiple other traffic vehicles. Other simulators like Sumo,[46] or Vissim[47] have many low fidelity vehicle models interacting with each other and traffic lights in a simulated road network. More modern simulation environments like the open-source AirSim are capable of simulating multiple vehicles in a detailed 3D environment.[48] Based on the Unreal Engine 4, AirSim has photorealistic graphics to depict simulated aerial and ground vehicles.[49]

However, none of these simulators can accommodate the combination of needs that MoVE addresses. MoVE fills a unique gap in the simulation landscape with a direct sim-to-real pathway, the ability to model vehicle-to-vehicle communications, and ability to gather live telemetry data from multiple real, aerial or ground vehicles for scientific data collection.

## 3 MoVE software architecture

MoVE is written in Python 3 and is available as an installable package on the PyPi repository.[50] MoVE has a GPLv3 Open Source License, and source code is publicly available at GitLab.[51] Individual MoVE vehicle processes gather and log data locally, on-board the sensing vehicle. If range allows, these processes also report periodic updates over a telemetry network to a central aggregating process for mapping.

The MoVE framework is composed of a Core process that communicates with the vehicle processes, a logging system for scenario recording and playback, a 2D mapping script based on Bokeh, an open source 2D plotting library, and a dashboard. The MoVE Core process runs on a base-station listening for network updates from each MoVE vehicle process. This is illustrated in Fig. 3 below. MoVE Core also stores live streaming telemetry from all vehicles in a single location with a common timestamp and common coordinate frame. The result is a single comma-separated value (csv) file on the MoVE Core computer containing low-bandwidth telemetry data, and the higher frequency data in timestamped logfiles on each of the $N$ separate vehicles. Typically, the logfile format remains fixed, which greatly reduces post-processing time and complexity because the post-processing script also remains the same. Typically, 2D plots of all vehicles are available (a) during the live flights on the dashboard maps and (b) immediately after flights are finished, the 3D plots can be post-processed, right in the field.

### 3.1 MoVE Core

MoVE Core is the central aggregating and control process for all data messages from $N$ vehicles. It manages all the incoming

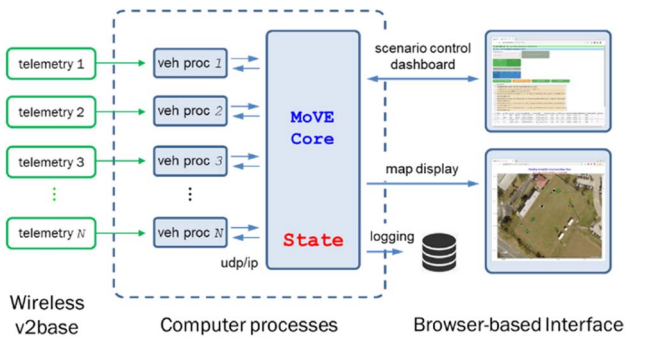


Fig. 3 MoVE architecture with $N$ real vehicles sending telemetry to $N$ vehicle model processes. MoVE Core receives updates from them and user commands before sending state updates to a 2D map, a live updating data table, and logger for post-processing and playback.

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

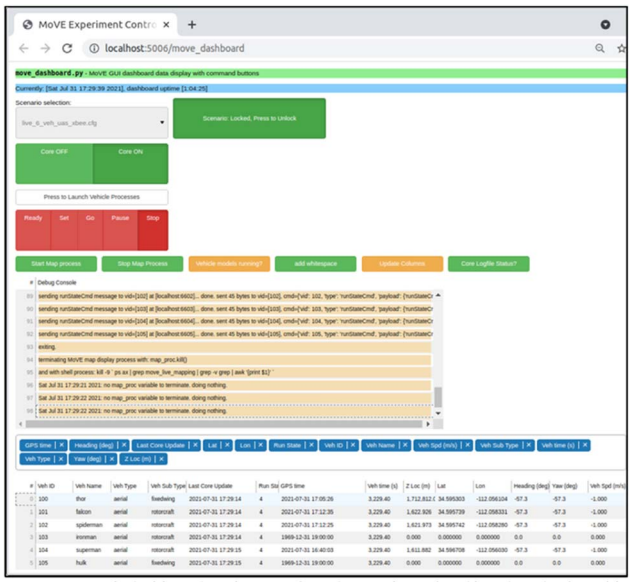

Fig. 4 MoVE's dashboard is a browser-based control panel and live data display table. What is displayed on the bottom indicates 6 vehicle processes with 4 receiving telemetry updates from real vehicle senders.

data flows from *N* vehicles, whether in simulation or from live telemetry. It also populates a live updating data table, and launches a 2d map-based display showing location of each vehicle on a terrain map. Individual MoVE vehicle processes can be one of two types: (a) a live-GPS-follower module or (b) a simulated mobility model with behavior scheduler. The Core sends RunState messages with Ready, Set, Go, Pause or Stop commands to all vehicle processes. Both types of vehicle processes report similar data messages to Core, so both populate the dashboard and live map display with similar data. The live-GPS-follower modules receive GPS latitude and longitude values from real GPS devices. For these vehicle processes, motion in the virtual environment is a result of motion in the real world. Simulated vehicle mobility models return global XY positions by numerical integration using a 4th order Runge–Kutta Ordinary Differential Equation (ODE) solver on transformed body-fixed velocities of simulated vehicles. A priority-based behavior scheduler similar to Rodney Brook's subsumption architecture provides commands to the vehicle mobility model from various behaviors.[52] MoVE Core simultaneously aggregates data from both real and simulated vehicles represented on the dashboard and live map displays.

### 3.2 MoVE dashboard with data table

MoVE dashboard is operated in a web browser, so it is cross-platform (Fig. 4 and 5). A pull-down menu selects the experiment's configuration file and buttons launch vehicle processes. Other buttons issue RunState commands.

These buttons execute command line programs that launch MoVE Core and MoVE vehicle processes. Configuration files capture detailed MoVE parameters, like total number of vehicles, *N*, the ODE solver stepsize for numerical simulation,

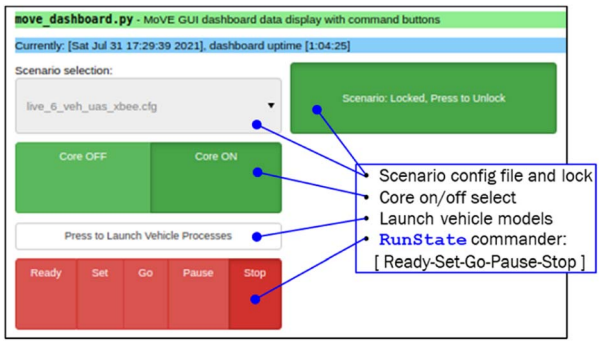


Fig. 5 Dashboard with scenario config file selection and control buttons for a multi-vehicle scenario.

output logging frequency, network port assignments, routes and gates for path following, and mission command sequences. For simulated vehicles, behaviors are specified, each with priority levels to ensure clarity when 2 or more behaviors are active. With real vehicle experiments the configuration file also defines vehicle names, and sets the latitude and longitude origin for UTM conversion.

### 3.3 Executing an experiment

A typical experiment consists of starting the MoVE dashboard and map, selecting a configuration file, then starting MoVE Core and launching *N* vehicle processes (Fig. 5–7). This is done on the MoVE Core computer, typically a PC or laptop. For simulated experiments, launching vehicle processes and stepping through the Ready, Set, Go, Stop RunStates is quite straightforward with the web-based interface. Watching the simulation's 2D map output clarifies what each vehicle is doing and what behavior is active in each vehicle. For real vehicle experiments with telemetry, each vehicle's on-board data logging system must be started, including the telemetry. For aircraft, starting telemetry is a part of the pilot's pre-flight checklist. Next, the MoVE Core computer must be sequenced through RunState commands Ready, then Set, then Go while the experiment is active, then Stop afterward. MoVE Core logs all telemetry from all vehicles, whether simulated or real. For real experiments the higher frequency logs with valuable scientific data are all located on each vehicle platform. This must be manually retrieved and post-processed.

Hardware setup in field experiments typically takes time and the Ready state is intended as a low-energy, yet responsive way

| Veh Name | GPS time | Veh time (s) | Z Loc (m) | Lat | Lon |
|---|---|---|---|---|---|
| thor | 2021-07-31 17:05:26 | 3,229.40 | 1,712,812.( | 34.595303 | -112.056104 |
| falcon | 2021-07-31 17:12:35 | 3,229.40 | 1,622.926 | 34.595739 | -112.058331 |
| spiderman | 2021-07-31 17:12:25 | 3,229.40 | 1,621.973 | 34.595742 | -112.058280 |
| ironman | 1969-12-31 19:00:00 | 3,229.40 | 0.000 | 0.000000 | 0.000000 |
| superman | 2021-07-31 16:40:03 | 3,229.40 | 1,611.882 | 34.596708 | -112.056030 |
| hulk | 1969-12-31 19:00:00 | 3,229.40 | 0.000 | 0.000000 | 0.000000 |

Fig. 6 Close-up of data table showing 6 vehicles with only 4 reporting properly.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

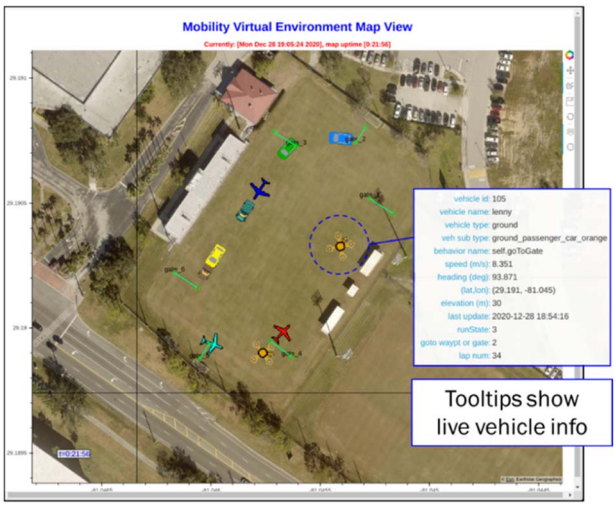


Fig. 7 MoVE 2D map display with icons of ground vehicles, fixed-wing aircraft, multirotor aircraft, and pedestrians. Fixed-wing and multirotor are different subtypes of the same aerial type.

to verify network connectivity from each hardware platform. The Set RunState command is primarily used for simulation-mode and establishes initial conditions for all vehicles in the scenario. The Go RunState command starts the csv logging functionality and is the primary intended RunState during a hardware experiment. Pause is also most useful for simulating scenarios and Stop ends all vehicle processes and Core logging. The data dashboard also has a simple output console in the web browser that provides user feedback on computer process outputs and aids in software development. The data table updates dynamically with vehicle process metrics like simulation time, name, vehicle ID, and real sensor values streaming in from the vehicle platforms.

The data table displayed in Fig. 6 allows researchers to monitor sensor data and quickly detect anomalies or interesting phenomena. The real-time updates are extremely helpful for detecting sensor errors and noting special circumstances during each test. Thus MoVE provides a more effective and efficient atmospheric field experiment.

The dynamic data table offers another menu to select which of the incoming variables to display. The basic telemetry options include time, GPS data (latitude, longitude, and altitude), run state, vehicle behavior, type, sub-type, and most recent vehicle-to-Core update time. Specific sensor values can also be toggled on and off depending on the vehicle's sensor data telemetry. Fig. 5 shows the drop-down menu with a configuration file selected (and locked) along with MoVE RunState command buttons. The vehicle launch button is also shown that commands all $N$ vehicle processes to start, in the background, and begin communicating with Core and any external telemetry configured. These features are also illustrated in Fig. 5.

The telemetry table and update times from each vehicle provides clarity when monitoring multiple vehicles. Typically, all vehicle GPS times are within a few seconds but occasional telemetry packet drops can influence this.

### 3.4 2D map display

The 2D Map Display is launched in a separate browser window (Fig. 7) using the start map process button. The map uses latitude and longitude coordinates to depict all real and virtual vehicles with their respective icons. There are currently three vehicle types: aerial, ground, and pedestrian. Each type has its own subtype, each with their own individual icon. The aerial type has subtypes, multirotor and fixed-wing. The fixed-wing subtype uses a small airplane icon on the map, the multirotor uses a multirotor drone icon, and the pedestrian uses an icon of a person. These icons scale with the map which allows the map to be zoomed in or out to encompass a large experiment area while still clearly displaying each vehicle icon.

Hovering over a vehicle icon will bring up a tooltip with current, dynamically updating information such as vehicle ID, name, type, GPS location, elevation, most recent GPS time, current behavior, and streaming sensor data. This information helps provide a quick 3D understanding of the scenario and identify any vehicles not reporting correct sensor data. The display also provides a check at-a-glance to monitor all vehicle health states. If any icons stop updating on the map, hovering over it can help troubleshoot the error. MoVE live display is only a 2D, top-down view but 3D plots are obtained during post-processing as discussed in Section 5.

### 3.5 Telemetry options with real vehicles and pedestrians

There are multiple approaches for achieving live streaming telemetry into MoVE. Desktop computers, mobile devices with cellular network connections, and even edge computing devices, like Raspberry-Pi class computers, can run MoVE and participate in creative, networked, multi-vehicle simulation and testing. The low computational overhead allows intricate, high vehicle count scenarios. For example, real vehicles in the National Airspace System (NAS) can be monitored in real time with virtual uncrewed aircraft (UA) inserted to test safety or logistical concerns with crewed and uncrewed aircraft. With the ability to represent live vehicles together with simulated vehicles in the same virtual environment, MoVE offers the open-source research community the opportunity to test various investigation scenarios.

User datagram protocol, or UDP/IP communication, is used for inter-process communications between vehicle processes and MoVE Core; however, for receiving vehicle telemetry data, a wireless communications link is needed from the real vehicles to transmit live GPS latitude, longitude, elevation, and sensor updates. Several communication options are possible and the discussion below explains each, along with the pros and cons of each. Design goals when selecting a communication strategy include little or no Federal Communications Commission (FCC) regulation, safety and legality using the FCC Industrial, Scientific, and Medical (ISM) bands, range, network cost, and availability of devices that are also lightweight, compact, and require low power. ISM is the group of frequencies defined by the FCC that can be used without a specialized FCC licence which has the benefit of lower barrier-to-use, but also the occasional risk of interference. In this use case, telemetry

Table 1 Wireless telemetry options

| Wireless technology | Range (m) | Specific example |
|---|---|---|
| Cellular connection | <10 km | Android device: HyperIMU, or iOS device: SensorLog |
| Cellular connection | <10 km | Custom real time data downlink device (RTDD) |
| 802.11 Wi-Fi | 100–500 m | Raspberry Pi or ESP32 built-in Wi-Fi; over 100 m only with hi-gain antenna |
| 802.15.4 | <1 km | Zigbee using Xbee mesh network at 2.4 GHz or 900 MHz |

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

interference is unlikely in an open field where UA fly but, if it did occur, would limit telemetry data. However, this would not be safety critical. Connection types include point-to-point, one-to-many, and mesh wireless topologies, a local wireless Wifi network with nearby router, or the cellular network. Table 1 lists the options tested for wireless telemetry.

### 3.6 Mobile phone app with cellular network

The simplest approach, and one that nearly anyone can setup using a smart device, is to use the cellular network. By downloading either an Android or iOS app, the mobile cellular device can transmit location to an internet-connected MoVE vehicle process. This approach has, perhaps, the world's most widely used and maintained network, but this approach is limited to the sensors built into the cellular device. The Android HyperIMU app[53] and iOS SensorLog[54] applications found on the Google Play Store and Apple App Store, respectively, provide good location transmission over the cellular network. These applications are simple to set up and are quite effective at sending latitude, longitude, elevation, current time, and any sensors onboard the wireless device. The limitation of these apps is an inability to also include custom sensor data to the internet data connection through the app. Only the sensors on the phone can be transmitted.

### 3.7 Microcontrollers with wireless cellular network

Another option tested successfully was a custom device with a microcontroller transmitting custom data packets over the cellular network. The Arduino MKR GSM 1400 is compact, low weight, low power and provides a cellular data connection.[55] This is ideal because it uses the same high-availability cellular network and allows custom sensors and data to be transmitted. This is the preferred option for ground-vehicles and tethered aircraft. But for 'mobile aerial' communications, the United States FCC prohibits any data transmission from a cellular network while in flight. This was likely developed with commercial transport aircraft in mind but, as of 2023, is still the law and applies to UA in flight also.

The Real-Time Data Downlink Device (RTDD) was designed as an improvement to the use of mobile phones with MoVE and allows custom, external sensor data telemetry over the cellular network (Fig. 8). The device is comprised of an Arduino MKR GSM 1400 microcontroller, with an antenna attachment, a 3.7 V LiPo battery and a working SIM card with a data plan. Shown in Fig. 8, it weighs around 170 grams in total and has about 8 hours of battery life. Unlike a mobile phone, the RTDD device, on its own, is not capable of pulling accurate GPS data to send over the network; so, it acts primarily as a sender of data. This in turn does not restrict the RTDD device to any set of sensors. The device can be connected to other microcontroller devices using any one of its three communication protocol options: UART (Serial), I2C and SPI.

With the global investment in cellular networks, the RTDD provides a viable mechanism for delivering live, custom telemetry streams to the MoVE architecture in multi-vehicle scenarios. In the U.S. the RTDD is most useful for pedestrian or ground vehicles but prohibited from "mobile aerial" use.[56] Some international regulations may allow use with aerial vehicles.

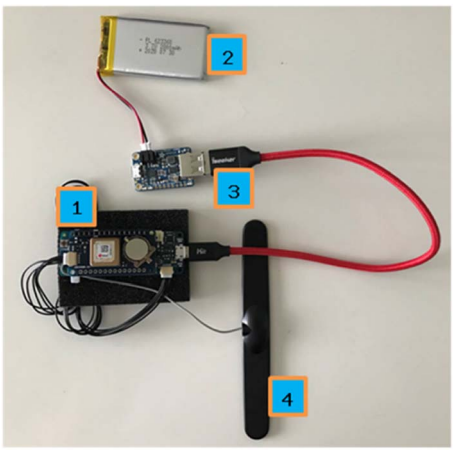


Fig. 8 Real time downlink device with (1) Arduino MKR GSM-1400, (2) LiPo battery, (3) Adafruit PowerBoost 1000, and (4) cellular antenna.

### 3.8 Local 802.11 with Wi-Fi routers outdoors

Next, an 802.11 Wi-Fi network is simple and easy to deploy, has very high bandwidth, can be used with aerial vehicles, but has limited range. For multi-vehicle scenarios conducted within less than approximately 100 m from the base station router and with few or no obstructions, a Wi-Fi network can function quite well. A tethered balloon experiment was quite successful with a 100 m Wi-Fi range. Also communications with vehicle platforms in the laboratory during indoor development have nearly always been through 802.11 Wi-Fi networks. All PCs have Secure Shell Protocol (ssh) clients to access the Raspberry Pi over Wi-Fi; therefore, using this connection outdoors only requires a power supply to operate a mobile Wi-Fi router. High-gain 802.11 antennas can be used for greater range, but the 2.4 GHz or 5 GHz spectrum and 802.11 protocol are not ideally suited for long-range communications, especially when the vehicle is performing aggressive maneuvers with trees or other infrastructure nearby. Achieving 500 m or even 1 km is possible with


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

directional long range antennas on both sides, but these are best suited for stationary vehicles like tethered balloons or multi-rotor vehicles performing vertical profiles with localized translational motion.

### 3.9 Zigbee 802.15.4 mesh network

The configuration tested that meets the FCC requirements and meets size, weight, power (SWaP), and cost requirements is the 802.15.4 Zigbee multipoint network.[57,58] The Xbee brand of microcontrollers implements the Zigbee 802.15.4 standard with a software API, good documentation, code examples, and testing software for monitoring network activity (Fig. 4). Section 3 discussed this more. The 2.4 GHz Xbee 3 Pro devices claim a range of 2 miles line-of-sight (LOS) but our tests indicated something closer to 500 m to 1 km with ideal LOS conditions. Future work includes migrating to sub-GHz communications (*e.g.* 900 MHz) or Lora protocol to achieve similar power, cost, weight, and FCC compliance in the ISM bands with improved range.

### 3.10 Regulatory considerations

In the US, the Federal Communications Commission (FCC) publishes a frequency table[56] with both international and US-based regulations specifying approved use at each frequency range. The Instrumentation, Scientific and Medical (ISM) bands are license-free which means they are both the simplest to use but also the most prone to interference by nearby Radio Frequency (RF) emitters. For telemetry purposes, sub-GHz ISM frequencies (*e.g.* 433 MHz or 900 MHz) provide greater range (kilometer or 10's of kilometers) which will provide excellent utility when flying with fixed-wing vehicles that can reach 1 or 2 km or more. Higher frequencies like 2.4 GHz or 5 GHz provide greater bandwidth but with less maximum range. The telemetry link budget is below 1 kbps data transfer; so, the preference is 2.4 GHz or 900 MHz ISM bands for greater range and license-free operation within the USA's FCC jurisdiction. The FCC explicitly restricts some ISM bands in the US to applications that are not aerial and mobile at the same time. International restrictions are similar but not identical, so the initial choice in this research was to use 2.4 GHz Xbee Pro 3 modules. This works well for multirotor platforms performing vertical profiles, but the 2.4 GHz devices did not provide adequate range for the fixed-wing aircraft, even in ideal RF conditions with direct line-of-sight.

No matter which telemetry link is used, MoVE vehicle processes receive individual vehicle updates from real vehicles directly into the corresponding vehicle process. The vehicle process then updates MoVE Core with the vehicle's position in space as it moves. This is illustrated on the left side of Fig. 3 (telemetry labels) and resulting data is seen in hover-over tooltips in Fig. 7.

## 4 A field campaign using MoVE with multiple instrumented drones

The Southwest United States has a monsoon season that is exceedingly challenging to forecast. The monsoon causes risk and real damage to people, property and livestock. In the summer of 2021, a group of Embry–Riddle researchers investigated specific locations within the greater Colorado Plateau geophysical region.[59] This group included pilots, atmospheric scientists, and engineers who met to study topographical influences on convective processes during the monsoon season. Use of UAS provided concurrent horizontal and vertical observations never before possible. Current Federal Aviation Administration (FAA) regulations constrain any given flight to within visual line of sight (VLOS) of the remote pilot. However, it was possible to meet FCC and FAA regulations while using multivehicle operations to increase the geographical area observed. This was accomplished with a series of concurrent but distinct adjacent flight operations where each vehicle remained within VLOS of the remote pilot. Consequently, multivehicle operations were used to increase the spatial extent and density of observations within the topography of interest. The net result were multiple days of time series data sets from multiple different computers. Generally, this results in a significant challenge in aggregating multiple datasets from distinct operations into a single cohesive picture of the atmosphere. This data aggregation task is what MoVE was designed to address and is described below.

MoVE was used in this field campaign to bring together separate data logs, over multiple days, from multiple vehicles and multiple sensors. The campaign involved six vehicles

Table 2 Instrumentation suite on field campaign vehicles.

| Sensor | $f_{sample}$ | Sensed parameters |
|---|---|---|
| **Devices on both multirotor and fixed-wing aircraft** | | |
| HC2 standard meteo probe | 1 Hz | • Temperature (°C)<br>• Relative humidity (%RH) |
| Pixhawk v4 mini | 10 Hz | • GPS<br>Latitude (dec deg)<br>Longitude (dec deg)<br>Altitude (m in MSL)<br>• Inertial measurement unit<br>Roll, pitch, yaw (deg)<br>$V_{roll}$, $V_{pitch}$, $V_{yaw}$ (deg s$^{-1}$)<br>$V_x$, $V_y$, $V_z$ (m s$^{-1}$)<br>• Barometric pressure sensor<br>Pressure (Pa) |
| **Sensors on multirotor UA only** | | |
| FT-205 sonic wind anemometer | 10 Hz | • 2D wind measurements<br>Wind magnitude (m s$^{-1}$)<br>Wind direction (deg) |
| **Sensors on fixed-wing UA only** | | |
| Trisonica mini | 10 Hz | • 3D wind measurements<br>Wind magnitude (m s$^{-1}$)<br>Wind direction (deg)<br>Wind $u$, $v$ and $w$ vectors |
| Multi-hole pressure probe (MHPP) | 250 Hz | • 3D wind measurements<br>Wind magnitude (m s$^{-1}$)<br>Wind direction (deg)<br>Wind $u$, v and $w$ vectors |

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Fig. 9 Four instrumented multirotors used in the campaign.

simultaneously collecting data in the air. Three instrumented multirotor UA flew vertical profiles at distinct locations, an instrumented fixed-wing UA covered a larger area executing vertically stacked lawnmower patterns, an instrumented manned aircraft also executed the same flight plan as the fixed-wing UA but over a higher and larger area, and weather balloons periodically launched in the middle of the flight operations area. The multirotors and fixed-wing UA were instrumented with sensors listed in Table 2, and are shown in Fig. 9–11 below.

A careful observer will notice 4 instrumented quad rotors in Fig. 9 with only three quad rotor vehicles shown in most figures in Section 6. This is because 1 vehicle experienced an unscheduled and uncontrolled landing during day 1 of the flight tests. This left only 3 instrumented multirotor vehicles available for all flight tests across multiple days.

### 4.1 UAS platforms and sensor overview

Basic components of the multirotor UAS platform are shown in Fig. 12. This includes the baseline UAS flight platform with battery, motors, and flight controller. The primary data acquisition computer is a Linux-based, single-board computer (Raspberry Pi 3b). The computer has multiple USB ports convenient for modular software and hardware development. Nearly all sensors can accommodate UART-based serial communications, or a Tx/Rx adapter can be used to provide a UART-based interface. Linux and UART serial interfaces are straightforward and reliable while computer temperatures do not exceed the computer's recommended operating temperatures. Custom command-line software was written in Python 3 to collect and log data from all sensors, at different rates, to a single csv file during flight.

The Raspberry Pi's built-in Wi-Fi and filesystem allow convenient access to troubleshoot any sensor during pre-flight checks. The sensor logging software runs when the Raspberry Pi is turned on and writes a high-frequency. csv log file with time-stamped data from each sensor. A logfile backup is created every minute using a cron job on each vehicle's Raspberry Pi computer during flight to reduce the likelihood of file corruption from sudden power loss, battery failure, or sd card failure. Also, the Xbee-based 802.15.4 wireless mesh network sends lower frequency updates wirelessly to a ground-based monitoring station displaying all vehicle's sensor data in real time.

### 4.2 Detailed sensor summary

Table 2 summarizes all sensors, reporting frequencies and sensed parameters. The first two rows present devices on both multirotor and fixed-wing aircraft, the middle portion shows sensors on the multirotor aircraft only and the bottom portion shows sensors only on the fixed-wing aircraft.

The multirotors and fixed-wing both had the same HC2 Meteo Probe and Pixhawk collecting the same data on each of them. The vehicle's instrumentation differs in the anemometer sensing vector wind speed. The multirotors had a FT-205 sonic anemometer, while the fixed-wing had two interchangeable anemometers, the Trisonica Mini and a Multi-Hole Pressure Probe (MHPP). Various locations across Arizona were chosen for multiple flights throughout the campaign. At each location, the fixed-wing was flown with both anemometers at least once. The MHPP's 250 Hz sampling frequency is much higher than any other sensors in the suite. This approach is excellent for turbulence statistics but the sample rate forced the Python logging script to log all sensors at 250 Hz. Threads were used for each sensor with threadsafe inter-process communication, so sensor updates at 10 Hz or 1 Hz were recorded accurately despite the csv file being written at 250 Hz. The MHPP sensor was only partially validated at the time of the campaign. To ensure valid data was recorded, the Trisonica Mini was used as a secondary validated wind sensor.

### 4.3 Real-time clock (RTC)

A battery-powered Real Time Clock (RTC) is an important hardware component necessary for any microcontroller without a built-in (RTC). The RTC is the only mechanism for rebooting with the correct time when no internet connection is available. This system used a DS3231 RTC specifically designed to integrate easily with the Raspberry Pi and Linux kernel. In the lab, Linux can synchronize local time when connected to the internet. But in the field, frequently no internet is available. The field-method for accessing the

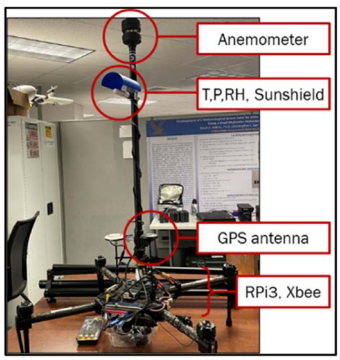


Fig. 10 Instrumented multirotor in lab with sun shield over temperature sensor.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

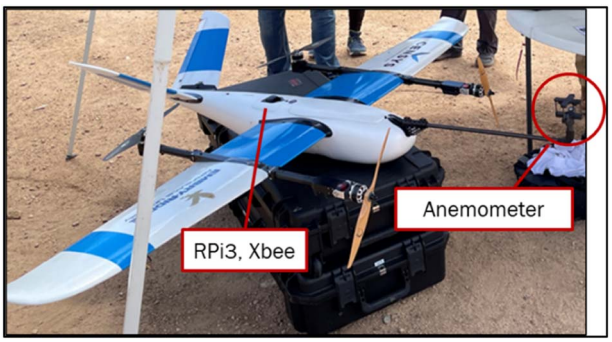


Fig. 11 Fixed-wing UA in pre-flight with Trisonica anemometer attached.

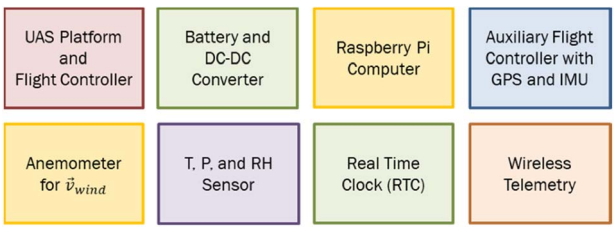


Fig. 12 Vehicle components for UAS-based atmospheric sensing.

computer is a standard 802.11 Wi-Fi connection with a router, but the router typically has no internet connection and serves as a wireless access point only. Each logfile name is created based on the current time and date. So, without the real-time clock, the Python logging script may inadvertently overwrite the previous logfile recorded in the field because the computer reboots to the same time when power cycles. The real time clock is critical for a multi-vehicle data collection system that relies on time to assemble logs from different computers, especially with multiple flights on each UAS data collection computer during a day.

### 4.4 Zigbee mesh network with Digi Xbees

Using the Xbee communication modules from Digi, a comprehensive mesh network can be designed with up to $n$ vehicles sending in data simultaneously to a main data collection system monitoring the process. Just like the RTDD device, the Xbees are not capable of sensing any required data themselves, they just work as data transmitters. The modules can be easily attached to a USB breakout board, allowing them to easily connect to any system. The devices self-form a self-healing mesh network (Fig. 13). The green Xbee is the ground-station coordinator node and others are repeater nodes on individual vehicles.

Xbees can be configured in three different modes: coordinator nodes, repeater nodes and end nodes. Each node type offers a different functionality to a mesh network. Each network needs one main coordinator node that servers as the main point for the entire network. The coordinator has a separate configuration and code from the repeaters. Tracking software versions for each device is important.

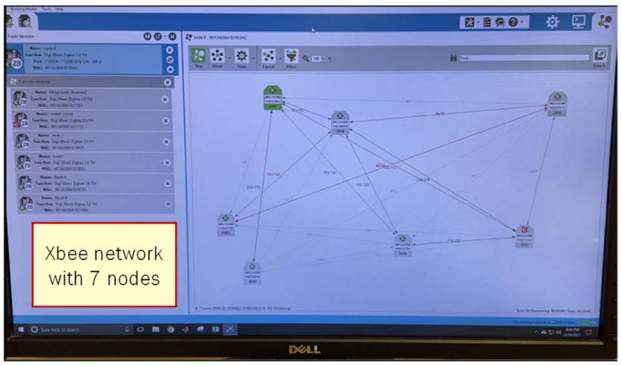


Fig. 13 Xbee mesh network view from Digi's Windows XCTU software showing 7 Xbee nodes. One node is on each of 3 multirotors and 1 fixed-wing aircraft. The others are repeater nodes to extend wireless reach, or the base station node.

Xbees form a mesh network allowing them to be used in remote areas far away from the base station or any other infrastructure. They only require a small input voltage and a network is formed with minimum pre-configuration. Xbees are limited to line-of-sight communication, that can be easily circumvented by strategically adding more Xbees at various locations to extend the range of the overall communication bubble. Additionally, they can send large packet sizes with minimal packet loss.

## 5 Flight tests and experiment execution

A multi-vehicle field campaign for atmospheric sensing is a substantial effort with logistical, personnel, regulatory, piloting, instrumentation, engineering, and science considerations. With science objectives focused on the impact of complex terrain on convective initiation, the complex terrain of the US Southwest during the North American Monsoon season provided an ideal environment. At the commencement of an Intensive Observation Period (IOP) and just prior to flight, at power-ON, each instrumented vehicle established wireless communications with Xbees configured for a self-forming mesh network. The telemetry mesh remained active before, during and after flight to report updates to the base station computer running MoVE Core, dashboard, and live updating map. Also, power-ON initiated the onboard sensor processing scripts, written in Python, that read from serial ports and logged data onboard the aircraft using a Raspberry Pi-3. The UAS pilot's only interaction with the instrumentation system was power-ON or power-OFF and verification with the telemetry engineer. The field operations trailer provided power, large monitors, and protection from the elements (Fig. 14). One wireless repeater station was placed approximately 20 feet above the ground at a terrain ledge to improve communications in the hilly, canyon-like terrain. This height was determined using a Fresnel zone calculator.[60] Fig. 15 shows the antenna stand and wireless repeater strategically located to provide direct radio line-of-sight

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

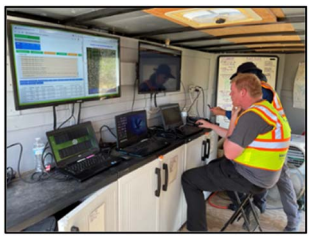

Fig. 14 Multi-vehicle telemetry base station and monitoring displays.

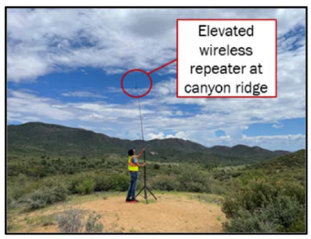


Fig. 15 Multi-vehicle telemetry base station and monitoring.

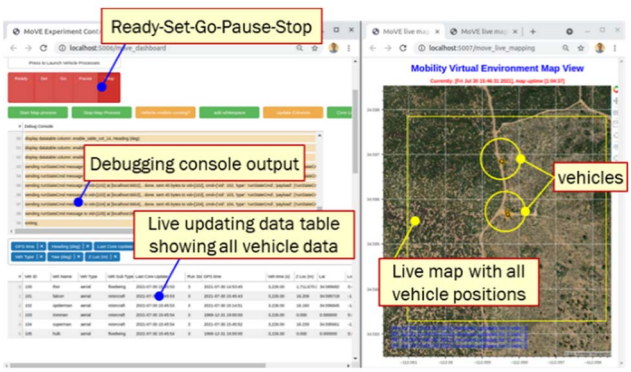


Fig. 16 Multi-vehicle telemetry base station and monitoring displays.

when UAS were flown in the nearby canyon, below the telemetry base station.

Fig. 16 shows a closer view of the large screen monitor providing real time vehicle updates during a multi-vehicle atmospheric sampling experiment. Multi-vehicle flights lasted anywhere from 20 to 40 minutes. One of the most important parameters engineers monitored was log file size on each vehicle. With a properly incrementing log file size, engineers received valuable confirmation that the Raspberry Pi computer, the onboard logging script, and the Xbee telemetry were all working. The second most important parameter monitored was the anemometer wind speed magnitudes. Reasonable wind speeds displayed in the telemetry data provided confidence the sensors were working properly and the flights were acquiring the sought-after data. The live streaming telemetry was critical for detecting errors during pre-flight checks and for monitoring the logging system and sensors while in flight. Fig. 16 illustrates the data table and live updating map updating the parameters of interest.

| # | UAS Pilot Steps | |
|---|---|---|
| 1 | Complete pre-flight checklist | |
| 2 | Enable telemetry and sensors | New |
| 3 | Exit base; Fly mission | |
| 4 | Mission Complete | |
| 5 | Turn off data collection | New |
| 6 | Return to base | |

Fig. 17 Two new steps were incorporated into the UAS pilot checklist to integrate engineering data collection steps.

Future improvements to the display include a limited set of plots for quick visual confirmation the data is being collected correctly. This could be logfile size *vs.* time or T, P, RH, or wind sensor values as a function of time, for each vehicle.

### 5.1 Flight operations modified for engineering telemetry

FAA certificated remote pilots have checklists that helps ensure safety and FAA part 107 compliance. Flying an UA with custom instrumentation streaming live telemetry data to a separate engineering ground station is beyond the training of most UAS pilots. The team of engineers and pilots developed two new checklist entries to ensure the data collection system was operational before takeoff and did not turn off the data logging computer until after the telemetry engineer provided an all-clear to power down. These two new steps are illustrated in Fig. 17.

### 5.2 Multi-vehicle flight operations

The fixed-wing aircraft was a vertical take-off and landing (VTOL) UA, as were the multirotor UA; so, very small launch and recovery areas were needed. Fig. 18 illustrates a vertical take-off

Fig. 18 Instrumented VTOL fixed-wing aircraft launching in airspace with 3 multirotors airborne nearby. The multirotors are not shown in this figure but are flying concurrent vertical profiles illustrated in Fig. 20 and 21.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Fig. 19 Calibrating sensors on multirotor aircraft is simplest with two people: one to rotate the vehicle and the other to monitor the flight control software.

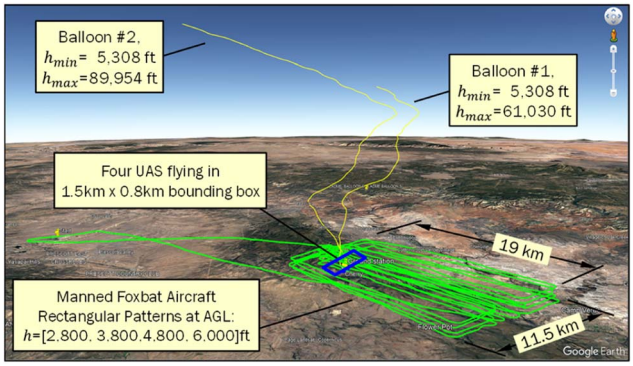


Fig. 20 Wide view in Google Earth with 2 balloon traces (yellow) and multiple rectangular aircraft patterns from a manned Foxbat (green) at different elevations. The four small drones flew concurrently within the small rectangle (blue) near the center.

maneuver from an unimproved road in the Arizona desert. The figure shows the anemometer mounted on a boom in front of the aircraft. A boom length greater than two times the greatest fuselage diameter was used such that the sensor was observing the ambient environment.[61] The fixed-wing UA covered an approximate area of 2 km × 1 km. This provided observations within a given fixed plane. It should be noted that the ground topology changed elevation considerably within the horizontal area, so the height above ground level (AGL) changed accordingly despite flying at a constant altitude Above Take Off (ATO).

Fig. 19 illustrates a multirotor vehicle during sensor calibration prior to launch. Calibrating IMUs immediately prior to flight is a standard part of UAS operations.

## 6 Experimental results

This section describes the data and post-processing methods used to synchronize time and geotagged data histories from separate vehicle computers. Experiments were performed across multiple days and the data presented here is a compilation of 30–31 July 2021 near Cherry, Arizona. The VTOL fixed-wing data shown is from 30 July 2021 and all multirotor flights are from 31 July 2021. Both data sets represent the best flight data across 3 days and approximately 4 flights per day. Reasons for choosing a data set over others includes occasionally poor GPS signal, lost data record for a particular vehicle because of a logging battery failure, or a sensor failure on a vehicle, pilot deviation from the flight plan, or even weather-induced deviations from the flight plan. These experiments were intentionally executed just prior to incoming storms, so timing multiple vehicles, plus a balloon and manned aircraft with a storm front was exceedingly challenging.

The experiment described utilized three multirotor UA and 1 fixed-wing VTOL UA. The science objectives necessitated a weather balloon carrying a reference measurement sensor package with GPS, temperature, pressure, and relative humidity to validate the UAS-based measurements, along with providing an additional vertical sounding of greater vertical extent. In addition, a manned aircraft was flying at higher altitudes above the UA airspace (1000 ft–1500 ft AGL) with an Xbee telemetry node transmitting GPS and time to the ground station, along with other atmospheric measurements. The goal was to gather UA collected data, weather balloon data, and manned aircraft data and present it in a coherent, single picture. GPS locations from all these platforms are shown in Google Earth (Fig. 20).

Google Earth is a free tool that allows multiple GPS traces to be observed in three dimensions and from any camera viewpoint. Google Earth is used for post-processing and not generally intended for real time 3D display. Also, it only shows location and time history of location. Sensor values need additional treatment which is explained in the next section.

The manned Aeroprakt A-22 Foxbat aircraft was too far from the Xbees to reliably receive transmitted signals. On just a few occasions, when the aircraft passed directly overhead, this node was reported on MoVE's dashboard display. Longer range telemetry is a future goal, for example using 900 MHz Lora meshing devices. Similarly, the weather balloon quickly rose to altitudes outside of the Xbee range of communication when released. The only reliable mechanism for incorporating the manned aircraft and weather balloon data was by coordinating the GPS timestamps with the MoVE GPS timestamps during post-processing.

In future field campaigns, a dedicated point-to-point telemetry link specially designed for airborne telemetry can be incorporated into MoVE. These could be similar to Radiosondes with telemetry ranging from 10's-of-km to 300 km using dedicated 400Mhz or 1676 MHz frequencies, or the DragonLink systems which are compact, lightweight, and low power telemetry systems designed specifically for small RC aircraft.[62] Multiple examples in the literature describe these systems for engineering and scientific purposes (*e.g.* ref. 63). This class of dedicated telemetry links use 433 MHz or 915 MHz frequencies for low bandwidth sensor telemetry ranges up to 50 km. The only drawback of radiosonde or DragonLink systems are their point-to-point nature. Individual links would need to be brought into MoVE for each individual vehicle. While this is certainly possible, it increases base-station complexity compared to a meshing system like Lora or Xbee where 1 base-

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

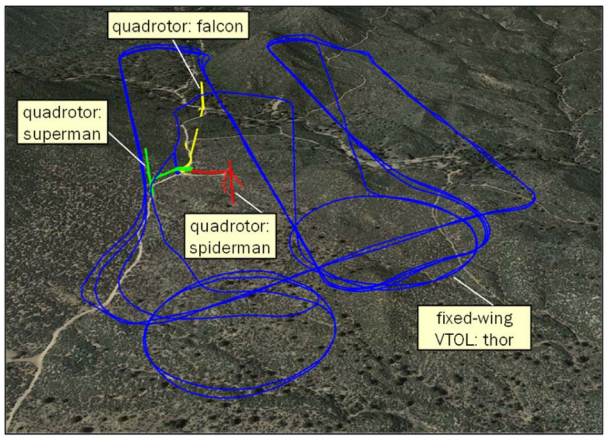


Fig. 21 The blue trace shows the fixed-wing UA flying horizontal transects with loiter circles to coordinate with three multirotors flying vertical profiles. Multirotors shown in yellow, green, and red.

station receiver provides all vehicle updates to MoVE across the mesh network. But, as just mentioned, Xbee and Lora devices lack the required range, so a mix of both telemetry systems could be a good future solution for complex multi-vehicle field campaigns with aircraft both far and nearby.

Fig. 21 and 22 show the Google Earth 3D visualization zoomed-in to the four UAS missions located within a bounding box of approximately 1.5 km by 0.8 km. The manned Aeroprakt A-22 Foxbat aircraft was flying concurrently but farther away making rectangular patterns approximately 19 km by 11.5 km. This means the manned aircraft was only periodically, just barely, within Xbee radio range. The two balloons were launched near the fixed-wing VTOL launch and landing site but the balloon traces are only shown in Fig. 20. The two balloon launch events were isolated events with the balloon trajectory exiting the location near the UAS within minutes, whereas the drone flights lasted for multiple 30 minute flights throughout three test days. The multirotors' vertical profiles are shown with the higher lawnmower pattern of the fixed-wing UA. Vehicle traces are labelled Superman, Spiderman, and Falcon in Fig. 21 and 22. The fixed-wing UA is a VTOL aircraft named Thor, which is shown in blue in Fig. 21 and 22. The blue vertical profile represents this vehicle's launch and recovery locations. These figures were created by importing GPS traces into Google Earth with an interface language called Keyhole Markup Language, or KML. The KML specification is publicly available and convenient for post-processing in Matlab, Octave, ArcGIS, Google Earth, Viking, or other 3D plotting and GPS manipulation tools.

The fixed-wing UAS aircraft covered a larger area horizontally. The multirotor vertical profiles are still visible near the fixed-wing takeoff and landing location as shown in Fig. 22.

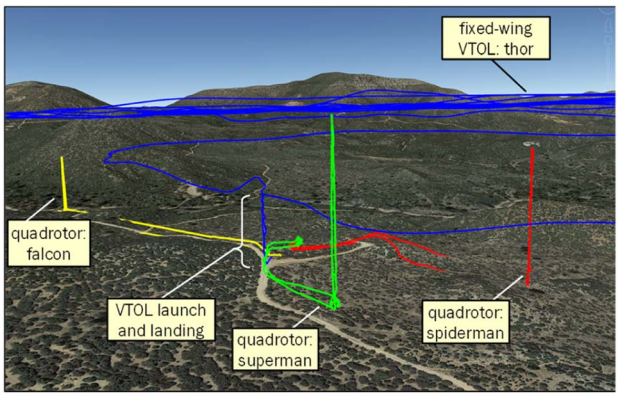


Fig. 22 The yellow, green and red vertical traces represent three multirotors flying vertical profiles, stopping every 50 ft for 60 seconds, sending atmosphere measurements to MoVE for live monitoring.

### 6.1 Fixed-wing and multirotor flight coordination

The fixed-wing VTOL's flight pattern was programmed for horizontal, transverse sections, or transects, with straight-line passes and prescribed turns on either end. Vertical take-off and landings were flown manually and then switched into an automated mode to sequence through a series of pre-defined waypoints. The transects are seen in Fig. 21 and 23 with launch and recovery locations illustrated in Fig. 22. The three multirotors undertook vertical profiles with prescribed 60 seconds hover events at specified, coordinated heights above takeoff. All flights were coordinated by an Air Boss, communicating with all pilots over hand-held radios. Vertical profile observations along the traces, shown in Fig. 21 and 22, are undertaken during ascent to sample undisturbed air. Only observed values taken during ascent are presented.

The fixed-wing UA undertook loiter patterns, seen as large circles, in Fig. 21. These flight plan elements were purposefully implemented to coordinate fixed-wing and multirotor observations. Coordinating the four UA, along with weather balloon launches and the instrumented manned aircraft to accomplish the scientific objectives, while maintaining operational safely and

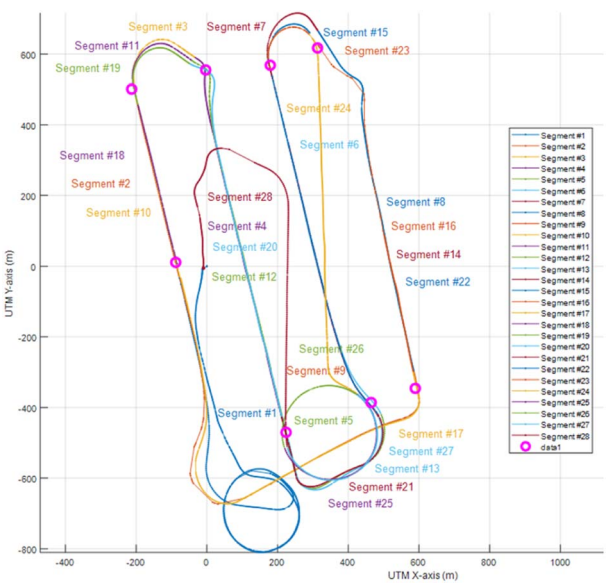


Fig. 23 Fixed-wing flight history with manually placed segmentation points.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

abiding by all FAA regulations, was a substantial challenge. However, MoVE made meeting these challenges significantly easier.

All vehicles' onboard sensors and data logging were enabled at power-ON. This was useful for providing confirmation during the pre-flight check that sensors and telemetry link were operational, but it also meant there were substantial portions in the sensor time history that were not useful for the scientific objectives. The startup, launch, landing, and loiter circles in the fixed-wing time series record were not of primary interest. However, these less-important portions of the data do provide context and verification that the recording system was operational and sampling properly before, during, and after the flight segments of interest.

### 6.2 Fixed-wing horizontal transect post-processing

Data were collected as a comma-separated value file (*i.e.* csv file) onboard each of the four UA using a Raspberry Pi v3. The onboard data collection devices gathered sensor data, logged locally, and also sent telemetry updates to MoVE Core. The onboard software was the MoVE sender from each vehicle to MoVE Core. The post-processing task was to extract, or segment, multiple vehicles' sensor data using a common timestamp and common spatial reference frame. Segmenting the datasets is the process of identifying and extracting portions of the data relevant for the science objectives.

Extracting, or segmenting, data for the fixed-wing UA was primarily accomplished by manual selection of locations in the flight record, described briefly here. Since GPS reports at 1, 5 or 10 Hz, depending on GPS configuration, and the data logging loop in Python was sampling sensors at both 10 Hz and 250 Hz, depending on connected sensors, there were many repeated GPS points in some logs. Using a UTM coordinate transformation, the latitude and longitude time history was converted to XY coordinates in units of meters. This facilitated simple Euclidian norm distance calculations. Repeated GPS points were removed using a distance threshold of 0.01 (m).

For the fixed-wing UA, the horizontal motion provides the simplest segmentation mechanism. Eight magenta circles were manually selected strategically near the ends of straight line segments (Fig. 23). These were referred to as segment demarcation points. The goal was to segment the complete, continuous flight path into more manageable segments for atmospheric scientists to identify and associate with sensor datasets. Using these 8 points, a simple algorithm was developed to identify segment beginning and ending points. Twenty-eight segments were identified within the 3 full laps flown in Fig. 21 and 22 highlighted as 8 magenta circles.

Of the 28 segments, many were turn-arounds or loitering circles. Four segments along the transects have been selected for highlighting. The UA ground speeds along these segments are shown in Fig. 24.

Fig. 25 shows air temperature measurements from the fixed-wing UA while flying the four highlighted segments. True airspeed (anemometer), pressure, and relative humidity were also observed for each of these segments. Selected results for both the fixed-wing and multirotor are shown in the next section.

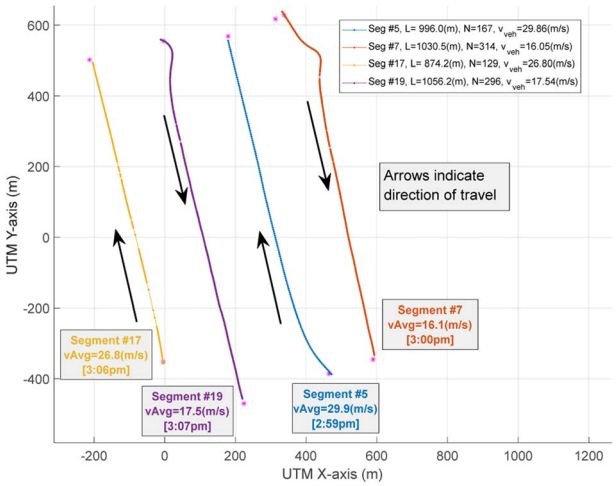


Fig. 24 Fixed-wing segments showing direction, distances, and average ground velocities.

### 6.3 Multirotor vertical profile post-processing

While the VTOL fixed-wing vehicle was loitering, the multirotor UA were commanded to take off and undertook vertical profiles. The Air Boss coordinated all multirotors to simultaneously ascend in 50-foot increments, AGL, and hover for 60 seconds. The weather balloon and fixed-wing aircraft also coordinated the execution of their flight plans with undertaking of these vertical profiles. Between 5 and 8 hover elevations were recorded by each multirotor during any given flight (Fig. 26). All data presented in Fig. 26–29 are all from two days, 30–31 July 2021, near Cherry, Arizona.

Fig. 26 shows altitudes (MSL) from the 3 multirotor UA collecting data simultaneously. Differences in altitude during a data record (flat line segment) are the result of different launch altitudes. Each vehicle's horizontal positioning is shown in Fig. 21, 22 and 27. The specific MSL altitudes were not targets. Rather, they were a result of the Air Boss commanding each multirotor pilot to achieve sequential 50 m increments Above Take Off (ATO). Each pilot's flight controller reports elevation ATO so this is a clear and unambiguous elevation for each pilot to achieve simultaneously. This approach was an intentional part of the science strategy to obtain atmospheric measurements. The objective was to observe T, P, RH, and wind speeds at uniform levels above the ground. The multirotor measurements achieved the objective.

Fig. 27 shows the multirotor anemometer sensed wind speed data during the vertical profiles. Each colored point represents the sensed wind speed over the duration of the 60 s data record, at their respective locations. Careful inspection will reveal Spiderman's highest (last) data point was not reported. The cause is unknown but may be from this vehicle reaching flight battery limits or possibly a miscommunication on the flight's conclusion. Battery life is an important part of electric aircraft flight planning.

Fig. 28 illustrates sensed air temperatures from each of the three multirotors' HC2 Meteo Probes at the 60 s hover locations. Only the hovering sensor data points are displayed. The

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

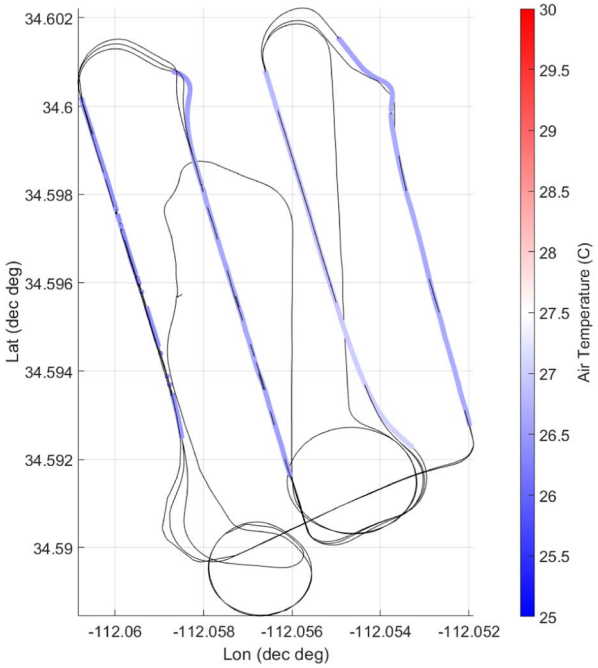


Fig. 25 Matlab plot illustrating fixed-wing air temperature data along the four horizontal segments of interest. The [25–30 C] scale matches multirotor plots.

multirotor aircraft are flown in vertical profiles for the specific purpose of capturing steady state atmospheric conditions at a specific location in a way the fixed-wing aircraft cannot. These sensor values during transitioning between elevations are not presented to clearly illustrate temperatures at discrete heights above take-off. The expected trend is apparent where temperatures near the ground surface are higher than air temperatures at higher elevations. The [25–30° C] temperature scales matches the fixed-wing UA in Fig. 25.

One of the overarching scientific objectives was to look at the influence of topography on convective initiation. Consequently,

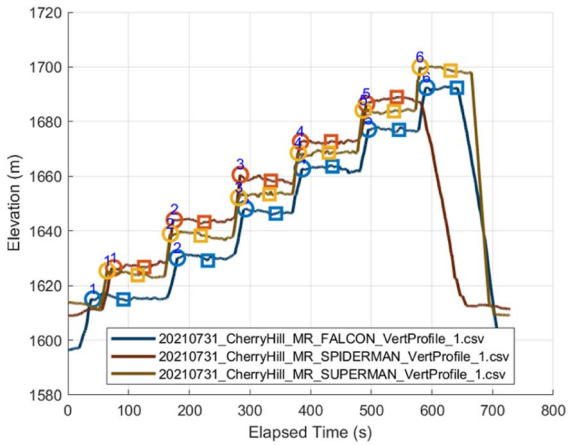


Fig. 26 Elevation (MSL) time histories provide a simple segmentation approach. Sensor data extracted for these segments illustrates each 60 seconds elevation hold.

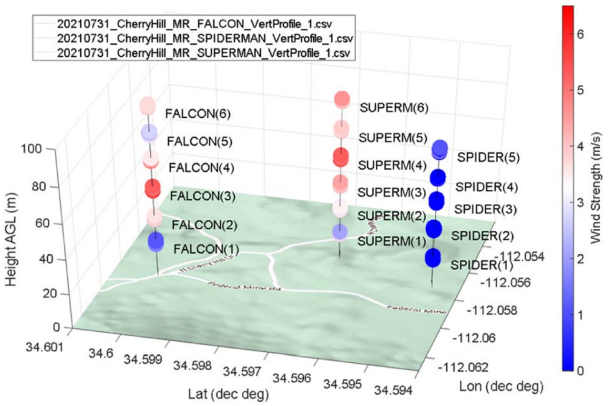


Fig. 27 Anemometer measurements for each hover elevation in the data record from Fig. 26.

the uncrewed aircraft were placed over varying terrain. Some aircraft were launched from valleys and some from higher ground. This experimental design not only created differences in local elevation above mean sea level but also different launch and recovery locations. Vegetation varied and was typically more fertile in the lower lying locations with expected higher evapotranspiration. Topography also caused differences in insolation based on terrain slope and exposure, and differences in diurnal evolution due to overnight cold pooling. Finally, and most predominately, the temperature variation near the surface was expected to be very nonlinear and, oftentimes superadiabatic as a result of high surface heating from the sun and conductive heat transfer to the air immediately above the surface. These phenomena result in large temperature changes at lower altitudes, which is shown in Fig. 28.

Fig. 29 showcases the combined multirotor and fixed-wing UA segments showing the vertical and horizontal coverage possible. Google Earth provides a compelling visualization that adds perspective from the surrounding hillside and valley, where the experiment took place. Matlab's Keyhole Markup Language (KML) exporting functionality provided a straightforward mechanism for using Google Earth to display sensor data

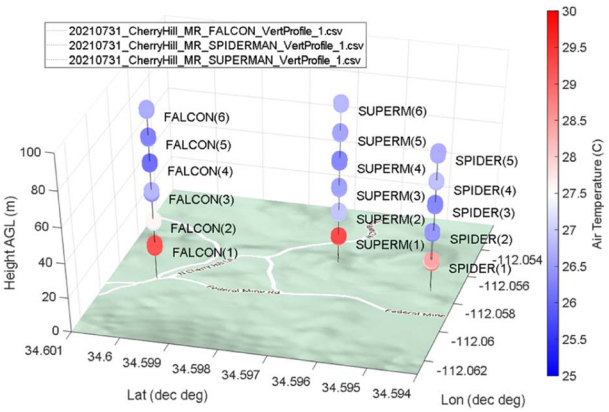


Fig. 28 Air temperature measurements for each hover data record displayed in Fig. 26. The [25–30 C] scale matches the fixed-wing plot.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Fig. 29 Google Earth image showcasing multi-vehicle, concurrent flight tests and resulting temperature data. The long horizontal segments are from a fixed-wing UA and the three vertical profiles are from 3 separate multirotor UA. Datasets shown are over 2 separate days on July 30th and 31st 2021 near Cherry, Arizona.

with compelling 3D terrain visuals. This figure illustrates data over 2 separate days.

## 7 Summary and lessons learned

The field campaign in Arizona consisted of multiple IOPs consisting of multi-vehicle operations over a large geographical area composed of complex terrain. Various aircraft executed unique flight plans within the complex topography. This required thoughtful consideration and planning, detailed pre-mission briefs, and continuous in-flight monitoring to ensure deconfliction and that the scientific objectives of the campaign were being met. The utilization of the MoVE software reduced the burden during each of these stages (planning, rehearsing, briefing, data acquisition, and post-processing) and allowed participants to more effectively focus on the task at hand.

Along with improving situational awareness, MoVE helped gather and ensure the integrity of streamed data with real-time updates. The Raspberry Pi running on each system would boot and run a multi-threaded Python code that would collect sensor data and send it out with wireless network messages. Each sensor's data was collected in a separate thread, and data packets were packed and sent in their own thread, totalling 5 threads in the main program. Using the Raspberry Pi's Linux operating system, each sensor was given a persistent name according to its unique device ID and physical port connection, allowing the program to run on boot. When, rarely, the persistent names would not apply correctly, or the sensor's onboard electronics would not send updates and all the data would read 0, MoVE's telemetry data readily alerted the engineering team and the issue easily fixed with a power cycle. Hence, MoVE was essential in verifying functional sensors and communications channels, or if a power cycle was required before flying. The connections, once established, would consistently return updates provided they had power.

Bringing together a multifaceted instrumentation system poses challenges in logistics, regulations, flight hardware, sensing and communications, data collection, and team procedures. The following lessons-learned are crucial for creating a reliable fielded system:

- Frequent software and hardware subsystem testing is critical for successful field operations. Built-in checks that aid troubleshooting in the field are helpful.
- Tests with real pilots and real vehicles undertaking atmospheric investigations in real time places a significant cognitive load on everyone involved. Reliable, tested subsystems that frequently report health status are critical.
- Real-time clocks (RTC) for reliable Raspberry Pi timestamps are critical for field work without an active internet connection. Most computers rely on the internet for acquiring accurate time, so without internet access in the field, logfiles can be overwritten or have incorrect timestamps which can cause confusion and uncertainty during post-processing.
- Reliable battery power, with or without a DC–DC converter, is critical to ensuring that voltage drops do not occur and cause the Raspberry Pi computer to reboot, resulting in inadvertent loses of data during the reboot.
- Planning a sensor power budget is critical. A low-cost USB digital multimeter is critical for understanding the power draw of each sensor through the USB interface. This approach allowed data collection system testing in the lab while the Raspberry Pi was accessible and helped inform our battery amp-hour ratings to ensure proper battery life.
- Post-processing and verifying all quantities of interest before the field campaign reduces accidental omissions discovered after the field campaign ends. It may seem burdensome to post-process test data but this exercise surfaces the need for logfile changes that are unknown but critical before undertaking the high-value field tests.
- Using a separate flight controller for engineering data collection, such as a Pixhawk4, provides a convenient set of flight data entirely separate from the flight controller. However, this device must be calibrated and armed to ensure it provides good IMU and GPS data.
- Sending current logfile size in each telemetry message helps provide confidence before, during, and after data collection that the flight will yield the desired data. An increasing logfile size, in bytes, shows that the data collection script is working and appending data properly.
- Lastly, to prevent inadvertent data loss from unexpected power-loss, a Linux Cron job was configured to make a backup every minute, locally, on the Raspberry Pi during flight to ensure safe logfile copies existed in more than one location.

The foremost priority for this campaign was capturing atmospheric data from UA in-flight. Certain atmospheric phenomena manifested quickly and existed over short temporal periods. Real-time observations afforded the opportunity to monitor the atmosphere and make more informed flight and science decisions while the weather changed.

## 8 Conclusions and future work

The Mobility Virtual Environment (MoVE) software helps make complex multi-vehicle field campaigns easier and more robust.

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

MoVE helps gather multiple data records from multiple computers using a common timestamp and common coordinate frame. Throughout each phase of a field campaign, MoVE can be utilized to better plan, rehearse, brief, execute, monitor, and post-process results. Tested in a multi-vehicle campaign to collect atmospheric data, the software proved to be integral in the overall success of the data collection efforts. With a modular design, additional network communication capabilities are being added. The software can be used for field experiments around the globe, according to different regulatory environments. Additionally, as an open-source project, MoVE source code is freely available and can be modified to better meet each research team's needs.

MoVE is under active development with improvements in simulation and live-vehicle data display. An ADS-B network will be added to display all crewed (manned) aircraft in the nearby airspace. Also, one specific planned addition is a set of live plots for quick visual inspection that each vehicle's sensor data is being recorded. Logfile size and T, P, RH, and wind sensor values are displayed in tabular form, but a live updating graph would improve monitoring during the experiments. Also, additional long-range telemetry links need to be added such as DragonLink or other links similar to radiosondes to meet long range needs for certain vehicles. Lastly, a vehicle-to-vehicle (V2V) communications model and multi-vehicle mission sequencer will be implemented to improve simulation-based planning. The software is developed in Linux and all libraries are supported by Windows.

## Author contributions

Dr Marc Compere was the originator and developer of the MoVE software. He designed and tested MoVE communications, logging, and post-processing scripts. Dr Adkins was the coordinator and lead scientists for all piloting and atmospheric science. He purchased and instrumented all aircraft and led the piloting logistics to accomplish field data collection. Mr Avinash Muthu Krishnan was a Masters and, subsequently, a PhD student who tirelessly helped test and develop wireless telemetry link, instrumentation, field operations, and data post-processing efforts. Dr Ronny Schroeder flew an Aeroprakt A-22 Foxbat manned aircraft with our GPS and Xbee transmitter during UAS operations. Dr James provided and launched multiple weather balloons to complement the drone data and helped edit the manuscript.

## Conflicts of interest

There are no conflicts of interests to declare.

## Acknowledgements

The authors would like to acknowledge Dr Dan Macchiarella as Co-Lead for the College of Aviation's Domestic Study Away program during Summer 2021. This was a tremendous help that gathered students and equipment for the field campaign. We would also like to thank Dr Ronny Schroeder and Dr Curtis James for Co-Leading the Low Altitude Drone Monsoon Experiment (LADME) effort from the Prescott Campus. We would also like to acknowledge Dr Mark Sinclair and Dr Michael Kaplan for atmospheric science expertise during the Arizona field campaign.

## References

1 D. C. Ahrens, *Meteorology Today: An Introduction to Weather, Climate, and the Environment*, Thomson/Brooks/Cole, Belmont, CA, 2007.
2 W. T. Weldon and J. Hupy, Investigating Methods For Integrating Unmanned Aerial Systems In Search And Rescue Operations, *Drones*, 2020, **4**(3), 38.
3 K. A. Adkins, W. Becker, S. Ayyalasomayajula, S. Lavenstein, K. Vlachou, D. Miller, M. Compere, A. Muthu Krishnan and N. Macchiarella, Hyper-Local Weather Predictions with the Enhanced General Urban Area Microclimate Predictions Tool, *Drones*, 2023, **7**, 428, DOI: **10.3390/drones7070428**.
4 M. Compere, K. Adkins, O. Legon, P. Currier, MoVE: A Mobility Virtual Environment for Testing Multi-Vehicle Scenarios, in *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, INDIA, Novi, MI, 2019, **http://gvsets.ndia-mich.org/publication.php?documentID=721**.
5 M. Compere, G. Holden and O. Legon, MoVE: A Mobility Virtual Environment for Autonomous Vehicle Testing, IMECE2019-10936, *Int. Mech. Eng. Congr. Expo.*, 2019, **59414**, V004T05A097.
6 F. Leconte, J. Bouyer, R. Claverie and M. Pétrissans, Analysis of nocturnal air temperature in districts using mobile measurements and a cooling indicator, *Theor. Appl. Climatol.*, 2017, **130**(1–2), 365–376, DOI: **10.1007/s00704-016-1886-7**.
7 L. W. A. van Hove, G. J. Steeneveld, C. M. Jacobs, B. G. Heusinkveld, J. A. Elbers and A. A. M. Holtslag, Exploring the Urban Heat Island Intensity of Dutch Cities, *Plant J.*, 2011, **34**, 2170.
8 J. J. Cassano, Weather Bike: A Bicycle-Based Weather Station for Observing Local Temperature Variations, *Bull. Am. Meteorol. Soc.*, 2014, **95**(2), 205–209. **https://journals.ametsoc.org/view/journals/bams/95/2/bams-d-13-00044.1.xml**.
9 N. B. Rajkovich and L. Larsen, A Bicycle-Based Field Measurement System for the Study of Thermal Exposure in Cuyahoga County, Ohio, USA, *Int. J. Environ. Res. Public Health*, 2016, **13**(2), 159, DOI: **10.3390/ijerph13020159**.
10 C. Cao, Y. Yang, Y. Lu, N. Schultze, P. Gu, Q. Zhou, J. Xu and X. Lee, Performance Evaluation of a Smart Mobile Air Temperature and Humidity Sensor for Characterizing Intracity Thermal Environment, *J. Atmos. Oceanic Tech.*, 2020, **37**(10), 1891–1905, DOI: **10.1175/JTECH-D-20-0012.1**.
11 A. Middel, S. Alkhaled, F. A. Schneider, B. Hagen and P. Coseo, 50 Grades of Shade, *Bull. Am. Meteorol. Soc.*, 2021, **102**(9), E1805–E1820.
12 H. Mayer, A. Matzarakis and M. G. Iziomon, Spatio-temporal variability of moisture conditions within the urban canopy layer, *Theor. Appl. Climatol.*, 2003, **76**, 165–179.


© 2024 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

13 G. de Boer, S. Waugh, A. Erwin, S. Borenstein, C. Dixon, W. Shanti, A. Houston and B. Argrow, Measurements from mobile surface vehicles during the lower atmospheric profiling studies at elevation – a remotely-piloted aircraft team experiment (LAPSE-RATE), *Earth Syst. Sci. Data.*, 2021, **13**(1), 155–169, DOI: **10.5194/essd-13-155-2021**.

14 S. Gonzalez, M. Bañon, J. V. Albero, R. Larramendi, H. Moreno, F. Vasallo, P. Sanz, A. Quesada and A. Justel, Weather Observations of Remote Polar Areas Using an AWS Onboard a Unique Zero-Emissions Polar Vehicle, *Bull. Am. Meteorol. Soc.*, 2019, **100**(10), 1891–1895. **https://journals.ametsoc.org/view/journals/bams/100/10/bams-d-19-0110.1.xml**.

15 A. Samad and U. Vogt, Mobile Air Quality Measurements using Bicycle to Obtain Spatial Distribution and High Temporal Resolution in and Around the City Center of Stuttgart, *Atmos. Environ.*, 2021, **244**, 117915.

16 E. I. F. de Bruijn, S. de Haan, F. C. Bosveld, B. W. Schreur and A. M. Holtslag, Observing boundary-layer winds from hot-air balloon flights, *Weather Forecast.*, 2016, **31**(5), 1451–1463.

17 A. Beck, J. Henneberger, S. Schöpfer, J. Fugal and U. Lohmann, HoloGondel: In situ cloud observations on a cable car in the swiss alps using a holographic imager, *Atmos. Meas. Tech.*, 2017, **10**(2), 459–476, DOI: **10.5194/amt-10-459-2017**.

18 L. Barbieri, S. Kral, S. Bailey, A. Frazier, J. Jacob, J. Reuder, D. Brus, *et al.*, Intercomparison of Small Unmanned Aircraft System (sUAS) Measurements for Atmospheric Science during the LAPSE-RATE Campaign, *Sensors*, 2019, **19**(9), 2179.

19 H. Luce, K. Lakshmi, H. Hiroyuki, L. Dale, M. Tyler, Y. Masanori and T. Toshitaka, Vertical Structure of the Lower Troposphere Derived from MU Radar, Unmanned Aerial Vehicle, and Balloon Measurements during ShUREX 2015, *Prog. Earth Planet. Sci.*, 2018, **5**(1), 1–16.

20 S. T. Kral, J. Reuder, T. Vihma, I. Suomi, E. O'Connor, R. Kouznetsov, B. Wrenger, A. Rautenberg, G. Urbancic, M. O. Jonassen, L. Båserud, B. Maronga, S. Mayer, T. Lorenz, A. Holtslag, G. Steeneveld, A. Seidl, M. Müller, C. Lindenberg, C. Langohr, H. Voss, J. Bange, M. Hundhausen, P. Hilsheimer and M. Schygulla, Innovative Strategies for Observations in the Arctic Atmospheric Boundary Layer (ISOBAR)—The Hailuoto 2017 Campaign, *Atmosphere*, 2018, **9**(7), 268.

21 T. J. Schuyler, C. B. Sean and I. G. Marcelo, Monitoring Tropospheric Gases with Small Unmanned Aerial Systems (sUAS) during the Second CLOUDMAP Flight Campaign, *Atmosphere*, 2019, **10**(8), 434.

22 G. C. Roberts, M. V. Ramana, C. Corrigan, D. Kim and V. Ramanathan, Simultaneous Observations of Aerosol-Cloud-Albedo Interactions with Three Stacked Unmanned Aerial Vehicles, *Proc. Natl. Acad. Sci. U.S.A.*, 2008, **105**(21), 7370–7375.

23 C. Reymann, R. Alessandro, L. Fayçal, B. Murat and L. Simon, Adaptive Sampling of Cumulus Clouds with UAVs, *Auton. Robots*, 2018, **42**(2), 491–512.

24 S. C. van den Heever, L. D. Grant, S. W. Freeman, P. J. Marinescu, J. Barnum, J. Bukowski, E. Casas, A. J. Drager, B. Fuchs, G. R. Herman, S. M. Hitchcock, P. C. Kennedy, E. R. Nielsen, J. M. Park, K. Rasmussen, M. N. Razin, R. Riesenberg, E. R. Dellaripa, C. J. Slocum, B. A. Toms and A. van den Heever, The Colorado State University Convective CLoud Outflows and UpDrafts Experiment (C3LOUD-Ex), *Bull. Am. Meteorol. Soc.*, 2021, **102**(7), E1283–E1305. url: **https://journals.ametsoc.org/view/journals/bams/102/7/BAMS-D-19-0013.1.xml**.

25 S. E. Koch, M. Fengler, P. B. Chilson, K. L. Elmore, B. Argrow, D. L. Andra and T. Lindley, On the use of Unmanned Aircraft for Sampling Mesoscale Phenomena in the Preconvective Boundary Layer, *J. Atmos. Oceanic Tech.*, 2018, **35**(11), 2265–2288.

26 G. de Boer, J. Jacob, P. B. Chilson, S. W. Smith, B. Argrow, D. Lawrence, J. Elston, D. Brus, O. Kemppinen, P. Klein, J. K. Lundquist, S. Waugh, S. C. C. Bailey, A. Frazier, M. P. Sama, C. Crick, D. Schmale III, J. Pinto, E. A. Pillar-Little, V. Natalie and A. Jensen, Data generated during the 2018 LAPSE-RATE campaign: an introduction and overview, *Earth Syst. Sci. Data.*, 2020, **12**(4), 3357–3366, DOI: **10.5194/essd-12-3357-2020**.

27 A. Jensen, J. O. Pinto, S. C. Bailey, R. A. Sobash, G. de Boer, A. L. Houston, P. B. Chilson, T. Bell, G. Romine, S. W. Smith, D. A. Lawrence, C. Dixon, J. K. Lundquist, J. D. Jacob, J. Elston, S. Waugh and M. Steiner, Assimilation of a Coordinated Fleet of Uncrewed Aircraft System Observations in Complex Terrain: EnKF System Design and Preliminary Assessment, *Mon Weather Rev.*, 2021, **149**(5), 1459–1480.

28 E. A. Pillar-Little, B. R. Greene, F. M. Lappin, T. M. Bell, A. R. Segales, G. B. H. de Azevedo, W. Doyle, S. T. Kanneganti, D. D. Tripp and P. B. Chilson, Observations of the thermodynamic and kinematic state of the atmospheric boundary layer over the San Luis Valley, CO, using the CopterSonde 2 remotely piloted aircraft system in support of the LAPSE-RATE field campaign, *Earth Syst. Sci. Data.*, 2021, **13**(2), 269–280, DOI: **10.5194/essd-13-269-2021**.

29 A. Grushin, A. Tyagi, J. Gluck, S. Mohseni, N. Nigam, M. Klopfenstein and R. S. Lee, *GUMP: General Urban Area Microclimate Predictions Tool*, AIAA AVIATION Forum, 2020.

30 W. Gülzow, *et al.*, One year of continuous measurements constraining methane emissions from the Baltic Sea to the atmosphere using a ship of opportunity, *Biogeosciences*, 2013, **10**(1), 81–99.

31 F. Gallo, *et al.*, Measurement report: aerosol vertical profiles over the western north Atlantic ocean during the North Atlantic Aerosols and Marine Ecosystems Study (NAAMES), *Atmos. Chem. Phys.*, 2023, **23**(2), 1465–1490.

32 Y. Yang, *et al.*, Modulation of wintertime canopy Urban Heat Island (CUHI) intensity in Beijing by synoptic weather pattern in planetary boundary layer, *J. Geophys. Res.: Atmos.*, 2022, **127**(8), e2021JD035988.

Open Access Article. Published on 12 de gener 2024. Downloaded on 22/4/2026 1:19:10.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

33 F. Ramelli, *et al.*, Using a holographic imager on a tethered balloon system for microphysical observations of boundary layer clouds, *Atmos. Meas. Tech.*, 2020, **13**(2), 925–939.
34 S. Kaur, M. J. Nieuwenhuijsen and R. N. Colvile, Pedestrian exposure to air pollution along a major road in Central London, UK, *Atmos. Environ.*, 2005, **39**(38), 7307–7320.
35 M. Caraivan, P. Burlacu and V. Dobref, Further Developments of Multi-Purpose Underwater Data Collection Devices Deployed with Remotely Operated Vehicles, *Sci. Bull. Nav. Acad.*, 2016, **19**(1), 154–159, DOI: **10.21279/1454-864X-16-I1-026**.
36 Z. Huang, C. Lu,and J. Zhong, December. A multi-objective hyper-heuristic for unmanned aerial vehicle data collection in wireless sensor networks, *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 1614–1621.
37 A. E. Frazier, A. J. Mathews, B. L. Hemingway, C. Crick, E. Martin and S. W. Smith, Integrating small unmanned air-craft systems (SUAS) into meteorology and atmospheric science: Challenges and opportunities for GIScience., *GI_Forum*, 2017, **5**, 189–199.
38 S. Anandakrishnan, S. G. Bilén, J. V. Urbina, R. G. Bock, P. G. Burkett and J. P. Portelli, The geoPebble System: Design and Implementation of a Wireless Sensor Network of GPS-Enabled Seismic Sensors for the Study of Glaciers and Ice Sheets, *Geosciences*, 2021, **12**(1), 17.
39 M. Volpe, Design, Development, and Field Testing of the GlacierHawk: an Unmanned Aerial System for Geoscience Data Retrieval, *Master's thesis*, Penn State University, 2019, **https://etda.libraries.psu.edu/catalog/17310mrv12**.
40 K. D. Baker and G. D. Sullivan, Performance assessment of model-based tracking, *Proceedings IEEE Workshop on Applications of Computer Vision*, IEEE Computer Society, 1992, pp. 28–29.
41 D. Koller, K. Daniilidis and H. H. Nagel, Model-based object tracking in monocular image sequences of road traffic scenes, *Int. J. Comput. Vis.*, 1993, **10**(3), 257–281.
42 R. F. Benekohal and J. Treiterer, CARSIM: Car-following model for simulation of traffic in normal and stop-and-go conditions, *Transp. Res. Rec.*, 1988, **1194**, 99–111.
43 MSC Adams, *Adams Car*, MSC Software, 2011.
44 *SimCreator*, Realtime Technologies, Inc., 2023.
45 Flight Simulator, *Microsoft*, 2020, **https://www.flightsimulator.com/**.
46 German Aerospace Center, *SUMO, Simulation of Urban MObility*, 2001.
47 PVT VisSim, *PTV Planung Transport Verkehr AG*, 1992.
48 AirSim, *Microsoft*, **https://microsoft.github.io/AirSim/**, accessed 26 Dec 2023.
49 Epic Games Inc., *Unreal Engine*, **https://www.unrealengine.com/**, accessed 26 Dec 2023.
50 M. Compere, *Mobility Virtual Environment, Python Package Index (PyPI)*, **https://pypi.org/project/mobility-virtual-environment/**, accessed 26 Dec 2023.
51 M. Compere, *Mobility Virtual Environment, MoVE, Software Repository*, **https://gitlab.com/comperem/move**, accessed 26 Dec 2023.
52 R. A. Brooks, Elephants don't play chess, *Robot. Auton. Syst.*, 1990, **6**(1–2), 3–15.
53 *C. Campisi as Ianovir, HyperIMU, Android App in Google Play Store*, 2023, **https://play.google.com/store/apps/details?id=com.ianovir.hyper_imu**, retrieved 26 Dec 2023.
54 B. Thomas, *SensorLog, Log and Stream Sensor Data, iOS App on Apple App Store*, **https://apps.apple.com/us/app/sensorlog/id388014573**, accessed 26 Dec 2023.
55 *Arduino MKR GSM 1400, Arduino.cc*, **https://store-usa.arduino.cc/products/arduino-mkr-gsm-1400**, accessed 25 Dec 2023.
56 *FCC Online Table of Frequency Allocations*, Federal Communications Commission (FCC), 2021, **https://transition.fcc.gov/oet/spectrum/table/fcctable.pdf**.
57 Digi International Inc., *Xbee Pro 3*, **https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee3-zigbee-3**, accessed 30 Jul 2022.
58 Digi International, Inc., *Zigbee Wireless Mesh Networking*, **https://www.digi.com/solutions/by-technology/zigbee-wireless-standard**, accessed 30 Jul 2022.
59 R. Schroeder, *et al.*, Detection of Convective Initiation and Suppression in Northern Arizonas Complex Terrain with Unmanned and Manned Aerial Systems, *AGU Fall Meeting Abstracts*, 2021, vol. 2021, **https://agu.confex.com/agu/fm21/meetingapp.cgi/Paper/822695**.
60 EverythingRF, *Fresnel Zone Calculator*, **https://www.everythingrf.com/rf-calculators/fresnel-zone-calculator**, accessed 30 Jul 2022.
61 W. Gracey, *Measurement of Aircraft Speed and Altitude*, John Wiley & Sons, 1981.
62 *Dragon Link Advanced 433 MHZ and 915 MHZ Telemetry System, DragonLink RC*, P.O. Box 388, Cortaro AZ, 85652, N.D., **http://www.dragonlinkrc.com/instructions/v3equipment/**.
63 E. J. Liu, *et al.*, Dynamics of outgassing and plume transport revealed by proximal unmanned aerial system (UAS) measurements at Volcán Villarrica, Chile, *Geochem. Geophys. Geosystems*, 2019, **20**(2), 730–750.


© 2024 The Author(s). Published by the Royal Society of Chemistry