# Hierarchical Mixture of Experts for Advanced Air Mobility Flight Phase Classification

Evan Kawamura *[1], Thomas Lombaerts †[1], Vahram Stepanyan‡[1], Keerthana Kannan §[1], Chester Dolph ¶[2], and Corey Ippolito ‖[1]

[1]*NASA Ames Research Center, Moffett Field, Mountain View, CA 94035*

[2]*NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666*

**Advanced Air Mobility (AAM) and Urban Air Mobility (UAM) operations will have numerous vehicles and aircraft flying in the airspace, which poses safety and security concerns. Commercial airlines utilize Air Traffic Management (ATM) and Air Traffic Control (ATC) for real-time monitoring, surveillance, traffic coordination, and rerouting to maintain safe and efficient flight patterns. Transferring ATM and ATC architectures to AAM/UAM will be challenging to implement since AAM/UAM aircraft fly at lower altitudes, have more static and dynamic obstacles, operate in highly dense environments, and have several more aircraft to monitor for a given volume of the national airspace (NAS). Aircraft typically have the following flight phases: takeoff, climb, cruise, descent, and landing. Classifying these flight phases provides insight into ensuring safe operations, providing situational awareness of the NAS, and monitoring flights in emergencies. Automatic flight phase classification will enhance the efficiencies of ATM/ATC-like architectures for AAM/UAM, especially since numerous aircraft will be flying in highly dense urban environments. Typical flight phase classification methods are all-or-nothing, which will not capture or accurately classify the transitions between flight phases. Utilizing hierarchical mixture of experts (HME) provides a flight phase classification solution that includes transitions between the flight phases by assigning weights based on ground-based distributed sensor readings from cameras and radar. Adding the transitions between flight phases increases the fidelity of flight phase classification and provides deeper insight into flight phase classification by leveraging distributed sensing concepts. Simulation results and post-processed flight test results demonstrate the utility of HME for automatic and robust flight phase classification for real-time AAM operations.**

## I. Introduction

Advanced Air Mobility (AAM) operations will have numerous vehicles and aircraft in the sky, which poses safety and security risks. Air Traffic Management (ATM) services for AAM will have dense 4D (3D space and time) trajectory-based operations, so airborne cooperative methods for air traffic coordination allow efficient and safe operations without relying only on detect and avoid algorithms [1]. Urban Air Mobility (UAM) operations have several proposed concepts with safety, social, system, and aircraft factors, leading to new infrastructure to support the proposed ideas and operations [2]. Scaling the current Air Traffic Control (ATC) architecture is a crucial constraint for UAM operations with challenges such as numerous operations, highly dense operations, services for low altitude flights, and different methods for interactions and training for pilots, automation, and aircraft than traditional ATC methods [3].

One of the components of AAM ATM services may include flight phase classification to provide situational awareness of the airspace, enhance safe AAM operations, and monitor AAM flights with online assistance in case emergencies arise. The International Civil Aviation Organization (ICAO) and the Commercial Aviation Safety Team (CAST) worked together to create the CAST/ICAO Common Taxonomy Team (CICTT) to develop definitions and

*Computer/GNC Engineer, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

†Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

‡Aerospace Research Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

§Software Engineer, KBR Wyle Services, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA.

¶Aerospace Engineer, Aeronautics Systems Engineering Branch, NASA Langley Research Center, 1 Nasa Dr, Hampton, VA 23666.

‖Aerospace Scientist, Intelligent Systems Division, NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA

taxonomies for aviation. One of CICTT's several developments includes definitions for flight phases, which have different categories: standing, pushback/towing, taxi, takeoff, initial climb, en route, maneuvering, approach, landing, emergency descent, uncontrolled descent, and post-impact [4]. The FAA's Airplane Flying Handbook states that the most critical flight phases are takeoff and landing, and at a minimum, pilots must have checklists for the following phases of flight: takeoff, cruise, descent, and landing [5].

Even though AAM/UAM aircraft do not have the same physics and governing equations as conventional airplanes, there are some overlapping principles in physics and flight phases. Overall, the three general types of AAM/UAM aircraft are 1) quadcopter (vertical propulsion system only), 2) lift plus cruise (LPC, separate vertical and forward propulsion system), and 3) tilt rotor/wing (vectored propulsion system) [6]. There are three general categories for flight phases: A, B, & C such that categories B & C have relevant components for AAM/UAM aircraft through takeoff, climb, cruise, descent, and landing [7]. Thus, this paper defines AAM/UAM flight phases in one of six categories, with hover being a special category for UAS/VTOL types of operations since UAM CONOPS includes UAS [8]:

1) Takeoff (TO)
2) Climb (CL)
3) Cruise (CR)
4) Descent (DE)
5) Landing (LA)
6) Hover (HO)

Figures 1 and 2 show the flight phases and similarities between commercial aircraft and AAM aircraft with the same flight phases: takeoff, climb, cruise, descent, and land.

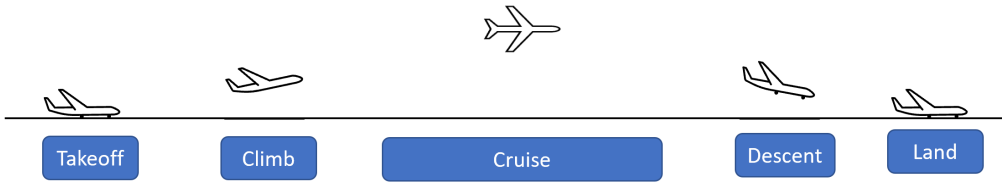



**Fig. 1 Commercial Aircraft Flight Phases**

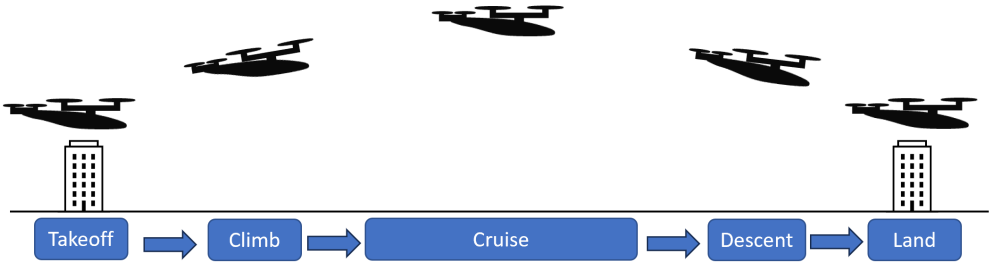



**Fig. 2 AAM Aircraft Flight Phases**

There are efforts in flight phase classification through various techniques and methods. One study utilizes decision trees for flight phase classification, demonstrating high efficiency, short computation times, and easy implementation [9]. Another approach has support vector machine models for tracking pilot's eye movements to determine the phases of flight [10]. Machine learning with ADS-B data can identify flight phases reliably and economically, which provides an improved noise and emission model [11]. Automatic flight phase identification with maximum posterior probabilities based on vertical speed and current barometrical altitude demonstrates accurate flight phase classification for taxiing, takeoff, en-route, and landing with minimum false detection and gaps in datasets [12].

Another method for classification applications is a hierarchical mixture of experts (HME). A literature survey on a mixture of experts shows several types and applications of a mixture of experts, and in particular, a mixture of multilayer perceptron (MLP) experts uses MLP in the experts and gating network, which demonstrates good classification and performance [13]. Applying HME to two Expectation–Maximization (EM) algorithms shows that the HME approach converges faster than back propagation by two orders of magnitude [14]. Utilizing HME for classification

demonstrates faster training than the EM algorithm for Exclusive-Or, N-Input Parity, and Two Spirals problems [15]. For solving the M-step of the EM iterations, training HME for classification tasks reduces the training time compared to the iteratively reweighted least squares method [16]. Another HME classification example includes a three-level structure with posterior probability estimation and training algorithms on various datasets from the ELENA, Pima Indians, and Ionosphere databases [17]. Combining Bayesian techniques with HME provides a framework for solving multi-dimensional regression problems [18].

Other works in HME classification involve practical, real-world applications. Implementing Bayesian analysis with HME demonstrates accurate vowel recognition for speech recognition tasks [19]. A spam filtering HME structure classifies spam emails, which performs well against other machine learning approaches [20]. Classifying thyroid disease with an extended mixture of experts network has a lower miscalculation rate than a typical mixture of experts for binary classification of hierarchical data [21]. Another medical application for HME involves outbreak detection with probabilistic models to fuse different detection methods, demonstrating better performance than simple majority voting algorithms in terms of sensitivity and specificity [22]. Applying an adversarial HME learning model demonstrates robustness against adversarial attacks on artificial and real data [23].

It is also possible to provide navigation solutions with HME. Adaptive Kalman filtering with a mixture of experts framework generates an unsupervised learning approach with adaptive weighting through a gating network, which can perform in real-time applications [24]. Applying HME with adaptive Kalman filters provides an interplanetary navigation solution for Mars Pathfinder [25]. Another example of HME involves state and parameter estimation for unmanned aerial vehicles by considering banks with and without acoustics [26]. A mixture of experts framework blends motor angular velocity models to yield a more accurate motor angular velocity estimation than the individual models [27].

The proposed research in this paper determines the flight phases for AAM/UAM aircraft through HME with heterogeneous sensors that detect AAM/UAM aircraft. Sensor readings of aircraft indicate the potential flight phase of the detected aircraft. For example, sensor readings of an aircraft in the takeoff flight phase should expect changes in elevation and altitude. Overall, flight phase classification with an HME approach that captures the primary flight phases and transitions provides deeper insight into flight phase classification in distributed sensing concepts.

This paper's organization is as follows. Section II describes the details for classifying the different flight phases, and section III provides the details about the proposed HME structure for flight phase classification. Section IV shows flight phase classification results based on a simulated AAM flight and post-processed Unmanned Aerial System (UAS) flight test data. Finally, section V ends with concluding remarks, key takeaways, and future work.

## II. Flight Phase Classification Description

This section describes the definitions for classifying the aforementioned flight phases: takeoff, climb, cruise, descent, land, and hover.

### A. Distributed Sensors: Cameras and Radar

The Distributed Sensing project under Transformational Tools & Technologies utilizes distributed cameras and radar in the environment to enable monitoring, surveillance, tracking, and navigation for AAM aircraft and operations [28]. Thus, the definitions for the flight phases include camera and radar measurements. Cameras provide pixel coordinates of the target, which feed into the computations for the azimuth and elevation angles. Similarly, radar also provides azimuth and elevation angles of the target but also the range. See Ref. [29, 30] for more details about the computations for the azimuth and elevation angles of the target. See Ref. [31, 32] for more details about the sensor models and characteristics.

### B. Flight Phase Classification Definitions

HME flight phase classification requires continuous data collection and looking at data from previous times. Therefore, a forward derivative numerical method to estimate the change in the sensor measurements will not be possible. However, a backward derivative numerical method will be attainable since previous data will be available. The three-point backward derivative has higher accuracy than the typical first-order Euler method by considering three points, the current point and two previous points [33]:

$$f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h} \tag{1}$$

with an upper bound error of $O(h^2)$ and $h$ as the step size. Since radar data tends to be noisy, it would be ideal to smooth out the data to reduce the noise, increasing the accuracy in the flight phase classification. After computing the three-point backward derivative for the data, applying an exponential moving average (EMA) smooths the derivative:

$$f'_{EMA,i} = \alpha f'_{3ptBkwd,i} + (1-\alpha) f'_{EMA,i-1} \tag{2}$$

in which $\alpha$ is the smoothing factor. Increasing $\alpha$ will favor the three-point backward derivative more than the previous EMA derivative, which will be more jagged. Consequently, the flight phase classification will have more switches between flight phases, which is not ideal and challenging to provide insight into airspace monitoring. Most AAM/UAM flights should be smooth with occasional bumps (staircase pattern) during the descent, so the expected derivative computations will weigh EMA more than the three-point backward derivative, i.e., smaller values for $\alpha$.

Table 1 shows the mathematical definitions for classifying the different flight phases based on the camera data with $\psi_c$ to denote the camera's azimuth angle and $\theta_c$ to denote the camera's elevation angle. Let $\Delta\psi_c$ and $\Delta\theta_c$ be the difference in azimuth and elevation angles between two iterations of EMA with the three-point backwards derivative from Eqn. (2).

**Table 1 Camera Flight Phase Classification Definitions**

| Flight Phase | Abbreviation | Definition | Notes |
|---|---|---|---|
| Takeoff | TO | $\Delta\psi_c = 0, \Delta\theta_c > 0$ | Assuming 1D vertical takeoff |
| Climb | CL | $\lvert\Delta\psi_c\rvert > 0, \Delta\theta_c > 0$ | Climb can be left or right (±) so need $\lvert\Delta\theta_c\rvert$ |
| Cruise | CR | $\lvert\Delta\psi_c\rvert > 0, \Delta\theta_c = 0$ | Cruise can be left or right (±) so need $\lvert\Delta\theta_c\rvert$ |
| Descent | DE | $\lvert\Delta\psi_c\rvert > 0, \Delta\theta_c < 0$ | Descent can be left or right (±) so need $\lvert\Delta\theta_c\rvert$ |
| Land | LA | $\Delta\psi_c = 0, \Delta\theta_c < 0$ | Assuming 1D vertical landing |
| Hover | HO | $\Delta\psi_c = 0, \Delta\theta_c = 0$ | No translation or rotation so no change |

Table 2 shows the mathematical definitions for classifying the different flight phases based on the radar data with $\psi_r$ to denote azimuth angle, $\theta_r$ to denote elevation angle, and $r_r$ to denote range. Let $\Delta\psi_r$, $\Delta\theta_r$, and $\Delta R_r$ be the difference in radar azimuth and elevation angles and range between two iterations of EMA with the three-point backwards derivative from Eqn. (2).

**Table 2 Radar Flight Phase Classification Definitions**

| Flight Phase | Abbreviation | Definition | Notes |
|---|---|---|---|
| Takeoff | TO | $\Delta\psi_r = 0, \Delta\theta_r > 0, \Delta R_r \neq 0$ | Assuming 1D vertical takeoff |
| Climb | CL | $\lvert\Delta\psi_r\rvert > 0, \Delta\theta_r > 0, \Delta R_r \neq 0$ | Climb can be left or right (±) so need $\lvert\Delta\theta_r\rvert$ |
| Cruise | CR | $\lvert\Delta\psi_r\rvert > 0, \Delta\theta_r = 0, \Delta R_r \neq 0$ | Cruise can be left or right (±) so need $\lvert\Delta\theta_r\rvert$ |
| Descent | DE | $\lvert\Delta\psi_r\rvert > 0, \Delta\theta_r < 0, \Delta R_r \neq 0$ | Descent can be left or right (±) so need $\lvert\Delta\theta\rvert$ |
| Land | LA | $\Delta\psi_r = 0, \Delta\theta_r < 0, \Delta R_r \neq 0$ | Assuming 1D vertical landing |
| Hover | HO | $\Delta\psi_r = 0, \Delta\theta_r = 0, \Delta R_r = 0$ | No translation or rotation so no change |

Table 3 shows the mathematical definitions for classifying the different flight phases based on the telemetry position data with $U$ as Up to denote altitude and $d_{EN}$ to denote the magnitude of the 2D position: East-North. Let $\Delta h$ be the difference in altitude and $\Delta d_{EN}$ between two iterations. The telemetry data is very smooth and lacks excessive sensor noises. Consequently, three-point backwards derivatives and exponential moving average are not needed to smooth out the telemetry data before computing the gating network weights.

## III. Proposed Hierarchical Mixture of Experts Description

The proposed solution utilizes the softmax function for flight phase classification to avoid completely neglecting irrelevant flight phases:

$$g_i = \frac{\exp(u_i)}{\sum_{i=1}^{4} \exp(u_i)} \tag{3}$$

**Table 3 Telemetry Flight Phase Classification Definitions**

| Flight Phase | Abbreviation | Definition | Notes |
|---|---|---|---|
| Takeoff | TO | $\Delta U > 0, \Delta d_{EN} = 0$ | Assuming 1D vertical takeoff |
| Climb | CL | $\Delta U > 0, \lvert\Delta d_{EN}\rvert > 0$ | Climb can be left or right (±) so need $\lvert\Delta d_{EN}\rvert$ |
| Cruise | CR | $\Delta U = 0, \lvert\Delta d_{EN}\rvert > 0$ | Cruise can be left or right (±) so need $\lvert\Delta d_{EN}\rvert$ |
| Descent | DE | $\Delta U < 0, \lvert\Delta d_{EN}\rvert > 0$ | Descent can be left or right (±) so need $\lvert\Delta d_{EN}\rvert$ |
| Land | LA | $\Delta U < 0, \Delta d_{EN} = 0$ | Assuming 1D vertical landing |
| Hover | HO | $\Delta U = 0, \Delta d_{EN} = 0$ | No translation or rotation so no change |

such that $g_i$ are the weights and $u_i$ are the intermediate gates [26]. The intermediate gates, $u_i$, for flight classification act as binary operations. When the target is in a particular flight phase based on the definitions provided in Tables 1 & 2, $u_i = 1$ for the associated flight phase. Conversely, when the target is not in a particular flight phase, $u_i = 0$. For example, if the target is in the takeoff phase at iteration $i$, $u_{i,TO} = 1, u_{i,CL} = 0, u_{i,CR} = 0, u_{i,DE} = 0, u_{i,LA} = 0, \& u_{i,HO} = 0$.

Figure 3 shows the high level of the HME architecture diagram for flight phase classification. There is no sensor fusion between the camera and radar gating network weights for ground nodes with only one type of sensor. Conversely, ground nodes with camera and radar will merge the camera and radar gating network weights to create a blended sensor flight phase classification. See the simulation results in section IV to see both types of phenomena.

Generally, the gating network with the softmax function yields weights such that it always includes all possible outcomes and heavily weighs the most preferred, favorable, or accurate outcome while summing to one. For example, at iteration $i$, the gating network weights are 35% takeoff, 13% climb, 13% cruise, 13% descent, 13% land, and 13% hover. Conversely, implementing a binary or all-or-nothing gating network would eliminate inactive or irrelevant possibilities such that their weights would be zeros.

Figure 4 shows the flight phase classification diagram for the main flight phases during typical flight operations and the flight phase classification criteria based on the flight phase definitions from Tables 1 & 2. For gating network applications that favor smaller values, inserting a minus sign in front of the intermediate gates weighs smaller values more than larger ones. An application for selecting smaller values would be weighing smaller RMS errors more than the larger RMS errors for a navigation solution [26]. Here are some important comments regarding sensors and flight phase classification:

1) All distributed sensors are the same frame, i.e., sensor frame or inertial frame.
2) Since the FAA's UAM CONOPS includes UAS, HME includes hover as a flight phase [8].
3) The climb and descent flight phases have 2D motion with blended flight phases. Climb has elements of takeoff and cruise, while descent has elements of cruise and land.

## IV. Results

This section describes the HME flight phase classification results for two different scenarios. The first scenario involves a simulated AAM flight from Oakland to San Francisco with seven cameras and two radar to view the aircraft throughout its trajectory. The second scenario has a UAS flight test at NASA Ames Research Center with four distributed cameras to view the UAS throughout most of its trajectory.

### A. Simulation Results

This subsection uses the simulated AAM aircraft flight and telemetry data from Middle Harbor Shoreline Park (MHP) in Oakland to Fifth & Mission Garage (FMG) as shown in Ref. [34]. The simulated data provides a synthetic dataset for testing the HME flight phase classification architecture. Figure 5 shows the flight path as a blue line with the yellow dot as the origin and the red dots as the sensor locations. It takes off at MHP, flies across the bay near the Oakland Bay Bridge, flies over or between several buildings in northeastern San Francisco, such as Google San Francisco, The Gap Headquarters, Salesforce Tower, LinkedIn, San Francisco Museum of Modern Art, Hyatt Regency San Francisco, & Marriot Marquis San Francisco, and then finally lands at FMG. Table 4 shows the sensor locations, altitude, heading, and elevation.

Figures 6a and 6b show the distributed sensor coverage of the simulated flight path [32]. The black cones show each

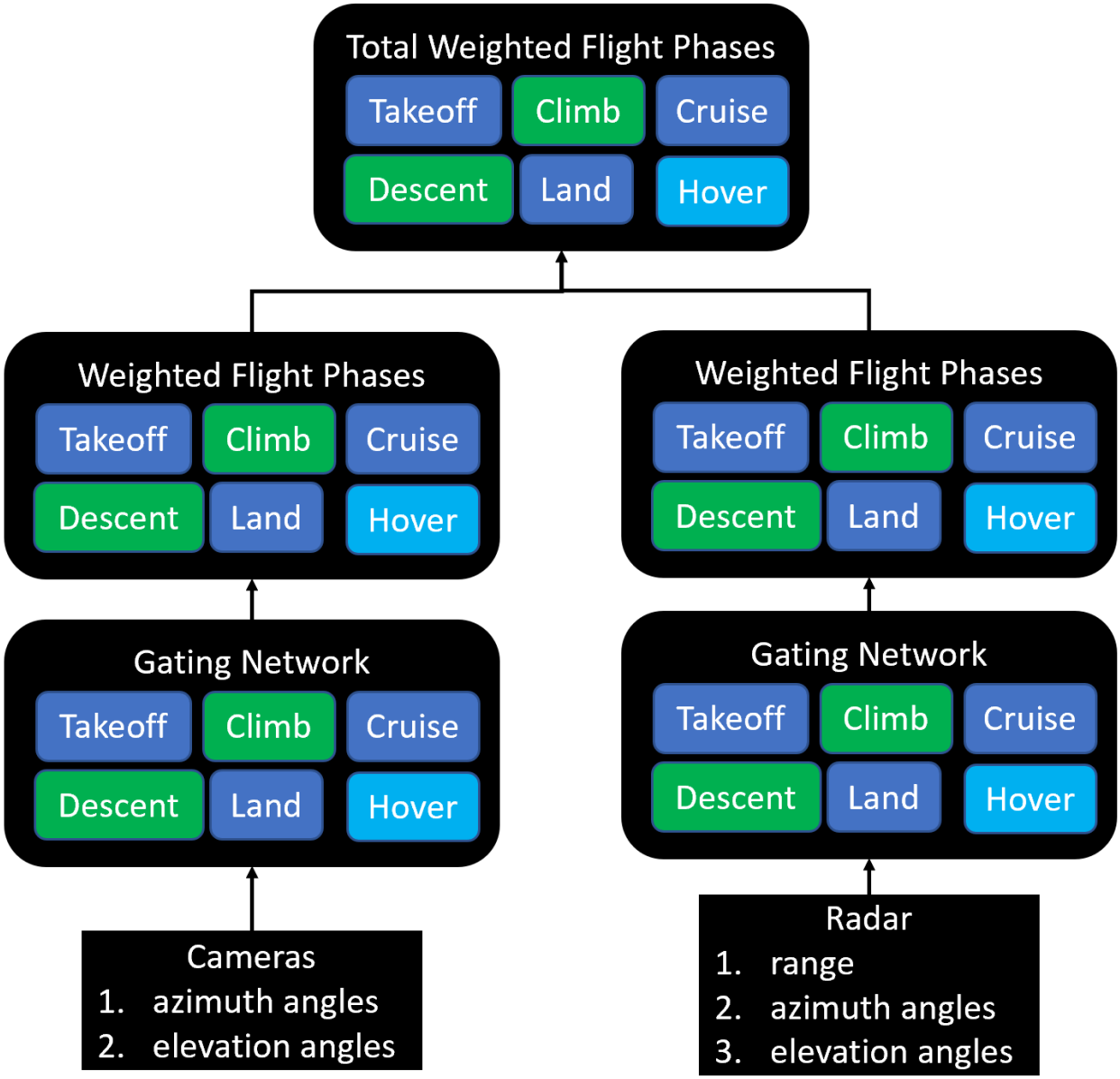



**Fig. 3 HME Flight Phase Classification Architecture Diagram**

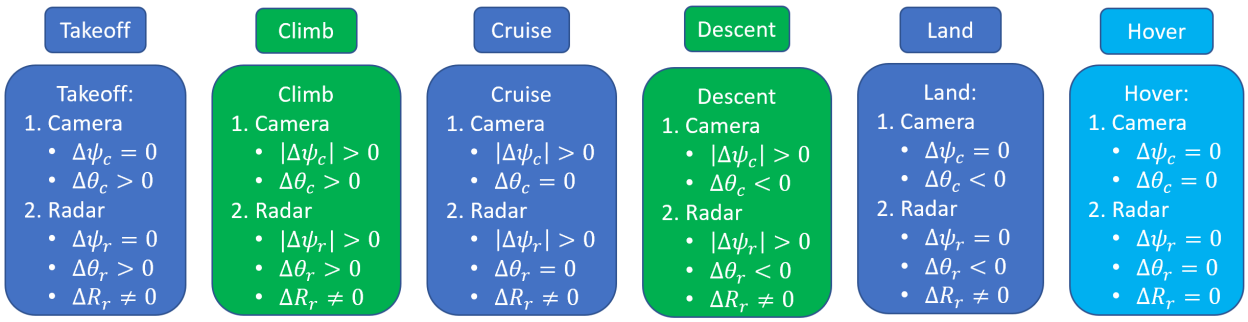



**Fig. 4 HME Flight Phase Classification Diagram: based on the flight phase definitions from Tables 1 & 2**

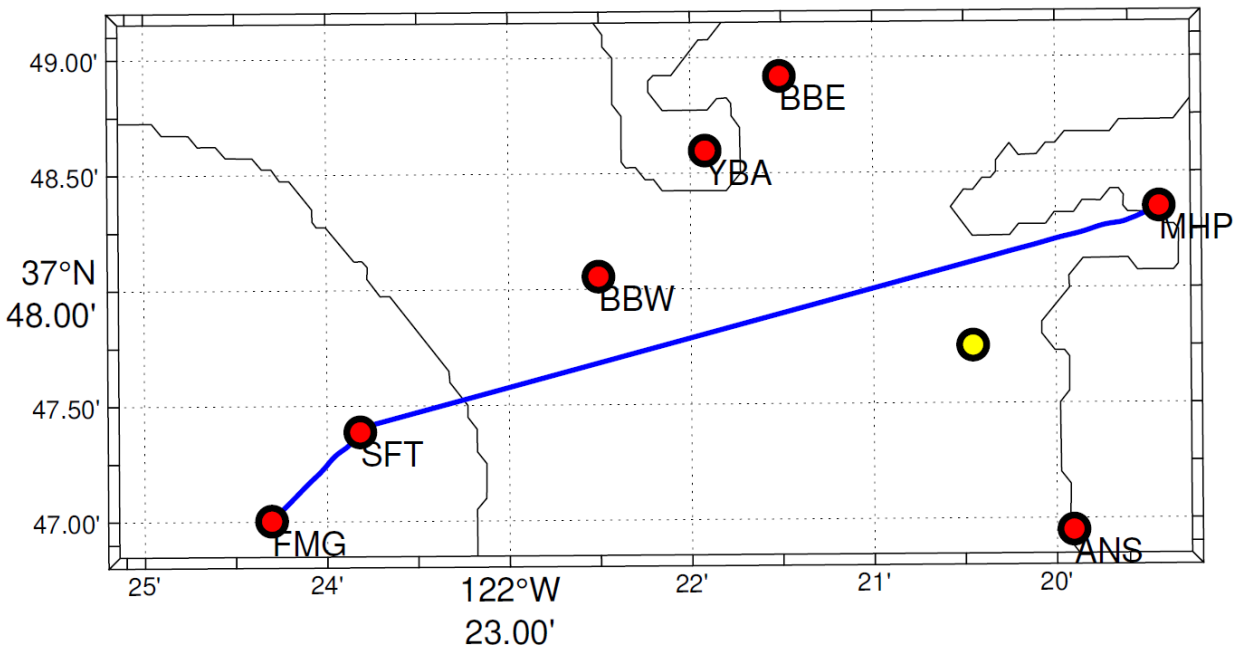
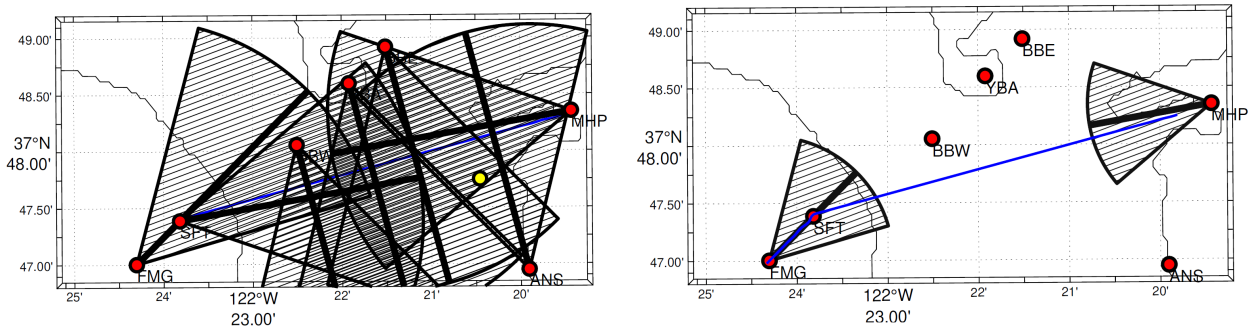



**Fig. 5 Map view of the flight track from Middle Harbor Shoreline Park to Fifth and Mission Garage: flight path (blue line) from MHP to FMG, the distributed sensor locations (red dots), and origin (yellow dot) (Fig. 3 of [32])**

**Table 4 Simulated MHP FMG Sensor Locations**

| Location Name (Abbreviation) | Latitude (°) | Longitude (°) | Alt. (m) | Heading (°) | Elev. (°) |
|---|---|---|---|---|---|
| Mission Harbor Shoreline Park (MHP) | 37.8058404 | -122.3238733 | 4.53 | 260 | 0 |
| Tower of Bay Bridge East Span (BBE) | 37.815309 | -122.358504 | 160 | 160 | 0 |
| Yerba Buena Island Antenna Tower (YBA) | 37.809956 | -122.365332 | 104 | 160 | 0 |
| Alameda Naval Station (ANS) | 37.782494 | -122.331785 | 0 | 340 | 0 |
| Tower of Bay Bridge West Span (BBW) | 37.800911 | -122.375117 | 160 | 160 | 0 |
| Salesforce Tower (SFT) | 37.789782 | -122.396968 | 326 | 80 | 0 |
| Fifth & Mission Garage (FMG) | 37.783379 | -122.405077 | 26 | 45 | 0 |

sensor's coverage area from a top-down view with a solid black line, indicating the centerline of the sensor's field of view. Figure 6a shows the distributed camera coverage with several overlapping areas to provide multiple aircraft tracks. Figure 6b shows the distributed radar coverage, which exists only at the takeoff and landing locations: MHP and FMG. Since MHP and FMG have cameras and radar, HME will have a top-level gating network that combines the camera and radar gating network weights to generate blended sensor flight phase classification solutions.




**(a) Camera Coverage** **(b) Radar Coverage**

**Fig. 6 Sensor coverage maps for all the distributed sensors in the scenario; left: cameras, right: radar - radar colocated with their cameras [32]**

Figure 7 shows the simulated path (red line) from a zoomed-out bird's eye view between the takeoff and landing locations: Middle Harbor Shoreline MHP and FMG (labeled yellow pins). Figure 8 shows a Google Earth zoomed-in view of the simulated AAM aircraft path when it is in San Francisco. It shows the latter portion of the path (red line) in Google Earth from MHP to FMG to see how the simulated AAM aircraft has to carefully weave through the buildings while gradually descending.

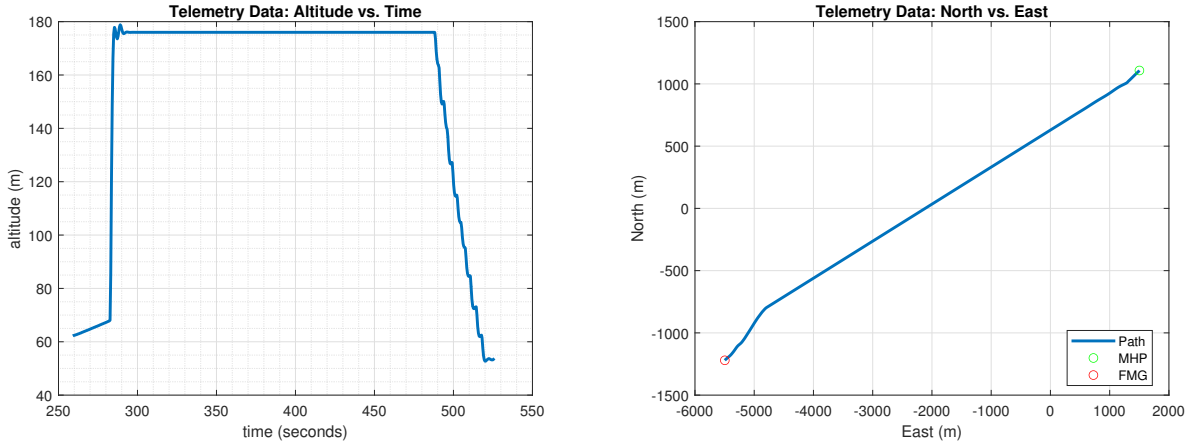

**Fig. 7 Google Earth: MHP to FMG**

Figure 9a shows the AAM/UAM aircraft's altitude vs. time as it flies from MHP to FMG. The initial takeoff phase is vertical with some translation and then an abrupt ascent to approximately 175 m. After a few fluctuations to reach the desired altitude, it begins cruising from Oakland towards San Francisco. It begins its descent phase around 490 seconds when it gets closer to the destination. A staircase pattern occurs throughout the descent because the simulated aircraft descends, moves forward slightly, and then drops down to a lower altitude. Finally, it lands on top of the rooftop at FMG in San Francisco, which is at a slightly lower altitude than its takeoff location at MHP in Oakland.

Figure 9b shows the 2D path of the AAM/UAM aircraft from MHP to FMG. The blue line shows the simulated AAM path, the green circle shows the takeoff location at MHP in Oakland, and the red circle shows the landing location at FMG in San Francisco. Most of the flight is straight between the takeoff and landing phases, and there are some

**Fig. 8 Google Earth: 3D Path FMG**

minor bumps and sharp corners at the beginning and end during the initial and final alignments during takeoff and landing. The caption in Figure 9 shows the human "eye" test results for manually classifying the flight phases based on the telemetry data.




**(a) Telemetry Data: Altitude vs. Time**

**(b) Telemetry Data: North vs. East**

**Fig. 9 Telemetry Data: AAM Trajectory from MHP to FMG (World Coordinate System) - human "eye" test would be climb, fluctuations between climb and descent, cruise, and descent with fluctuations (staircase descent)**

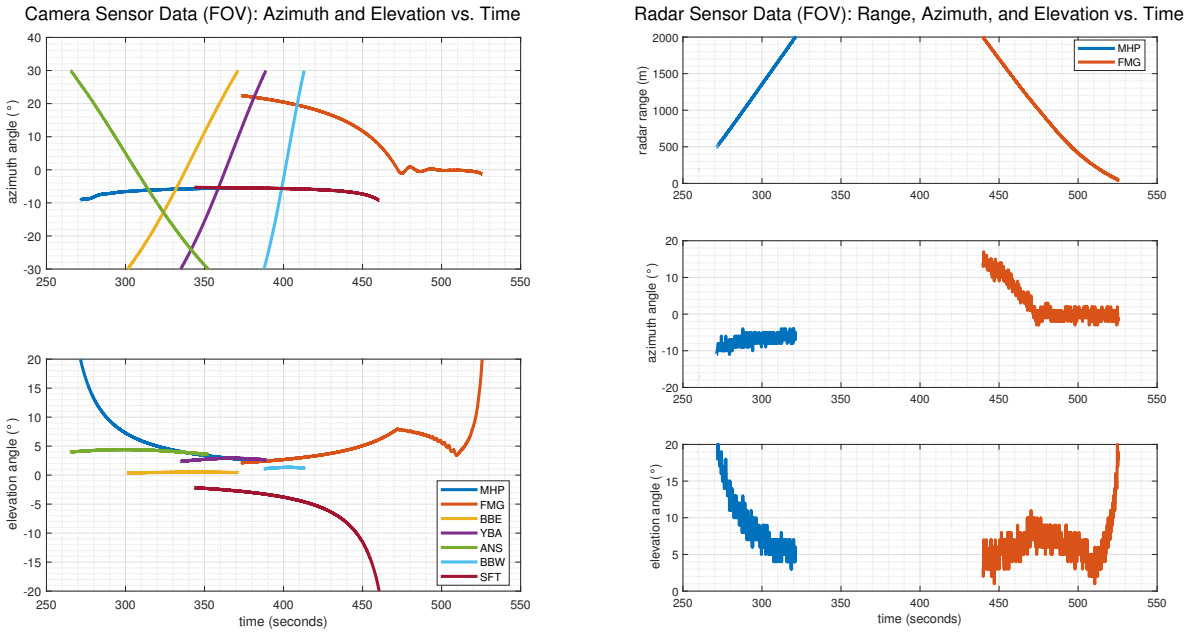
Figure 10 shows the distributed camera and radar's azimuth and elevation angles of the simulated AAM aircraft. Figure 10a shows the azimuth and elevation angles for each camera (seven total), while Figure 10b shows the range and azimuth and elevation angles for the radar at MHP and FMG. The radar data is much noisier and has several fluctuations but has a general trend that follows the camera data.

For example, the camera and radar data at MHP both show a decrease in elevation angle and a slight increase in the azimuth angle. The camera and radar data at FMG show an increase in the elevation angle, then a decrease in elevation angle, and finally, a sharp increase in elevation as the simulated aircraft is nearly out of the field of view during the final moments in the landing phase. Overall, Table 5 shows the human "eye" test results for classifying the sensor data for manually classifying the flight phases based on the individual sensor's measurements. For example, based on the camera and radar data for MHP, one would expect descent to be the main flight phase since the elevation angle decreases and the azimuth angle increases (moving right relative to the camera's field of view).

For the simulation results in this paper, set $\alpha = 0.05$ in Eqn. (2) such that the overall derivative of the sensor

**Table 5 Human Eye Test for Flight Phase Classification: Sensor Data**

| Location | Sensor Type | Flight Phases |
|---|---|---|
| ANS | camera | climb and then desecent |
| BBE | camera | cruise or switches between climb & descent |
| BBW | camera | cruise or a switch from climb to descent |
| FMG | camera | climb, descent, & climb |
| FMG | radar | climb, descent, & climb |
| MHP | camera | descent |
| MHP | radar | descent |
| SFT | camera | descent |
| YBA | camera | climb then descent |




**(a) Cameras: Azimuth and Elevation Angles**

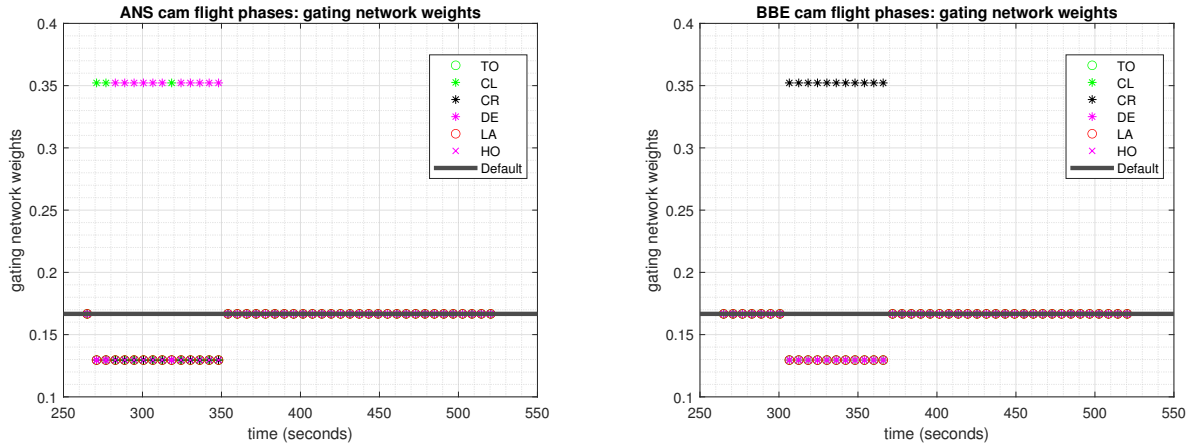
**(b) Radar: Azimuth & Elevation Angles and Range**

**Fig. 10 Distributed Camera and Radar Data: the radar measurements tend to be noisier than the camera measurements and have larger fluctuations and spikes, especially in the azimuth and elevation angles [32].**

measurements weighs the previous EMA derivative heavier than the current three-point backward derivative, which makes the flight phase classification results smoother. The step size between iterations is about 6 seconds. It is long enough to view sufficient movement between iterations since the AAM aircraft appears very small, causing the aircraft to have tiny movements between smaller step sizes.

The subsequent figures show the gating network weights for each of the sensors. The horizontal black line at approximately 0.16 is the default value for the gating network weights, which makes sense since $1/6 \approx 0.16$. In other words, all flight phases have the same weights because the sensor lacks measurements during this time.

Figure 11 shows the gating network weights for the cameras at Alameda Naval Station (ANS) and the tower of Bay Bridge East Span (BBE). ANS has climb and descent with a climb near the middle during a switch from climb to descent, so cruise occurs between them. However, the step size is slightly too big, so HME misses cruise. BBE has only cruise without any switches between climb and descent. Overall, the gating network weights for ANS and BBE follow the expected human "eye" test results from Table 5.




**(a) Gating Network Weights: Camera - ANS**

**(b) Gating Network Weights: Camera - BBE**

**Fig. 11 Gating Network Weights the Cameras: ANS & BBE**

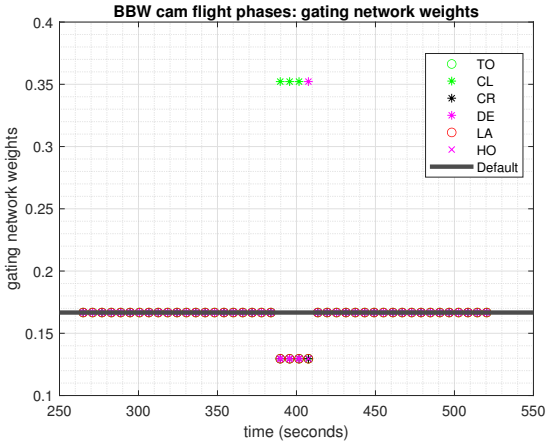
Figure 12 shows the gating network weights for the cameras at the Bay Bridge West tower (BBW) and Fifth & Mission Garage (FMG). BBW has climb and switches to descent, so cruise occurs between them. However, the step size is slightly too big, so HME misses cruise. FMG has climb and descent, and its first switch occurs as expected. The next switch occurs during the landing phase as it has a staircase pattern, as shown in Figure 9a. The last switch to climb occurs as the aircraft flies over the sensor and out of the field of view. Overall, the gating network weights for BBW and FMG follow the expected human "eye" test results from Table 5.

Figure 13 shows the gating network weights for the cameras at Middle Harbor Shoreline Park (MHP) and Salesforce Tower (SFT). As expected, MHP has only descent. SFT has climb and descent with its first switch near the beginning and another switch from climb back to descent shortly after. A couple of hits for cruise should occur if the step size were smaller, but ultimately, SFT has mostly descent, as expected. Overall, the gating network weights for MHP and SFT follow the expected human "eye" test results from Table 5 despite the two small switches in SFT near the beginning.

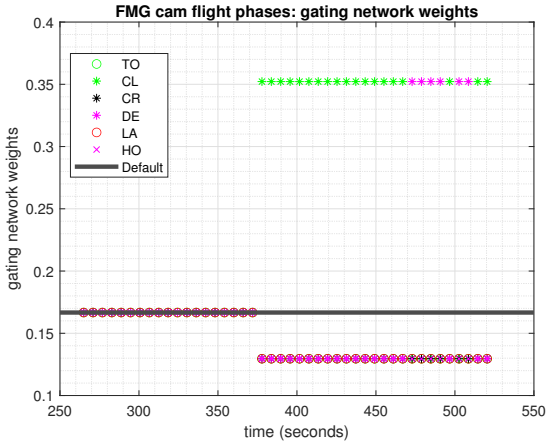
Figure 14 shows the gating network weights for Yerba Buena Island Antenna Tower (YBA). YBA has mostly climb in the beginning and middle. It also has a couple of switches between climb and descent, so cruise occurs between them. However, the step size is slightly too big, so HME misses cruise for YBA. Overall, the gating network weights for YBA follow the expected human "eye" test results from Table 5 with climb and then descent.

Figure 15 shows the gating network weights for the radar at Middle Harbor Shoreline Park (MHP) and Salesforce Tower (SFT). Since the radar data is noisier than the camera data, more measurement fluctuations and spikes can cause more switches between flight phases. For example, there are switches between climb and descent in the gating network weights for MHP's radar even though MHP's camera showed only descent. Similarly, FMG's radar shows switches between climb and descent in the gating network weights.

Since there are cameras and radar at MHP and FMG, combining their gating network weights yields a set of fused gating network weights. Figure 16 shows the combined gating network weights for the cameras and radar at Middle

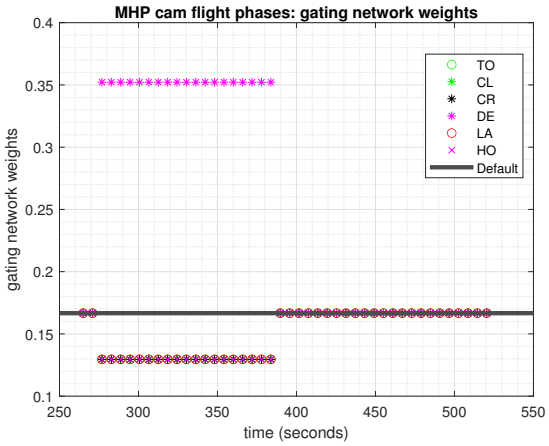



(a) Gating Network Weights: Camera - BBW

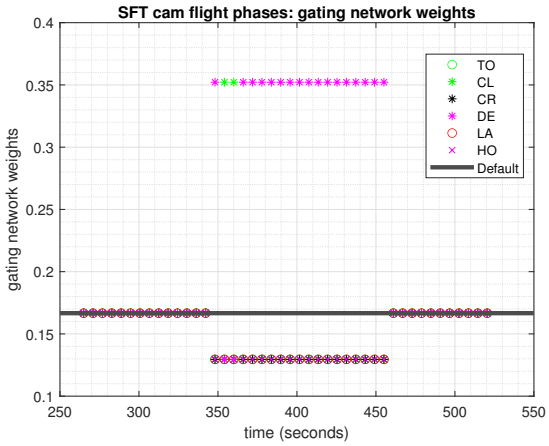



(b) Gating Network Weights: Camera - FMG

**Fig. 12 Gating Network Weights the Cameras: BBW & FMG**




(a) Gating Network Weights: Camera - MHP




(b) Gating Network Weights: Camera - SFT

**Fig. 13 Gating Network Weights: Cameras - MHP & SFT**

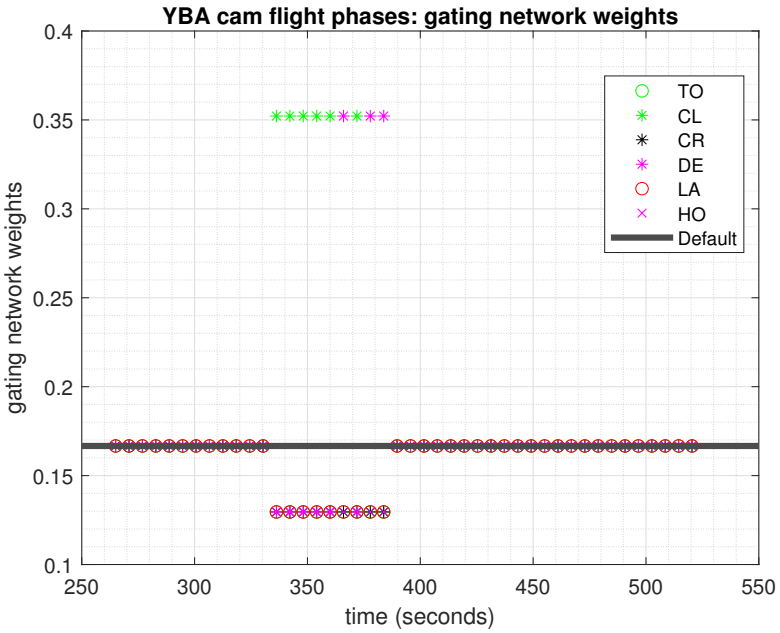
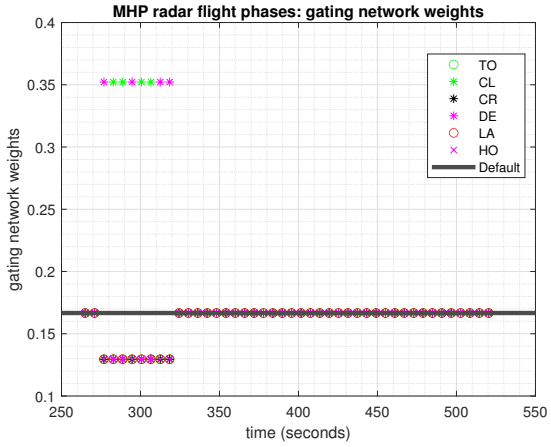
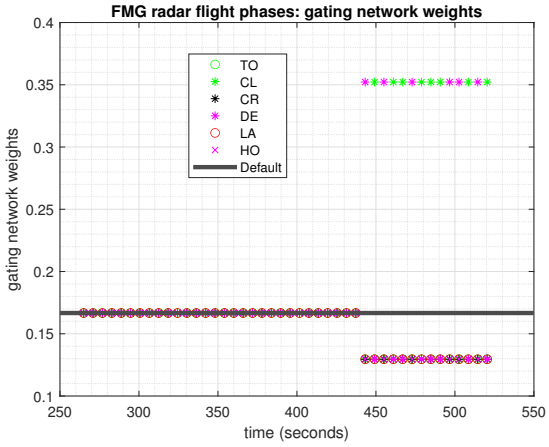



**Fig. 14 Gating Network Weights: Camera - YBA**




**(a) Gating Network Weights: Radar - MHP**



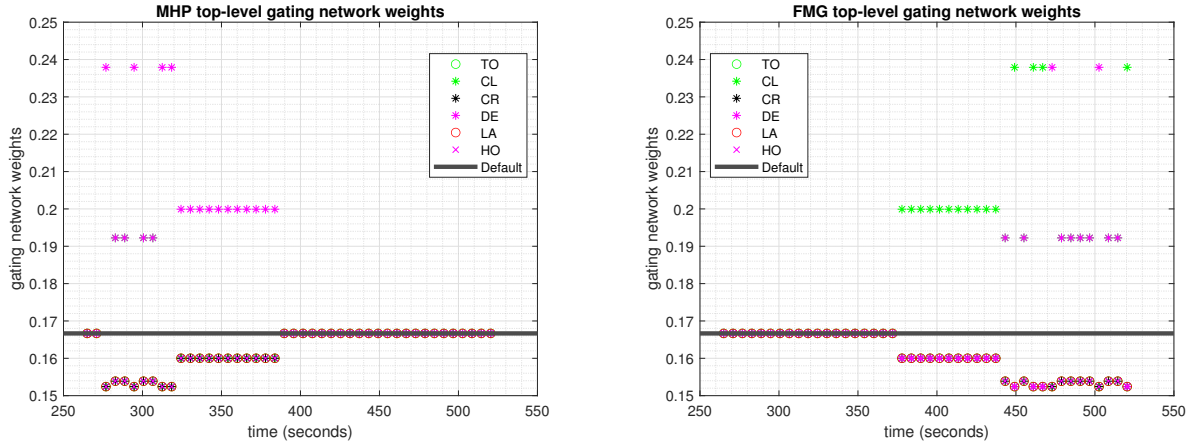

**(b) Gating Network Weights: Radar - FMG**

**Fig. 15 Gating Network Weights: Cameras - MHP & SFT**

Harbor Shoreline Park (MHP) and Fifth & Mission Garage (FMG). The top-level (TL) gating network weights for MHP and FMG are:

$$g_{TL,i} = \frac{exp(g_{cam,i,j} + g_{radar,i,j})}{\Sigma_j exp(g_{cam,i,j} + g_{radar,i,j})} \tag{4}$$

such that $i$ is the number of iterations. The variable, $j$, is the index for the different flight phases with 1 for takeoff, 2 for climb, 3 for cruise, 4 for descent, 5 for land, and 6 for hover. Overall, the fused gating network weights for MHP show mostly descent, and the fused gating network weights for FMG show mostly climb, then descent with some fluctuations between climb and descent, and then climb at the end.




**(a) Combined Gating Network Weights: MHP** **(b) Combined Gating Network Weights: FMG**

**Fig. 16 Combined Gating Network Weights Cameras: MHP & SFT**

Recall that Figure 9 shows the AAM aircraft telemetry data with Figure 9a showing altitude vs. time and Figure 9b showing the East-North coordinates (inertial world coordinate system). Instead of considering the individual sensors' flight phase classification, applying HME on the telemetry position data provides insight for a global or holistic flight phase classification by using the telemetry flight phase classification definitions in Table 3.

Figure 17 shows the flight phase classifications of the simulated AAM trajectory by applying HME on the telemetry data with a step size of about 5 seconds. Notice that the gating network weights match the human "eye" test results from the caption in Figure 9: mostly climb at the beginning, some fluctuations between climb and descent, mainly cruise, and then mostly descent with some switches between climb and descent during the staircase descent for the landing phase.

Table 6 shows the number of times each flight phase showed up per camera during the simulated flight from MHP to FMG. The climb and descent flight phases are the most common. One-dimensional vertical ascents and descents never occurred, so takeoff and landing have zeros. There were no moments without translation or rotation, so hover had zero hits.

**Table 6 Number of Flight Phases for the Distributed Cameras**

| | ANS | BBE | BBW | FMG | MHP | SFT | YBA |
|---|---|---|---|---|---|---|---|
| Takeoff | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Climb | 3 | 0 | 3 | 19 | 0 | 2 | 6 |
| Cruise | 0 | 11 | 0 | 0 | 0 | 0 | 0 |
| Descent | 11 | 0 | 1 | 6 | 19 | 17 | 3 |
| Land | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Hover | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 7 shows the total number of flight phases for the radar at MHP and FMG. There is a somewhat even distribution between climb and descent due to the noisy radar measurements, which cause fluctuations and spikes in the sensor

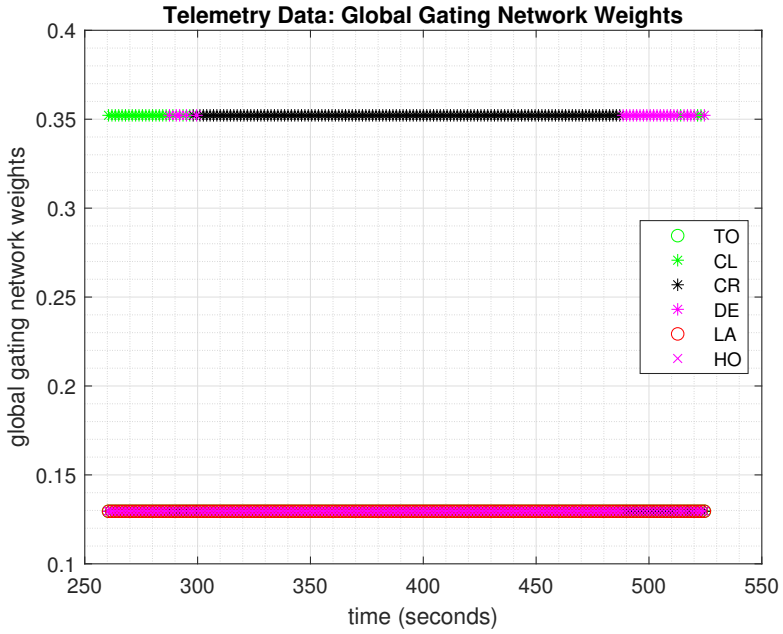



**Fig. 17 HME Global Flight Phase Classification: AAM Telemetry Data**

data. Table 8 shows the total number of flight phases for the telemetry data. As expected, most of the flight is on cruise, and the other two categories are climb and descent. Takeoff, land, and hover have zero hits since there is a lack of one-dimensional vertical ascents and descents and loitering idly (hovering). Overall, HME performs flight phase

**Table 7 Flight Phase Count: Radar**

| | FMG | MHP |
|---|---|---|
| Takeoff | 0 | 0 |
| Climb | 8 | 4 |
| Cruise | 0 | 0 |
| Descent | 6 | 4 |
| Land | 0 | 0 |
| Hover | 0 | 0 |

**Table 8 Flight Phase Count: AAM Telemetry**

| | Telemetry Data |
|---|---|
| Takeoff | 0 |
| Climb | 24 |
| Cruise | 126 |
| Descent | 27 |
| Land | 0 |
| Hover | 0 |

classification very quickly for the simulated AAM trajectory from MHP to FMG. Classifying the camera and radar measurements takes an average of only 2.64 milliseconds per iteration. It runs even faster for classifying the telemetry data's flight phases, with an average runtime per iteration of only 0.3037 microseconds.

### B. Post-Processed UAS Flight Test Results

UAS flight tests near the NASA Ames Research Center DART site provide experimental data to test the HME flight phase classification architecture via post-processing. Figures 18 and 19 show the UAV path (blue line) during the flight tests at NASA Ames Research Center. Figure 20 shows the approximate sensor locations (yellow circles), and Figure 21 shows their real locations based on the camera calibration at the flight test with $GN_i$ indicating ground node $i$. Table 9 lists each sensor station's locations and viewing angles. The heading angles are relative to true north with clockwise being positive, and the elevation angles are relative to the center of the camera's images with counterclockwise being positive, i.e., a positive elevation angle means the camera tilts up. The origin, O, lacks sensors and has no heading or elevation angles.

Figure 22a shows the UAS altitude vs. time as it flies from the top left of the NASA Ames campus, follows the road

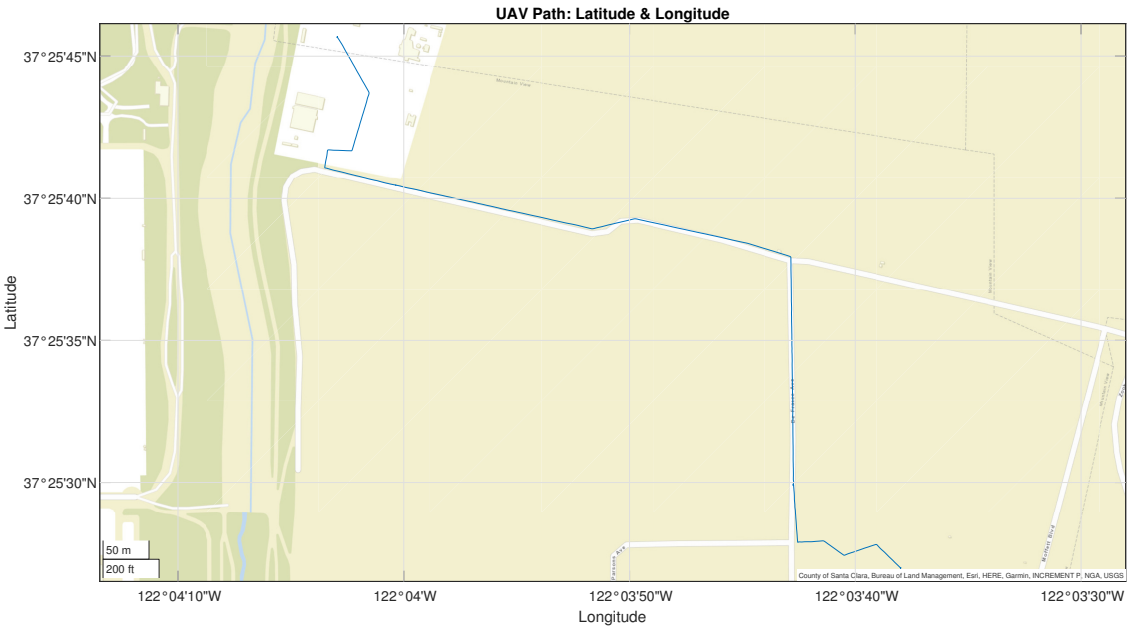



**Fig. 18 UAV Path: Latitude & Longitude**

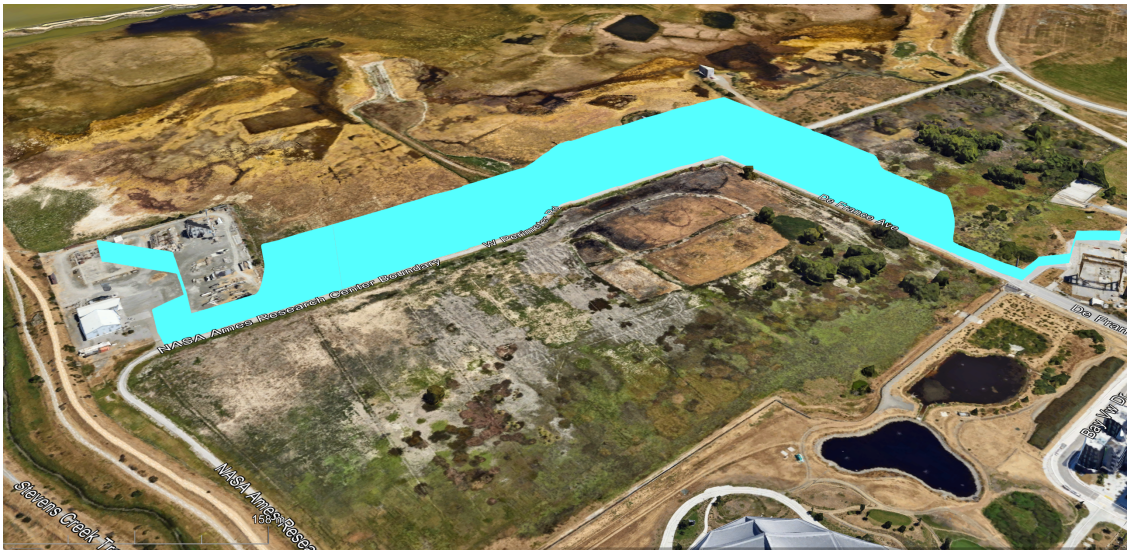



**Fig. 19 UAV Path: Google Earth**

**Table 9 NASA Ames Build 1 Scenario Sensor Locations and Angles**

| Location | Latitude (°) | Longitude (°) | Alt. (m) | Heading (°) | Elev. (°) | Range (m) |
|---|---|---|---|---|---|---|
| 1 | 37.429540 | -122.067712 | 1.54 | 66.53 | 9.68 | 600 |
| 2 | 37.425155 | -122.059832 | 1.54 | 246.54 | 29.29 | 600 |
| 3 | 37.423476 | -122.061238 | 1.57 | 303.14 | 34.0 | 600 |
| 4 | 37.426551 | -122.067970 | 1.52 | 44.78 | 26.31 | 600 |
| O | 37.426861 | -122.066015 | 0 | N/A | N/A | N/A |


**Fig. 20 Ames Research Center Flight Test - Build 1: Google Earth - approximate sensor station locations (yellow), origin (black dot), & UAS path (purple line) [31, 34]**

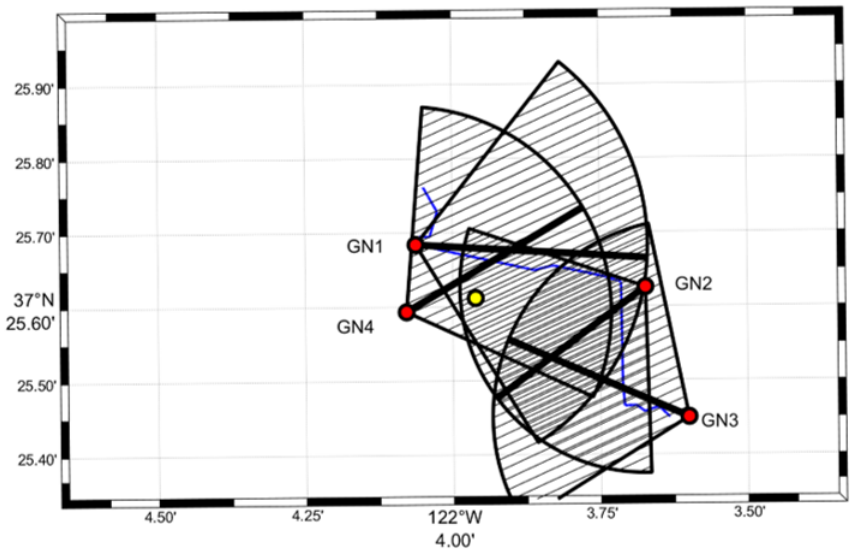
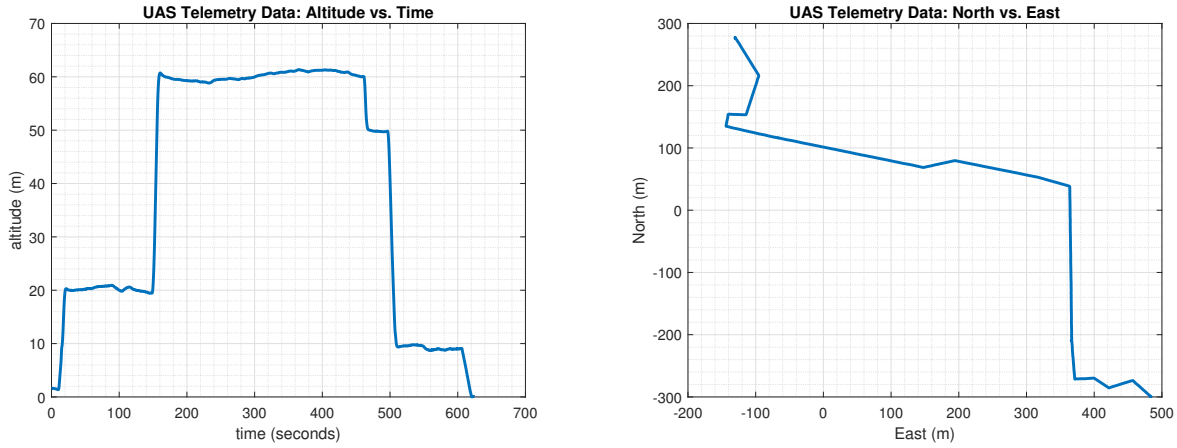



**Fig. 21 Ames Research Center Flight Test - Build 1: actual sensor station locations (red dots), black cones show the camera's coverage, and the black line in the middle of each cone represents the centerline of the camera's field of view**

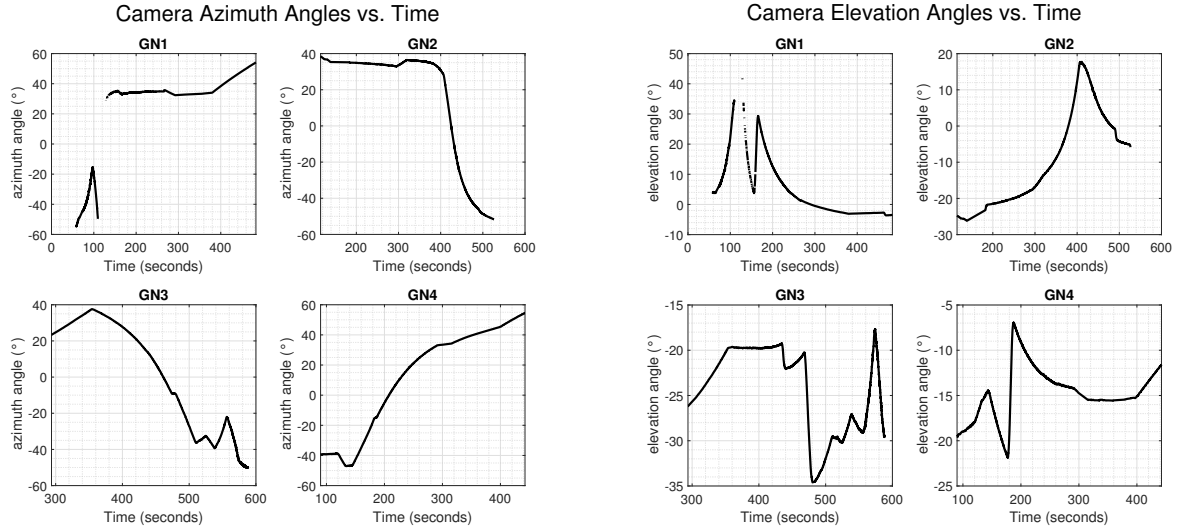
to the east and south, and finally lands after flying slightly east. The altitude has some fluctuations due to GPS noise. The caption in Figure 22 shows the human "eye" test results for manually classifying the flight phases based on the telemetry data: mostly climb and descent with some possible hits on cruise in the middle.




**(a) UAS Telemetry Data: Altitude vs. Time** **(b) Telemetry Data: North vs. East**

**Fig. 22 Telemetry Data: AAM Trajectory from MHP to FMG (World Coordinate System) - human "eye" test would be climb, fluctuations between climb and descent with some hits on cruise, and descent with fluctuations (short staircase descent)**

Figures 23a and 23b show the synced camera azimuth and elevation angles for the four ground-based cameras. The camera at ground node 1 ($GN_1$) has a gap near the beginning because the UAS leaves the camera's field of view at the top left and returns within its field of view at the top right.




**(a) Distributed Cameras: Azimuth Angles** **(b) Distributed Cameras: Elevation Angles**

**Fig. 23 NASA Ames Build 1 Scenario: Camera Azimuth and Elevation Angles vs. Time**

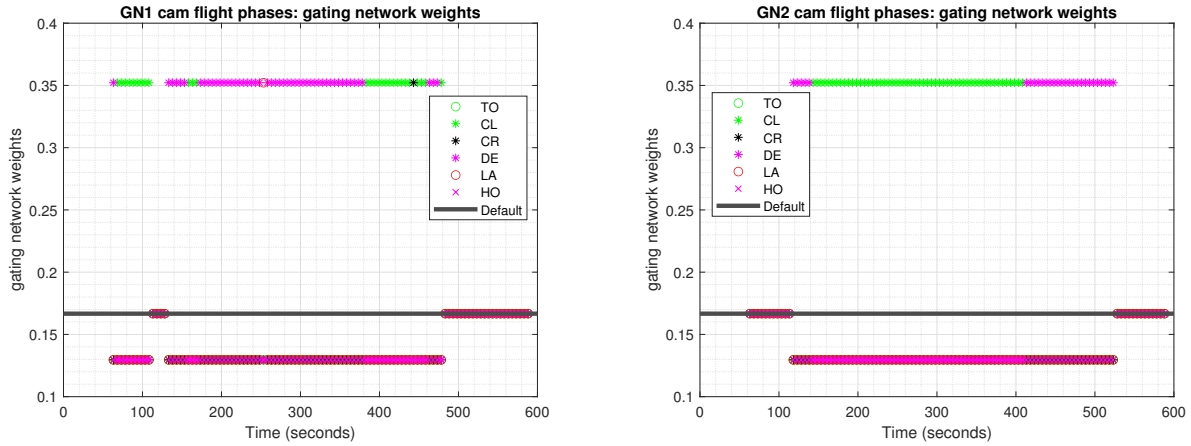
Table 10 shows the flight phase classification based on the human "eye" test. Looking at the camera azimuth and elevation angles in Figure 23, most flight phases should be classified as climb and descent with some switches between them. Due to the switches and step size, there could also be a few hits on cruise.

Figures 24 and 25 show the gating network weights for the ground node cameras. Ground node 1's camera has

**Table 10 Human Eye Test for Flight Phase Classification: UAS Telemetry Data**

| Camera | Flight Phases |
|---|---|
| $GN_1$ | climb, descent, switch between climb & descent, descent, and slight climb |
| $GN_2$ | descent, climb, and descent |
| $GN_3$ | climb, switches between climb & descent, descent, climb with switches between descent, climb, and descent |
| $GN_4$ | climb, descent, climb, descent, and climb |

mostly descent in the beginning and middle with a couple of switches with climb before ending in mostly climb. One of the switches gets a hit for cruise due to aligning with the step size. As expected, ground node 2's camera has an initial descent, climb, and ends in descent. At ground node 3, the camera has climb with some switches between climb and descent, and then ends with descent. It also has one hit for cruise during one of the switches. Ground node 4's camera starts with climb, switches to descent back to climb, has mostly descent, and then ends with climb. There are also two hits for cruise in ground node 4's flight phase classification. The first hit for the cruise is in the middle of the descent at approximately 270 seconds, which occurs due to deceleration in altitude during descent, causing the UAS to level off briefly. Consequently, there is a short moment when the altitude remains constant, registering as cruise. The second hit for cruise occurs between a switch between climb and cruise at approximately 340 seconds. Overall, as expected, several hits for climb and descent occur throughout the UAS trajectory from each camera, which matches the human "eye" test results despite the few instances with cruises during switches or deceleration in the vertical direction.




**(a) Gating Network Weights: GN1**

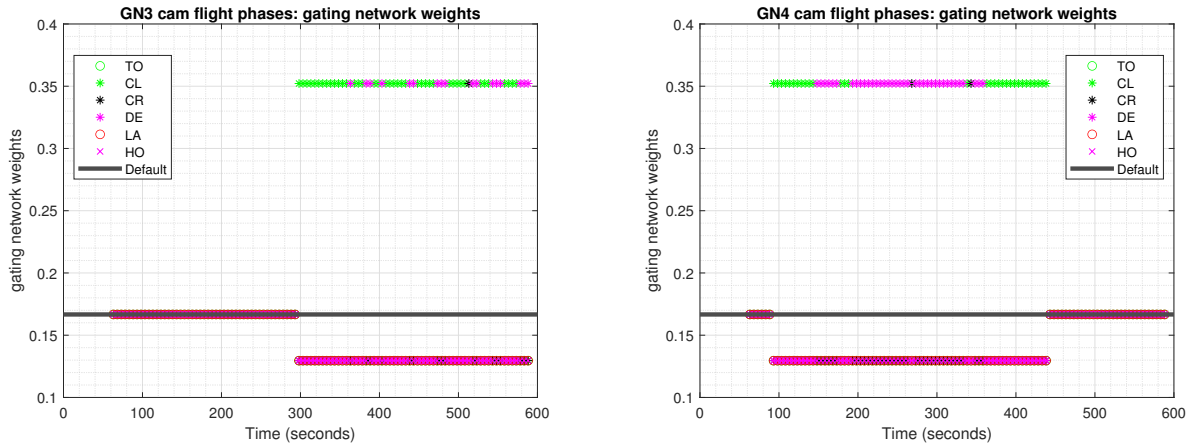
**(b) Gating Network Weights: GN2**

**Fig. 24 Gating Network Weights: GN1 & GN2**

Figure 26 shows the global flight phase classification gating network weights for the UAS telemetry data based on the telemetry flight phase definitions in Table 3. Notice that the gating network weights match the human "eye" test results from the caption in Figure 22: mostly climb at the beginning, some fluctuations between climb and descent or hits on cruise, and then mostly descent during the short staircase descent for the landing phase. It also has five hits for cruise during switches between climb and descent.

Table 11 shows the individual camera's flight phase counts. Table 12 shows the total number of flight phases based on the UAS telemetry data after applying HME on the telemetry data to provide global flight phase classification. There are some fluctuations in the altitude, which will cause switches between climb and descent. Overall, the average runtime for global flight phase classification of the UAS telemetry data is 0.001 ms, which is very fast. The AAM simulation and post-processed UAS flight test results have very short runtimes. Therefore, future ATM/ATC architectures tailored for AAM/UAM operations can implement HME flight phase classification for real-time airspace monitoring.

Future work can include creating a framework that combines all the distributed sensors' flight phase classifications to merge into a global flight phase classification. The main two pre-requisites include: 1) setting up the distributed sensors such that there are at least two overlapping fields of view to have at least two measurements on the same target




**(a) Gating Network Weights: GN3** **(b) Gating Network Weights: GN4**

**Fig. 25 Gating Network Weights: GN3 & GN4**

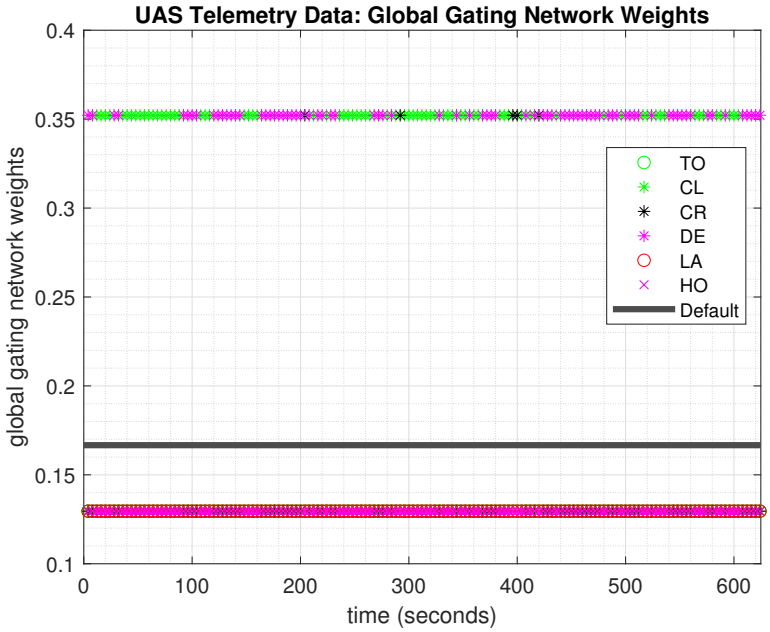



**Fig. 26 HME Global Flight Phase Classification: UAS Telemetry Data**

**Table 11 Flight Phase Count: GN1-GN4**

| | GN1 | GN2 | GN3 | GN4 |
|---|---|---|---|---|
| Takeoff | 0 | 0 | 0 | 0 |
| Climb | 28 | 54 | 41 | 31 |
| Cruise | 1 | 0 | 1 | 2 |
| Descent | 46 | 28 | 17 | 37 |
| Land | 1 | 0 | 0 | 0 |
| Hover | 0 | 0 | 0 | 0 |

**Table 12 Flight Phase Count: UAS Telemetry**

| | Telemetry |
|---|---|
| Takeoff | 0 |
| Climb | 73 |
| Cruise | 5 |
| Descent | 78 |
| Land | 0 |
| Hover | 0 |

and 2) matching and associating the target among all the sensors such that the sensors know they are looking at the same target but from their unique, individual perspectives.

## V. Conclusion

Utilizing an HME architecture for flight phase classification through distributed sensing provides deeper insight into AAM/UAM flight phase classification. The proposed solution runs in real-time and does not require any training of neural networks to learn the different flight phases, which addresses common AI drawbacks such as overfitting and long training times. The HME architecture captures the transitional flight phases to yield higher fidelity for flight phase classification, which does not occur in traditional binary-type classification methods like decision trees or binary methods (all or nothing). Even though hover does not appear in the camera, radar, or telemetry flight phase classifications, it is a vital flight phase to include for future ATM/ATC operations for monitoring aircraft that will yield to emergency response aircraft. Future work can combine the distributed sensors' flight phase classifications to generate a global classification. Ultimately, having higher fidelity flight phase classification leads to higher fidelity tracking estimation solutions for ATC/ATM-like structures for future AAM/UAM operations.

## Acknowledgments

The authors would like to thank the Transformative Tools and Technologies (TTT) Project under the NASA Aeronautics Research Mission Directorate (ARMD) for funding this research.

## References

[1] de Oliveira, Í. R., Neto, E. C. P., Matsumoto, T. T., and Yu, H., "Decentralized air traffic management for advanced air mobility," *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, IEEE, 2021, pp. 1–8. https://doi.org/10.1109/ICNS52807.2021.9441552.

[2] Bauranov, A., and Rakas, J., "Designing airspace for urban air mobility: A review of concepts and approaches," *Progress in Aerospace Sciences*, Vol. 125, 2021, p. 100726. https://doi.org/10.1016/j.paerosci.2021.100726.

[3] Vascik, P. D., and Hansman, R. J., "Scaling constraints for urban air mobility operations: Air traffic control, ground infrastructure, and noise," *2018 aviation technology, integration, and operations conference*, 2018, p. 3849. https://doi.org/10.2514/6.2018-3849.

[4] (CAST), C. A. S. T., and (CICTT), I. C. A. O. I. C. T. T., "Phase of Flight - Definitions and Usage notes," , 2013. URL https://www.ntsb.gov/safety/data/Documents/datafiles/PhaseofFlightDefinitions.pdf.

[5] Federal Aviation Administration, "Airplane Flying Handbook (FAA-H-8083-C)," U.S. Department of Transportation, 2021. URL https://www.faa.gov/sites/faa.gov/files/regulations_policies/handbooks_manuals/aviation/airplane_handbook/00_afh_full.pdf.

[6] Lombaerts, T., Kaneshige, J., and Feary, M., "Control concepts for simplified vehicle operations of a quadrotor eVTOL vehicle," *AIAA Aviation 2020 Forum*, 2020, pp. 1–31. https://doi.org/10.2514/6.2020-3189.

[7] Roskam, J., *Airplane flight dynamics and automatic flight controls*, DARcorporation, 1998.

[8] Federal Aviation Administration, "Urban Air Mobility (UAM) Concept of Operations v2.0," U.S. Department of Transportation, 2023. URL https://www.faa.gov/sites/faa.gov/files/Urban%20Air%20Mobility%20%28UAM%29%20Concept%20of%20Operations%202.0_0.pdf.

[9] Tian, F., Cheng, X., Meng, G., and Xu, Y., "Research on flight phase division based on decision tree classifier," *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, IEEE, 2017, pp. 372–375. https://doi.org/10.1109/CIAPP.2017.8167242.

[10] Peysakhovich, V., Ledegang, W., Houben, M., and Groen, E., "Classification of flight phases based on pilots' visual scanning strategies," *2022 Symposium on Eye Tracking Research and Applications*, 2022, pp. 1–7. https://doi.org/10.1145/3517031.3529641.

[11] Zhang, Q., Mott, J. H., Johnson, M. E., and Springer, J. A., "Development of a reliable method for general aviation flight phase identification," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 8, 2021, pp. 11729–11738. https://doi.org/10.1109/TITS.2021.3106774.

[12] Kuzmenko, N., Averyanova, Y., Zaliskyi, M., Ostroumov, I., Larin, V., Solomentsev, O., Bezkorovainyi, Y., and Sushchenko, O., "Airplane Flight Phase Identification Using Maximum Posterior Probability Method," *2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC)*, 2022, pp. 1–5. https://doi.org/10.1109/SAIC57818.2022.9922913.

[13] Masoudnia, S., and Ebrahimpour, R., "Mixture of experts: a literature survey," *Artificial Intelligence Review*, Vol. 42, 2014, pp. 275–293.

[14] Jordan, M. I., and Jacobs, R. A., "Hierarchical mixtures of experts and the EM algorithm," *Neural computation*, Vol. 6, No. 2, 1994, pp. 181–214. https://doi.org/10.1162/neco.1994.6.2.181.

[15] Waterhouse, S. R., and Robinson, A. J., "Classification using hierarchical mixtures of experts," *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, IEEE, 1994, pp. 177–186. https://doi.org/10.1109/NNSP.1994.366050.

[16] Ramamurti, V., and Ghosh, J., "Advances in using hierarchical mixture of experts for signal classification," *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 6, IEEE, 1996, pp. 3569–3572.

[17] Titsias, M. K., and Likas, A., "Mixture of experts classification using a hierarchical mixture model," *Neural Computation*, Vol. 14, No. 9, 2002, pp. 2221–2244. https://doi.org/10.1162/089976602320264060.

[18] Bishop, C. M., and Svensén, M., "Bayesian hierarchical mixtures of experts," *arXiv preprint arXiv:1212.2447*, 2012.

[19] Peng, F., Jacobs, R. A., and Tanner, M. A., "Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition," *Journal of the American Statistical Association*, Vol. 91, No. 435, 1996, pp. 953–960. https://doi.org/10.1080/01621459.1996.10476965.

[20] Belsis, P., Fragos, K., Gritzalis, S., and Skourlas, C., "SF-HME system: a hierarchical mixtures-of-experts classification system for spam filtering," *Proceedings of the 2006 ACM symposium on Applied computing*, 2006, pp. 354–360.

[21] Ng, S.-K., and McLachlan, G. J., "Extension of mixture-of-experts networks for binary classification of hierarchical data," *Artificial Intelligence in Medicine*, Vol. 41, No. 1, 2007, pp. 57–67.

[22] Jafarpour, N., Precup, D., Izadi, M., and Buckeridge, D., "Using hierarchical mixture of experts model for fusion of outbreak detection methods," *AMIA Annual Symposium Proceedings*, Vol. 2013, American Medical Informatics Association, 2013, p. 663.

[23] Zhou, Y., and Kantarcioglu, M., "Adversarial learning with bayesian hierarchical mixtures of experts," *Proceedings of the 2014 SIAM International Conference on Data Mining*, SIAM, 2014, pp. 929–937.

[24] Chaer, W. S., Bishop, R. H., and Ghosh, J., "A mixture-of-experts framework for adaptive Kalman filtering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 27, No. 3, 1997, pp. 452–464. https://doi.org/10.1109/3477.584952.

[25] Chaer, W. S., Bishop, R. H., and Ghosh, J., "Hierarchical adaptive Kalman filtering for interplanetary orbit determination," *IEEE transactions on aerospace and electronic systems*, Vol. 34, No. 3, 1998, pp. 883–896. https://doi.org/10.1109/7.705895.

[26] Kawamura, E., Azimov, D., Allen, J. S., and Ippolito, C., "Hierarchical mixture of experts for autonomous unmanned aerial vehicles utilizing thrust models and acoustics," *Robotics and Autonomous Systems*, Vol. 162, 2023, p. 104369. https://doi.org/10.1016/j.robot.2023.104369.

[27] Kawamura, E., and Azimov, D., "Mixture of Experts for Unmanned Aerial Vehicle Motor Thrust Models," *ASME Letters in Dynamic Systems and Control*, Vol. 3, No. 1, 2023, p. 011004. https://doi.org/10.1115/1.4056935.

[28] Ippolito, C. A., Hashemi, K., Kawamura, E., Gorospe, G., Holforty, W., Kannan, K., Stepanyan, V., Lombaerts, T., Brown, N., Jaffe, A., et al., "Concepts for Distributed Sensing and Collaborative Airspace Autonomy in Advanced Urban Air Mobility," *AIAA SciTech 2023 Forum*, 2023, p. 20.

[29] Kawamura, E., Dolph, C., Kannan, K., Lombaerts, T., and Ippolito, C. A., "Simulated Vision-based Approach and Landing System Advanced Air Mobility," *AIAA SciTech 2023 Forum*, AIAA-2023-2195, 2023. https://doi.org/10.2514/6.2023-2195.

[30] Kawamura, E., Kannan, K., Lombaerts, T., Stepanyan, V., Dolph, C., and Ippolito, C. A., "Ground-Based Vision Tracker for Advanced Air Mobility and Urban Air Mobility," *AIAA SciTech 2024 Forum*, 2024.

[31] Stepanyan, V., Kannan, K., Kawamura, E., Lombaerts, T., and Ippolito, C., "Target Tracking with Distributed Sensing and Optimal Data Migration," *AIAA SciTech 2023 Forum*, AIAA-2023-2194, 2023, p. 16. https://doi.org/10.2514/6.2023-2194.

[32] Lombaerts, T., Kannan, K., Kawamura, E., Dolph, C., Stepanyan, V., George, G., and Ippolito, C., "Distributed Ground Sensor Fusion Based Object Tracking for Autonomous Advanced Air Mobility Operations," *AIAA SciTech 2023 Forum*, AIAA-2023-0896, 2023. https://doi.org/10.2514/6.2023-0896.

[33] Chapra, S. C., *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 3rd ed., The McGraw-Hill Companies, Inc., 2012.

[34] Kannan, K., Baculi, J., Lombaerts, T., Kawamura, E., Gorospe, G., Holforty, W., Ippolito, C., Stepanyan, V., Dolph, C., and Brown, N., "A Simulation Architecture for Air Traffic Over Urban Environments Supporting Autonomy Research in Advanced Air Mobility," *AIAA SciTech 2023 Forum*, AIAA-2023-0895, 2023, pp. 1–17. https://doi.org/10.2514/6.2023-0895.