Cross Validation of Sensor Inputs as Part of Position Estimation for Urban Air Mobility

Thomas Lombaerts* Evan Kawamura† Keerthana Kannan‡ Corey Ippolito§ Vahram Stepanyan¶ George Gorospe‖
NASA Ames Research Center, Moffett Field, CA 94035

Chester Dolph**
NASA Langley Research Center, Langley, VA 94035

Large scale networks of redundant ground nodes with heterogeneous sensors need some form of cross validation to determine which measurements are most trustworthy. One way to do this is residual based, by evaluating the residual between real measurements and 'estimated measurements' based on the estimated position. These residuals are an indicator of the reliability of the individual ground nodes which varies in time. One of the possible methods is based on the hierarchical mixture of experts (HME) and is discussed in this paper. This info can later be used in the Kalman Filter to improve performance (this is future work and not part of this paper).

I. Introduction

Recently, there have been numerous developments in the area of advanced air taxi operations. Combined with other novel transportation concepts such as autonomous aerial cargo delivery etc., they form the Advanced Air Mobility (AAM) ecosystem that will help to meet the challenging mobility needs of an increasing population density in metropolitan areas, taking into account the limited capacity of the currently available transportation infrastructure. To realize the full vision of AAM, technology will be needed to allow a few operators to operate many vehicles (m:N). As most m:N implementations will require highly autonomous vehicles, the current state-of-the-art will have to advance for on-board autonomous functionality. One of the key areas requiring further technology development is sensor-based perception techniques. For this perception, techniques are being developed to address problems such as GPS-free Alternative Positioning, Navigation and Timing (APNT) [1]. Advanced sensing and tracking algorithms will be necessary from the ground in an urban environment, where traditional traffic tracking infrastructure such as air traffic radars will be useless. This infrastructure will be needed for traffic monitoring, traffic guidance, as well as spacing and separation in order to increase overall capacity and efficiency in the airspace. In this context, ground-based perception systems, capable of APNT, are critical to safe autonomous operations. This APNT relates to all moving objects in the sky, which can be cooperative (aircraft, AAM vehicles, etc.) as well as noncooperative (smaller drones, birds, etc.). The distinction between both lies in the fact that there is some form of data exchange (ADS-B or similar) with the former category in order to coordinate path planning and separation. This exchange of information is nonexistent for the latter category, which means that other sensors will be required to detect and track these objects independently from the ground. Ground-based object detection and tracking are two essential prerequisites for the purpose of traffic monitoring, and also to provide some off-board redundancy for detect and avoid (DAA). For multi-target tracking, there is an additional challenge of observation-to-track association.

Previous research has shown preliminary results for localization of a flying target in different contexts, namely:

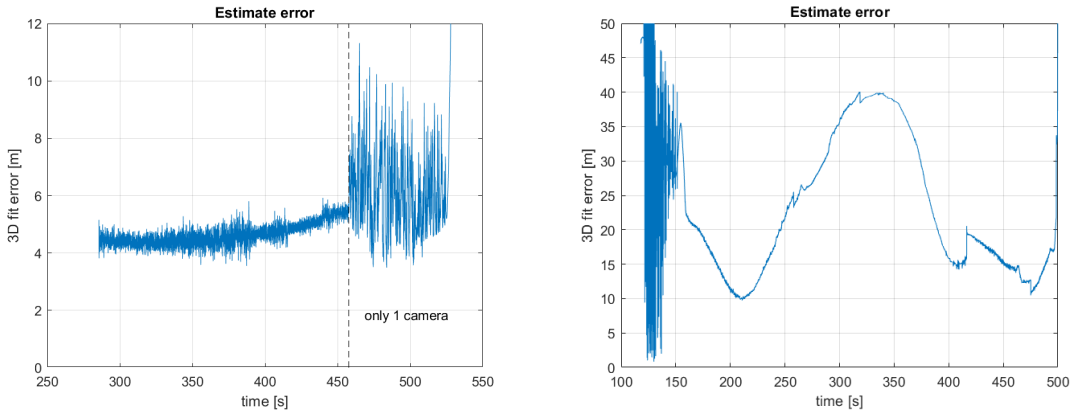
- from another flying vehicle, evaluated in a simulation environment, [2]
- from a distributed network of ground nodes (cameras and radars), evaluated in a simulation environment [3]
- from a distributed network of ground nodes (cameras or RF antennas), evaluated in simulations and field tests [4]
- considerations for optimal sensor placement (cameras only) to achieve higher precision object localization, evaluated with synthetic sensor measurements [5]

*Aerospace Research Engineer, KBR, Intelligent Systems Division, Mail Stop 269-1, AIAA Associate Fellow, email: thomas.lombaerts@nasa.gov
†Computer/GNC Engineer, NASA Ames Research Center, Intelligent Systems Division, Mail Stop 269-1, AIAA member, email: evan.kawamura@nasa.gov
‡Software Engineer, KBR, Intelligent Systems Division, Mail Stop 269-1, AIAA member, email: keerthana.kannan@nasa.gov
§Aerospace Scientist, NASA Ames Research Center, Intelligent Systems Division, Mail Stop 269-1, email: corey.a.ippolito@nasa.gov
¶Principal Scientist, KBR, Intelligent Systems Division, Mail Stop 269-1, AIAA senior member, email: vahram.stepanyan@nasa.gov.
‖Senior Research Scientist, NASA Ames Research Center, Intelligent Systems Division, Mail Stop 269-1, email: george.e.gorospe@nasa.gov
**Aerospace Engineer, NASA Langley Research Center, Aeronautics Systems Engineering Branch

This research fits in the larger project context of Perception and Distributed Sensing as elaborated in the following references. [6] elaborates on distributed sensing and advanced perception technologies to enable Advanced Air Mobility (AAM). [7] describes the simulation environment that was used to evaluate the sensor fusion approach that was developed in [3]. [8] and [9] discuss the field test set up and the sensor payload and its flight test configuration that were relevant to the work presented in [4]. [10] elaborates on the image processing of a ground based vision tracker, where its output is used as raw sensor input data by the sensor fusion approach that was discussed in [4].

Some major observations in the aforementioned research were the need for proper sensor calibration and accurate sensor boresight alignment as well as the need for optimal sensor placement with sufficient overlapping coverage as well as a well-defined geometry as defined by the ground node locations and the localized object. The ultimate goal is to achieve equivalent or better accuracy than GPS (1 to 3 meter level), which was not yet achieved in the aforementioned research. The work in this publication aims to increase these accuracy levels. In cruise, 1 to 3 meter precision levels are expected to be adequate, but higher accuracy (1m and lower) may be needed for specific phases of flight, such as takeoff and landing. Especially for higher density operations around the vertiports, where GPS signals degrade in an urban environment, APNT techniques have the potential to become enabling technologies.

Fig. 1 shows typical fitting error value levels for the simulation based evaluations (Fig. 1(a)) as well as for the field tests (Fig. 1(b)), which are respectively around 2 – 10 m for the synthetic scenarios and 10 – 40 m in real life (without compensation for lens distortion).



(a) Simulation based evaluations in [3]

(b) Field test based evaluations in [4]

Fig. 1 Typical fitting error value levels for previous research results

These error values are clearly not yet at the intended accuracy level. The trends in these error values show correlations with the number of ground nodes that are observing the target at that time instant, as well as with the topology of the ground nodes, i.e. the relative angles between the lines connecting the ground nodes with the tracked object. The latter help to distinguish between well-posed and ill-posed geometries, are related to optimal sensor placement, and are discussed in depth in [5].

Large scale networks of ground nodes also need some kind of cross validation between the different sensors to determine which measurements are most trustworthy. Various cross-validation techniques are discussed in the literature, among others based on fit and RMSE [11], leave-one-out based methods [12], and mixture of experts [13]. Sec. III elaborates in depth on this topic.

The distinction between surveillance and navigation is primarily that surveillance is ground-based and mostly independent from the localized object to account for non-cooperative objects. Navigation on the other hand is vehicle centric and should be seen in the wider context of the GNC challenges of the vehicle in a GPS-denied or degraded environment.

In this study, object tracking is achieved by means of triangulation via a Kalman Filter, similarly as in previous work [3, 4].

II. Modification of Kalman Filter to prevent overcorrection in the correction step

Since the Kalman Filter uses a gradient based optimization approach, it has the tendency to overcompensate significantly during the initial convergence phase, even when the initial condition for the estimated state is chosen in the vicinity of the starting point. Due to this overcompensating behavior, the estimate takes a long time to finally converge. Therefore, the Kalman gain matrix **W** is rescaled so that its most extreme matrix entry value doesn't exceed a certain threshold $\varepsilon_{\text{limit}}$ while preserving sign and relative differences between the matrix values so that the gradient direction is maintained. The Kalman Filter procedure was described in detail in previous publications [3, 4] and is included, with rescaling modification, in the Appendix. For this specific application, the threshold was defined as: $\varepsilon_{\text{limit}} = 6 \times 10^3$, based on closer inspection of overcompensating behavior in some scenarios.

III. Cross Validation techniques for the sensor measurements

Various cross-validation techniques are discussed in the literature, among others based on fit and RMSE [11], leave-one-out based methods [12], and mixture of experts [13]. In this context, one cross validation strategy is by making use of the residuals of the sensor measurements. These residuals are calculated by subtracting the true measurement from the synthetic measurement, which is calculated with the estimated state provided by the state estimator. In this specific case, where the state estimator is a Kalman Filter, one uses the corrected state as provided by the Kalman Filter as estimated state. Fig. 2 illustrates this concept and highlights the difference between the residual and the innovation. In contrast to the residual, as specified above, the innovation is the difference between the actual measurement and the predicted measurement.

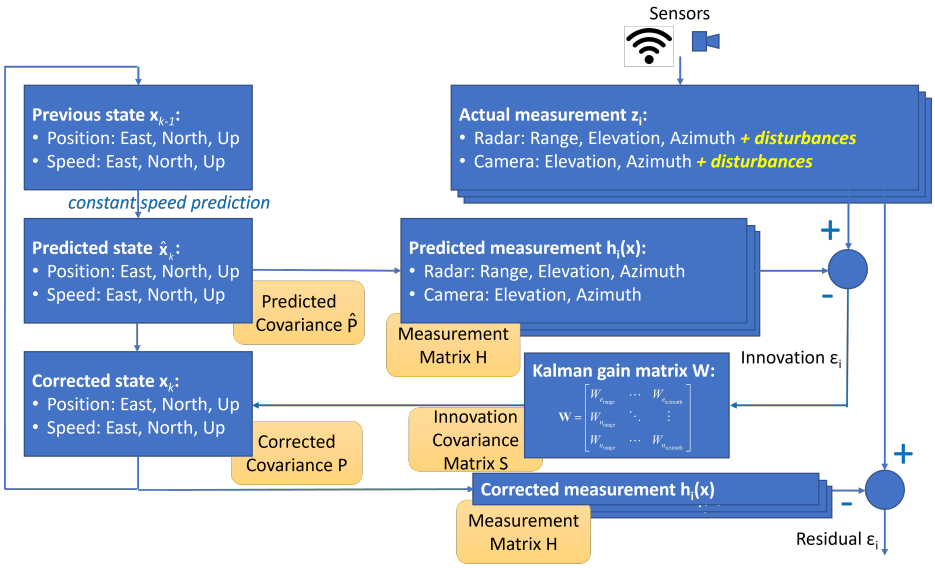


Fig. 2 Kalman Filter diagram with calculation of sensor measurement residual

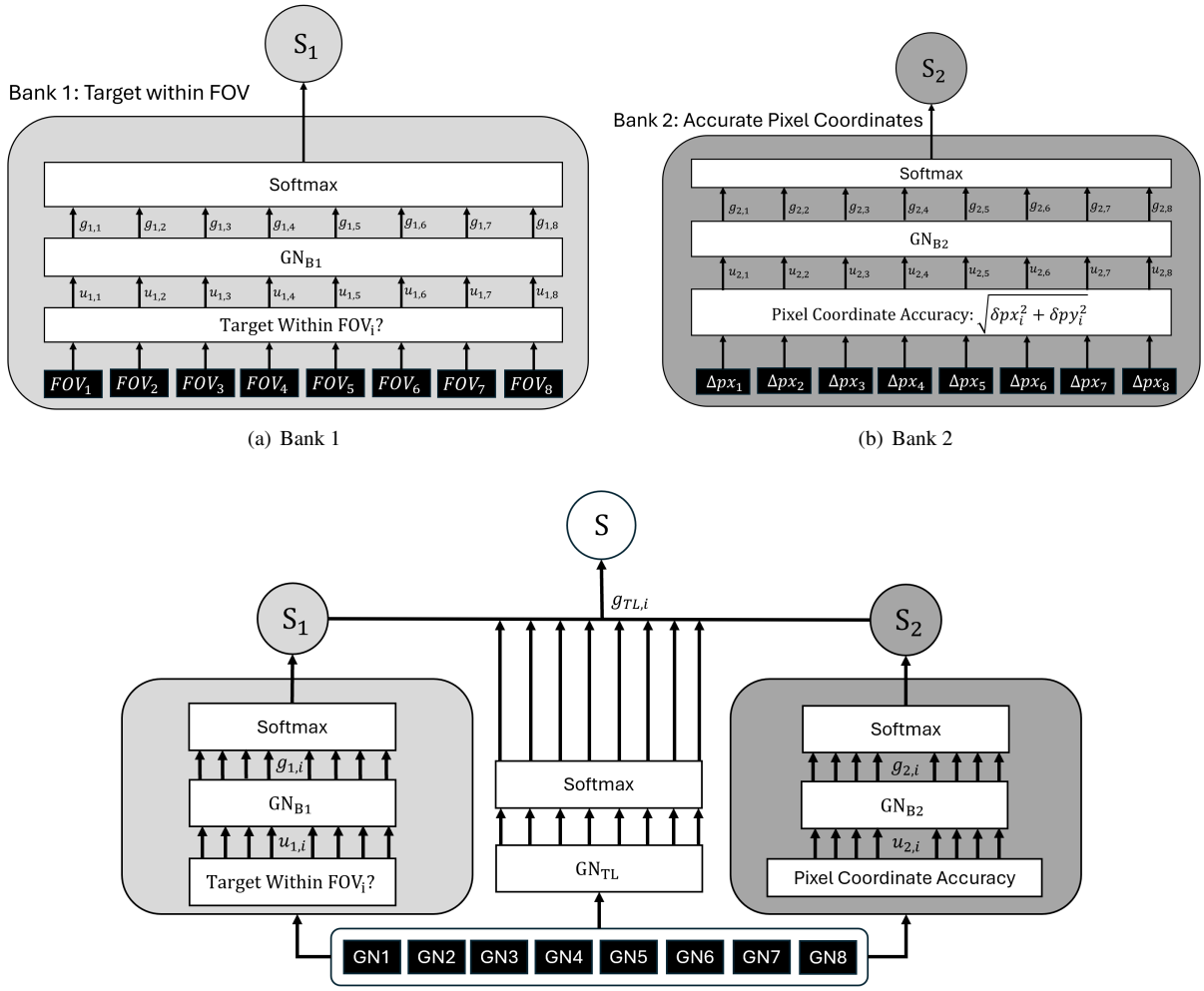
Several cross validation metrics are based on this aforementioned residual, like the RMSE as mentioned in [11] or the Mixture of Experts. Sec. III.A elaborates on the latter.

A. Hierarchical Mixture of Experts (HME) technique

There are previous studies with hierarchical mixture of experts (HME) for various other applications. One study uses HME to classify various flight phases (takeoff, climb, cruise, descend, land, and hover) for Advanced and Urban Air Mobility aircraft operating in dense, low-altitude environments, which improves situational awareness and supports safer, more efficient air traffic management compared to traditional all-or-nothing classification approaches [14]. Another study evaluates three UAV motor thrust models—two dynamic (Burgers and Staples) and one static (Gibiansky)—against flight telemetry to determine accuracy in predicting motor angular velocity by using a mixture of experts architecture to weighh each model based on performance across different maneuvers, which shows how the combined solution achieves greater accuracy than any individual model [15]. Extending the HME framework to improve state and motor angular

velocity estimation for quadcopters involves incorporating multiple thrust models and onboard acoustic measurements with extended Kalman filters and acoustic data, showing a significant enhancement in the estimation accuracy [16]. This paper extends HME for a novel application for sensor trustworthiness based on which sensors detect the target and their accuracy compared to the expected synthetic measurements of the target.

Figure 3 shows the structure of the proposed HME structure for sensor trustworthiness.



(a) Bank 1

(b) Bank 2

(c) Top level

Fig. 3 Gating network diagrams

Bank 1 (light gray) has the sensor information of which sensors see the target within their field of view, while bank 2 (dark gray) has the sensor accuracies compared to their expected synthetic measurements of the target. There are two types of gating networks to combine the sensor information. Bank 1's gating network favors the sensors that see the target and has the following form:

$$g_{1,i} = \frac{\exp(u_{1,i})}{\sum_{i=1}^{n_{\text{sensors}}} \exp(u_{1,i})} \tag{1}$$

in which $u_{1,i}$ are binary values for which sensors see the target, i.e., 1 for seeing the target and 0 for not seeing the target. Bank 2 is more complex since it favors the sensors with the highest accuracy. The first step involves obtaining the magnitude of the residuals between the actual measurement and synthetic reconstructed measurement, as was explained

previously. These residuals are defined in pixel coordinates px and py:

$$\begin{aligned} \delta px &= px_{\text{actual}} - px_{\text{synthetic}} \\ \delta py &= py_{\text{actual}} - py_{\text{synthetic}} \\ \delta pxy &= \sqrt{\delta px^2 + \delta py^2}. \end{aligned} \tag{2}$$

Then, the intermediate gates use an inverse normalization with the magnitude of the pixel coordinate residuals:

$$u_{2,i} = \frac{\sum_{i=1}^{n_{\text{sensors}}} \delta pxy_i}{\delta pxy_i} \tag{3}$$

The gating network for bank 2 resembles the formulation for bank 1, as shown earlier in Eqn. (1):

$$g_{2,i} = \frac{\exp(u_{2,i})}{\sum_{i=1}^{n_{\text{sensors}}} \exp(u_{2,i})}. \tag{4}$$

Next, the top-level (TL) intermediate gates include the gating network weights from bank 1 and 2:

$$u_{TL,i} = g_{1,i} + g_{2,i}. \tag{5}$$

Finally, the gating network weights at the top resemble the previous gating networks:

$$g_{TL,i} = \frac{\exp(u_{TL,i})}{\sum_{i=1}^{n_{\text{sensors}}} \exp(u_{TL,i})}. \tag{6}$$

These top level gating network weights provide time-varying trustworthiness or reliability metrics for each individual sensor in the network of ground nodes. This will be illustrated in an application scenario later in this study.

IV. Sensor Characteristics, Measurement Processing and Data Calibration

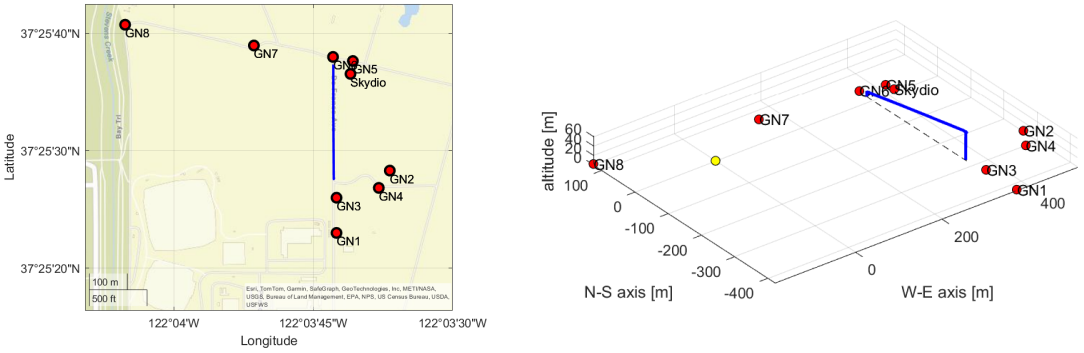
Only one type of sensor is considered in this study, namely electro-optical cameras. Radars, wireless RF equipment and distance measuring equipment (DME) were considered in previous work [3, 4] and can be modeled in the Kalman Filter observation models, but were not incorporated for this study. Cameras provide azimuth and elevation only and have specific sensor characteristics, such as limitations on field of view and useful range, noise, latency, update rate and resolution. Some characteristic values from the literature of state-of-the-art sensor hardware [17] are given in Table 1. [9] provides more details on sensor hardware. Specific details of the sensor orientations and measurement readings for the specific scenario considered in this study are given in Sec. V for the field tests.

Table 1 Sensor characteristics

sensor	EO camera
sensed variables	azimuth [deg]
	elevation [deg]
model / type	GoPro
(hardware) [9]	Hero 10
limits on field of view	az: 92deg
(16/9)	el: 60.44deg
max useful range	700m
min useful range	0m
noise sigma	$9.1 \cdot 10^{-3}$deg
latency	0.1s
update rate	30fps
resolution	$1.3 \cdot 10^{-3}$deg

V. Application scenario

The main scenario in this publication is at Ames Research Center, between the gantry of the Outdoor Aerodynamic Research Facility (OARF) and the Ames's Disaster Assistance and Rescue Team (DART) training area, with strategically placed cameras throughout the area, as shown in Fig. 4. It is a similar set up as discussed in [4], but for a shorter flight segment with a sloped angle, representing a descent or climb depending on the direction flown, and with a larger number of ground nodes. This scenario also includes one airborne observer. More details about the design and implementation of the scenarios are given in [8]. The current study discussed in this paper is using both synthetically generated ground node measurement data as well as the real life ones. Comparing both allows to analyze the impact of real life disturbances which are not yet modeled in the sensor model characteristics as discussed in Sec. IV. Raw data image processing is a crucial aspect but is not discussed here–the procedure is explained in [10].



(a) Map view of the field test set up

(b) 3D view of the field test set up

Fig. 4 Map and 3D view of the set up of the scenario with location of observers and trajectory between OARF and DART.

Each ground node as well as the airborne observer are equipped with a camera that is facing in a specific direction horizontally as well as vertically, and tracking the moving object flying along the track, in this specific case an Alta X multicopter drone, shown in Fig. 5(a). The airborne observer is a Skydio drone, shown in Fig. 5(b).



(a) Multicopter drone Alta X, used as tracked vehicle

(b) Skydio drone, used as airborne observer

Fig. 5 Pictures of multicopter drone and airborne observer drone Skydio used for the test

Tab. 2 shows accurate location and viewing angle and range information for each individual ground node. Experiments have shown that the accuracy of the estimates depends to a large extent on the precision of this ground node information.

The map view of the vehicle path with observation cones per sensor are shown in Fig. 6. The measured pixel coordinates as seen per ground node are shown in Fig. 7. These are converted to azimuth and elevation angles over time per ground node. These angles are shown in Fig. 8. Fig. 8(a) are the synthetic measurements which reflect the sensor characteristics which were discussed in Sec. IV. Fig. 8(b) are the true measurements from the field test, as provided by

Table 2 Distributed sensors locations and angles around DART site for field test

Observer	Lat (°)	Long (°)	Abs. Alt. (m)	Heading (°)	Elev. (°)	Range (m)
GN 1	37.4230556	-122.061803	1.27	-2.6373	12.0529	600
GN 2	37.4245310	-122.060209	1.51	-73.6693	1.7465	600
GN 3	37.4238889	-122.061803	0.83	-3.1361	1.6263	600
GN 4	37.4241220	-122.060538	1.51	-49.7013	17.9006	600
GN 5	37.4271260	-122.061314	1.52	-123.2481	24.9353	600
GN 6	37.4272230	-122.061907	1.75	174.7963	27.6693	600
GN 7	37.4274930	-122.064273	1.54	115.3256	24.0814	600
GN 8	37.4279850	-122.068125	1.71	105.0186	-0.1237	600
Skydio	37.4268210	-122.061385	10.00	242.5	0.00	600

the image processing routine. Both look very similar, but there are subtle differences. These are caused by, among others, a slightly different field of view, lens distortion, and other real-world effects that are impossible to model in the synthetic measurements.

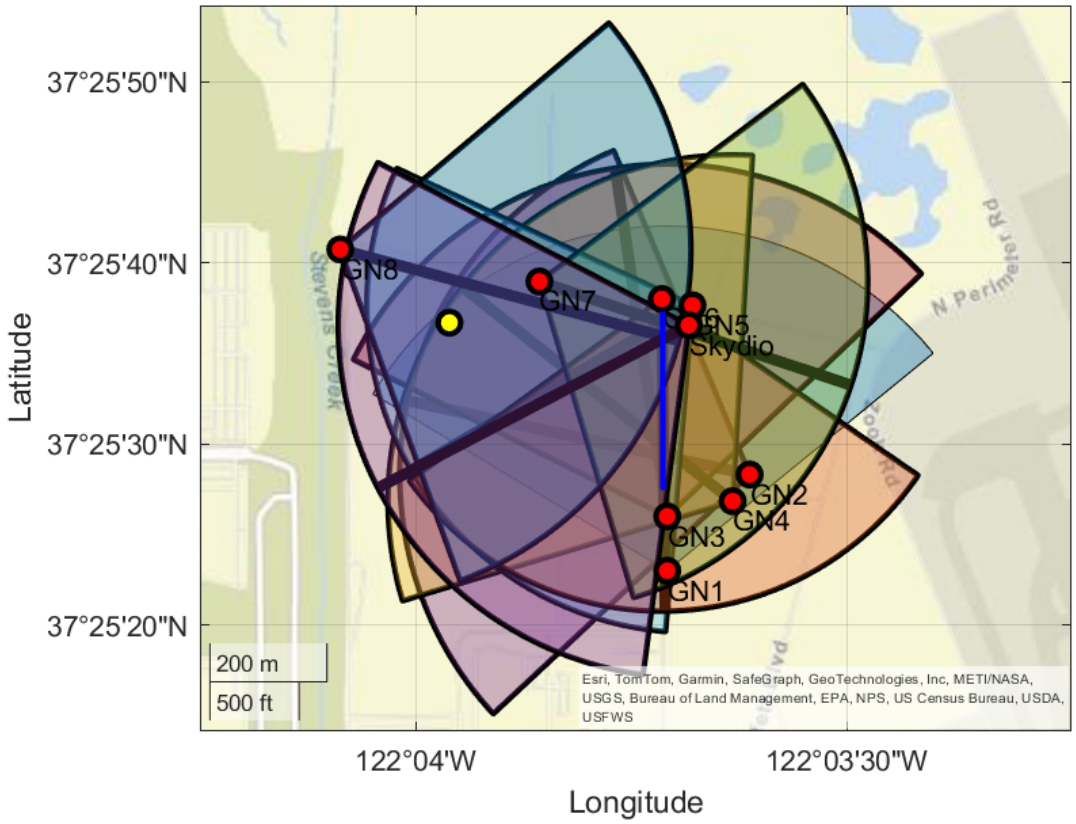


Fig. 6 Map view of the field test set up with sensor cones

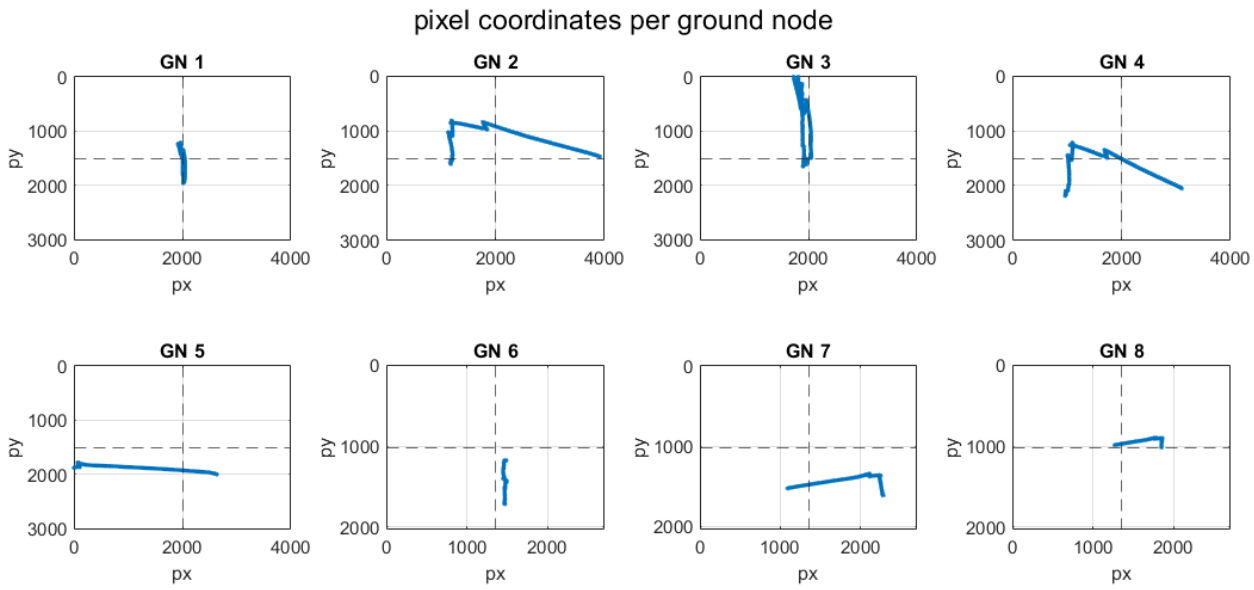
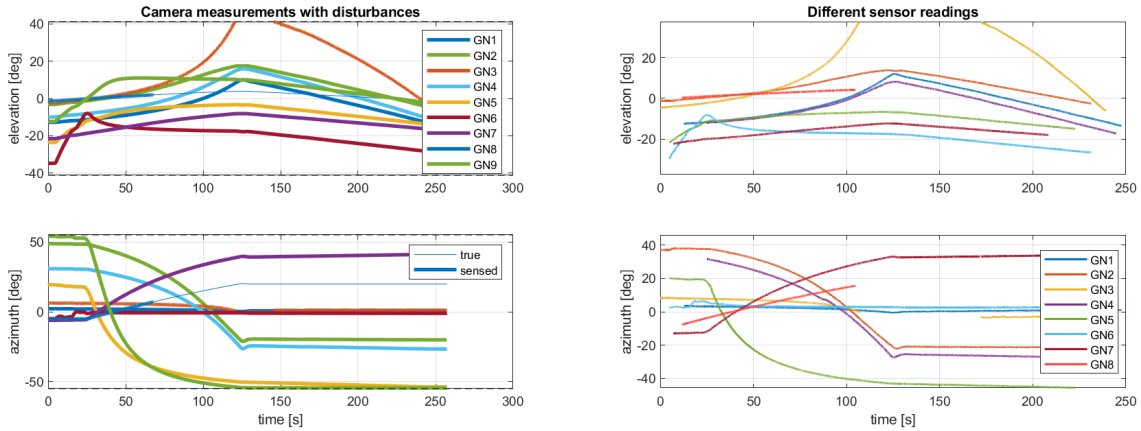


Fig. 7 2D pixel traces per GN



(a) Synthetic measurements

(b) Field test measurements

Fig. 8 Synthetic and field test measurements

VI. Results

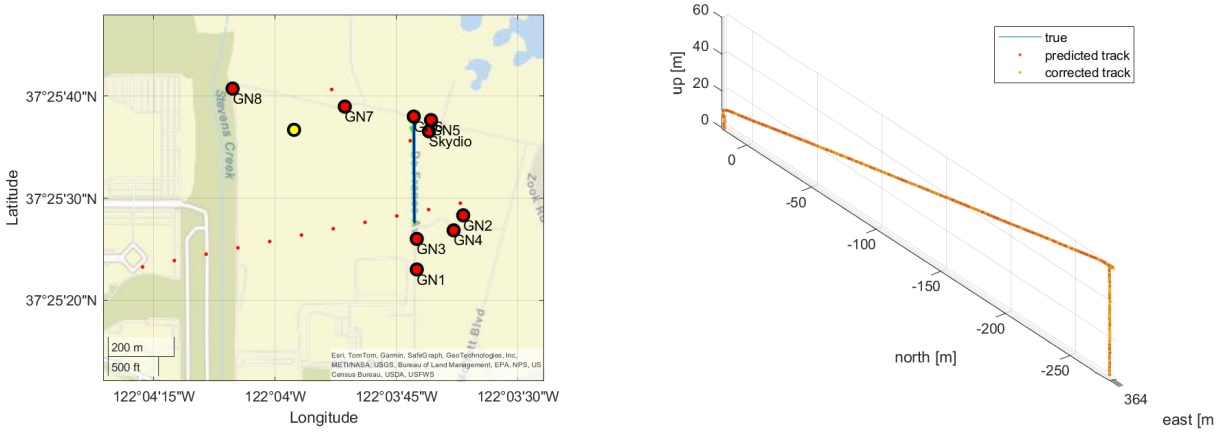
The results section consists of two major parts. First the Kalman filter results for the application scenario are discussed in Sec. VI.A. The cross-validation evaluation work is covered in Sec. VI.B.

A. Kalman Filter results for application scenario

In this section, synthetic results are first discussed in Sec. VI.A.1 and thereafter the results based on the field test measurements in Sec. VI.A.2. Finally, both are compared.

1. Synthetic measurements by 8 ground nodes and an airborne observer

Fig. 9 compares the estimated path and the true path. Fig. 9(a) shows the map view with all the GN locations as well as the aerial observer. The estimated locations are color coded red for low fidelity (high condition number for the observability matrix, greater than 10^{10}) and green for high fidelity (condition number smaller than 10^{10}). The corresponding values of the condition number over time are shown in Fig. 11(b). Fig. 9(b) shows the estimated 3D path in the East-North-Up reference frame, comparing the true path with predicted and corrected estimates from the KF. All results appear fairly accurate on this scale throughout the maneuver.



(a) Map view of the estimated track with 8GNs and the airborne observer

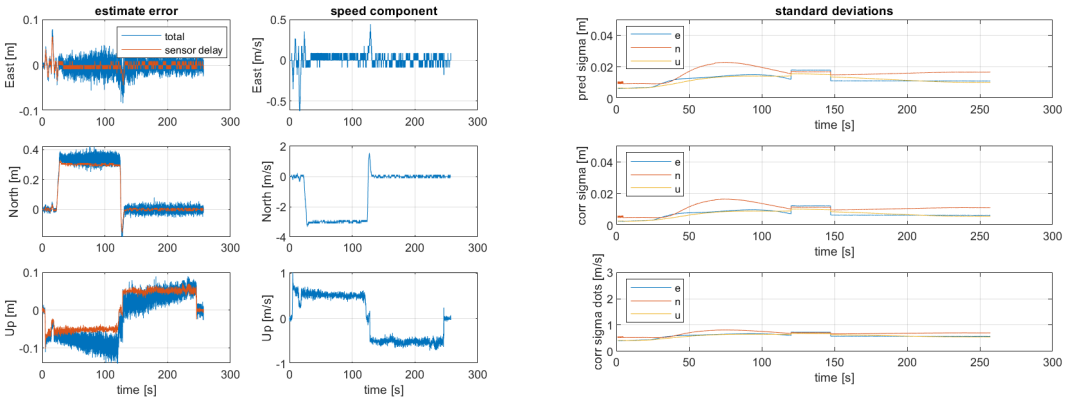
(b) 3D ENU view comparing estimated and true path

Fig. 9 Map and 3D view comparing estimate track and true path

The East-North-Up (ENU) errors as components per axis direction are shown in the left column of Fig. 10(a). The East component of the estimate fit error is dominated by sensor noise, but this does not apply for the North and Up components. Here, the sensor time delay causes a small but significant fit error. The figures in the right column of Fig. 10(a) show the ENU velocity components, and the red lines in the left column show the values of these velocity components multiplied by the time delay. This shows that the largest share of the fit error is caused by the sensor time delay. The magnitude and direction of fit errors North and Up are consistent with a 3 deg climbing flight up to 120s and a vertically descending flight segment afterwards. Fig. 10(b) shows the standard deviations of the estimates per axis and confirms the accuracy of the results.

The full rank and the low value of the condition number of the observability matrix, both shown in Fig. 11, confirm that the estimation problem is well posed in this scenario. In general, these two metrics of rank and condition number of the observability matrix show how well the position and speed components of the tracked object (KF states) can be inferred from knowledge of the sensor measurements (KF measurements).

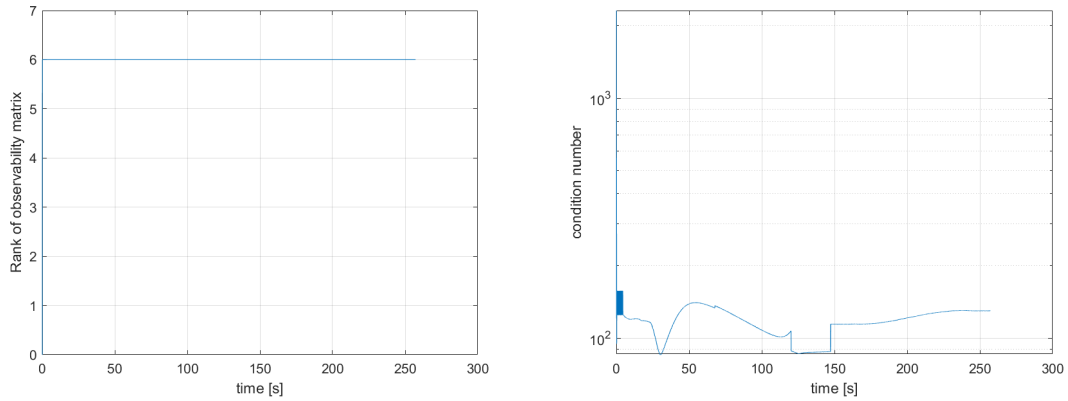
The 3D fit error over time between estimate and true path in Fig. 12(a) shows that the average error is around 0.35m in the higher speed 3 deg climbing flight segment and between 0.05 and 0.1m in the vertically descending segment, well within the required precision of 1m to 3m. Finally, Fig. 12(b) compares the actual camera disturbances on elevation and azimuth in the left column with the fit error induced deviations in the reconstructed sensed variables in the right column. Comparing the left and right columns in Fig. 12(b) shows the beneficial effect of the KF in removing sensor characteristics.



(a) ENU estimate errors per axis

(b) ENU standard deviations of estimate per axis

Fig. 10 ENU errors and standard deviations per axis



(a) Rank of the observability matrix

(b) Condition number of the observability matrix

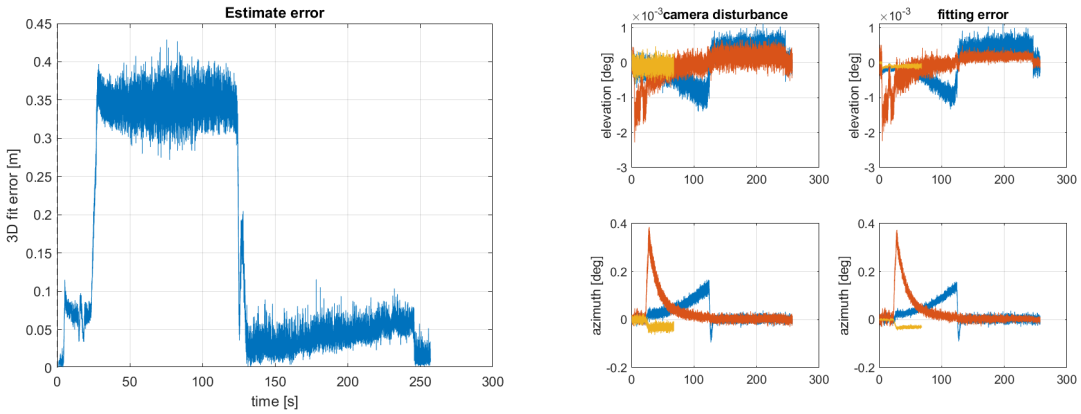
Fig. 11 Rank and condition number of the observability matrix

2. Real life measurements by 8 ground nodes only

This section follows the same outline as the previous one, but this time all results are based on the real life measurements provided by the 8 ground nodes only. The aerial observer was not included due to the lack of sufficiently accurate position and attitude as well as camera boresight information from the drone.

Fig. 13 compares the estimated path and the true path. Fig. 13(a) shows a satellite view with all the GN locations. The estimated locations are color coded green for high fidelity (condition number smaller than 10^{10}). The corresponding values of the condition number over time are shown in Fig. 15. By assuming that the takeoff location is known, the time for convergence of the estimate was reduced from 27s to only 10s. Fig. 13(b) shows the estimated 3D path in the East-North-Up reference frame, comparing the true path with predicted and corrected estimates from the KF. All results appear fairly accurate on this scale for the first part of the maneuver. Towards the top of the flight path, the target leaves the observation cones of two ground nodes, which happens shortly after 100s as shown in Fig. 16. This reduction in the number of measurement data points results in a deterioration of precision of the estimated position. Overall, the sudden jumps in the estimate occur every time when there is one fewer ground node tracking the target.

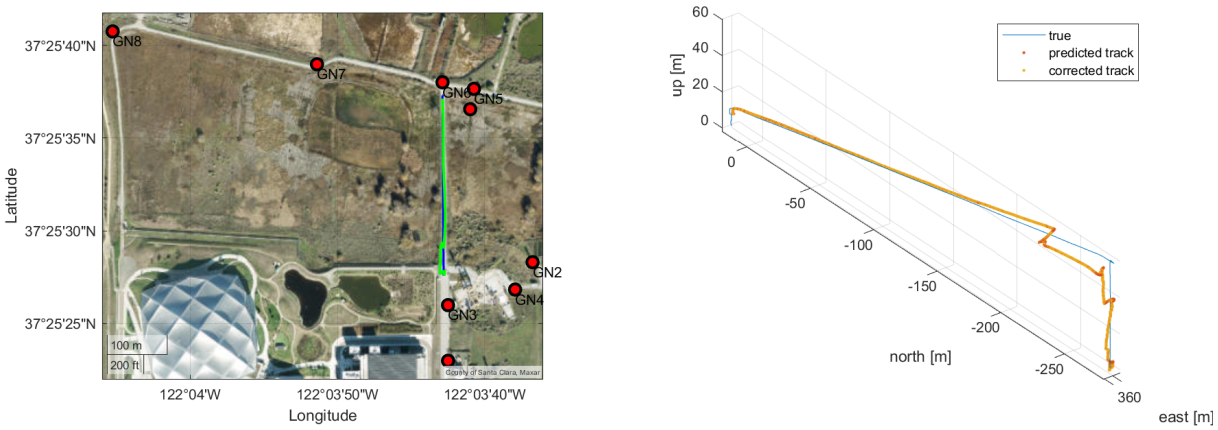
The East-North-Up (ENU) errors as components per axis direction are shown in Fig. 14(a), together with the 2 and 3 σ standard deviations. These results are consistent with the fitting errors staying within 2 σ bounds for around 95% of



(a) 3D ENU fit error

(b) Comparison between sensor disturbance and fit errors per axis

Fig. 12 3D ENU fit error and comparison between sensor disturbance and fit errors



(a) Satellite view of the estimated track with 8GNs

(b) 3D ENU view comparing estimated and true path

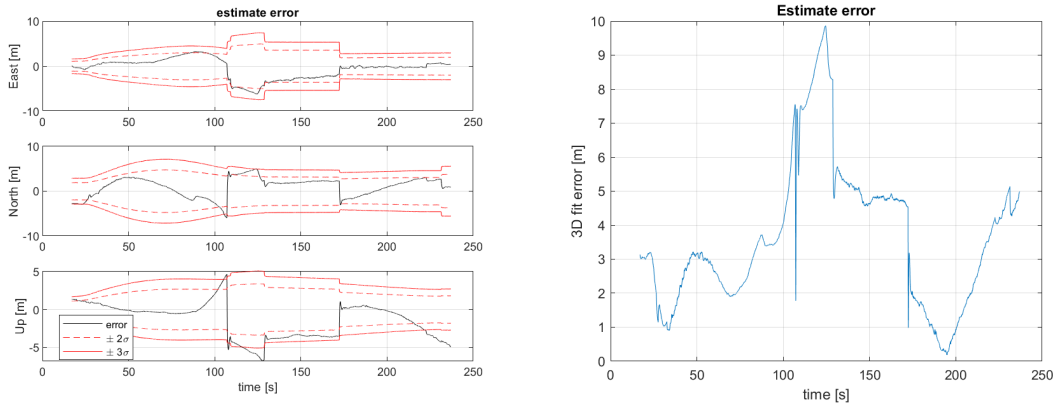
Fig. 13 Satellite view and 3D view comparing estimate track and true path

the time and about 99% of the time within the 3 σ bounds. Due to the various real life effects that are not modeled in the synthetic measurement data, these fit errors are significantly larger compared to the synthetic counterpart in Sec. VI.A.1. However, the 3D fit error in Fig. 14(b) shows that the 3D fit error does not exceed 10m and is around 3.2m on average throughout the run. This is very close to the GPS precision threshold. The largest error occurs halfway between takeoff and landing, where the distance from the GNs is the longest.

The full rank and the low value of the condition number of the observability matrix, both shown in Fig. 15, confirm that the estimation problem is well posed in this scenario, as was the case in the previous section. Except for the very end, the condition number value between synthetic and real life field test is very similar, around 100.

Fig. 16 shows the number of sensors that have the target in sight at each time instant during the run. It is the reduction in coverage towards the end that causes the deteriorating rank and condition number of the observability matrix.

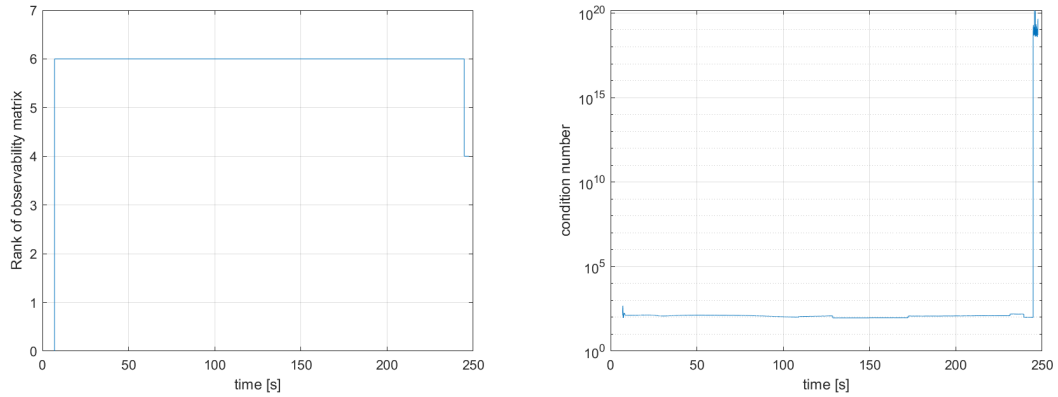
Overall, comparing Fig. 12(a) and 14(b) shows that real world effects that are not modeled by the sensor characteristics incorporated in the synthetic measurements result in a significantly larger fit error. However, the real life accuracy is still significantly better compared to previous work in [4], of which the fit error was shown in Fig. 1(b), largely thanks to higher precision values for the viewing angles of the ground nodes, a larger number of ground nodes, and various other lessons learned in the first field test.



(a) ENU estimate errors and standard deviations per axis

(b) 3D fit error

Fig. 14 ENU errors and standard deviations per axis and 3D fit error



(a) Rank of the observability matrix

(b) Condition number of the observability matrix

Fig. 15 Rank and condition number of the observability matrix

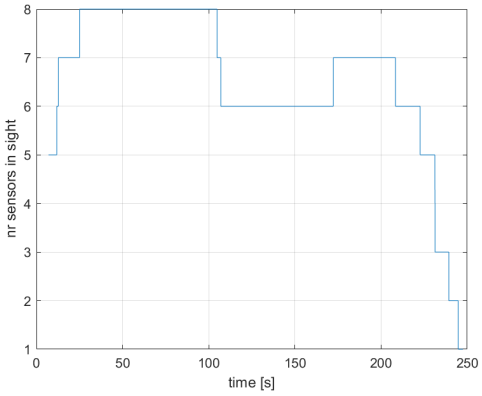


Fig. 16 Number of sensors in sight

B. Cross validation results of sensor measurements

For the cross validation analysis of the sensor measurements, one can create 'expected' sensor measurements and compare them to the true ones. Fig. 17 shows both expressed in pixel coordinates for all the ground nodes.

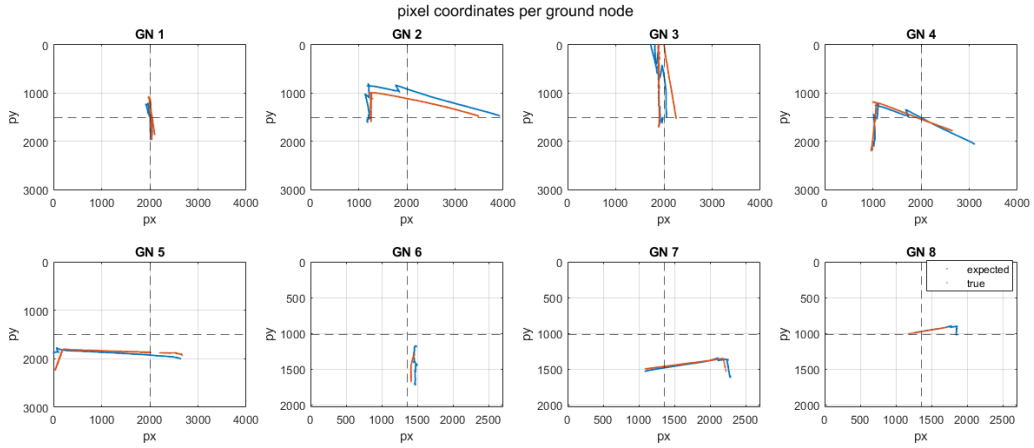


Fig. 17 2D pixel traces per GN

Calculating the difference between both traces shown in Fig. 17 results in the previously defined sensor measurement residuals, which are shown in Fig. 18. Larger residuals indicate a larger measurement error for that specific sensor. Sensors with smaller residuals are more reliable for the sensor fusion task. Fig. 18 also shows that reliability is time dependent. The more reliable ground nodes differ between the various phases of flight. This depends on the topology, distance from target, etc.

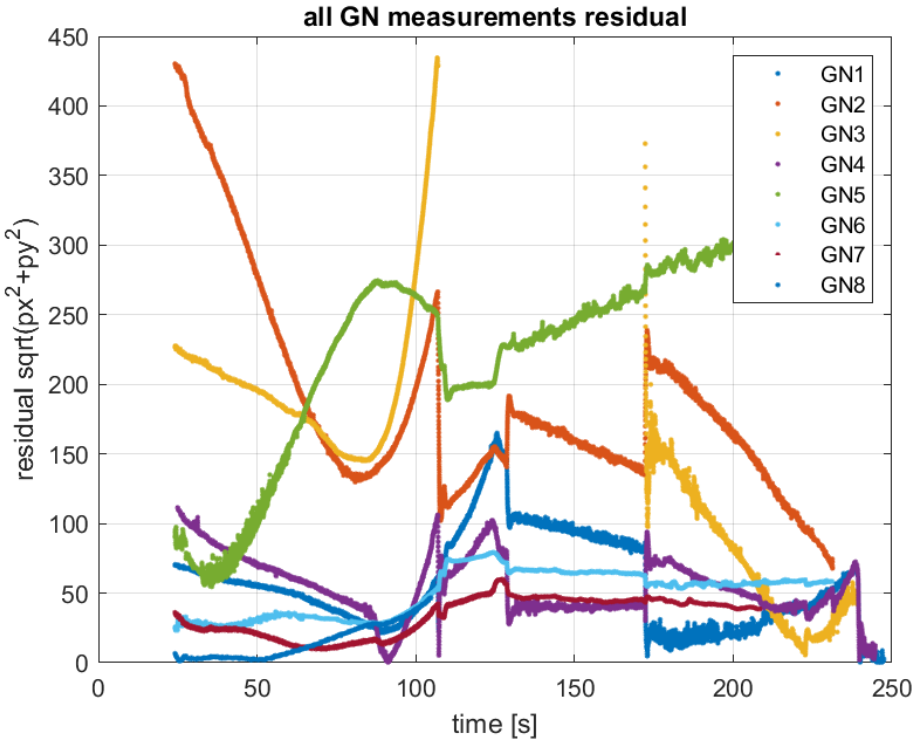
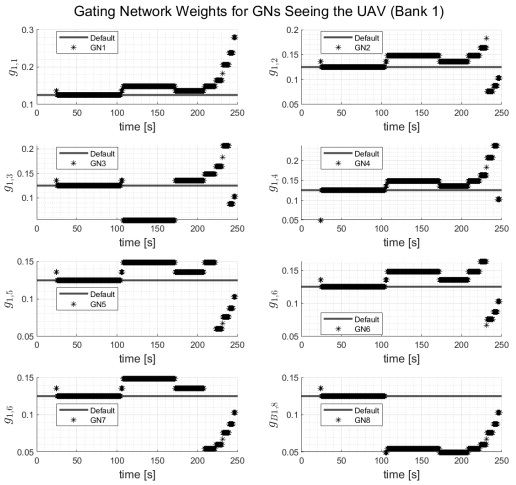
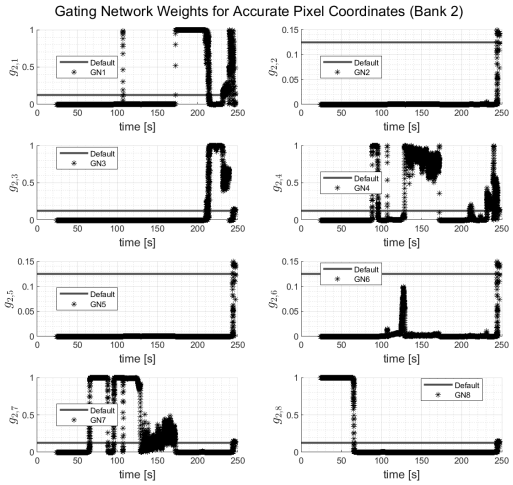


Fig. 18 Sensor measurement residuals

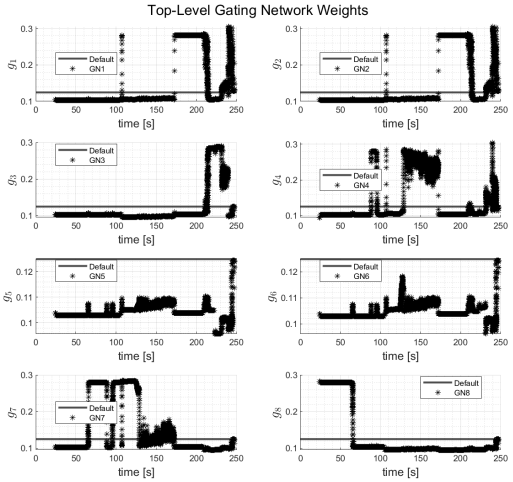
The previously explained Hierarchical Mixture of Experts (HME) technique (Sec. III.A) was applied on these sensor measurement residuals and the results are shown in Fig. 19. The subfigures distinguish between the different banks, namely Fig. 19(a) shows the results for bank 1 which focuses on if the target is in sight for the respective GN. Bank 2 considers the residual values and these weightings are shown in Fig. 19(b). Finally, Fig. 19(c) show the combined weightings. Looking at the combined weightings, the most reliable GN varies over time and is GN8 up to 65s, then it is replaced by GN7 up to 85s, GN4 until 90s, again GN7 until 130s, GN4 up to 170s, GN1 up to 210s, GN3 as far as 240s into the run, and finally GN1 until the end. These observations are confirmed by Fig. 18. It is important to know that one GN is insufficient for triangulation of the target's location. Combining measurements from at least 4 of them results in higher precision of the estimates. In general, GNs at shorter distances have smaller residuals.



(a) Bank 1



(b) Bank 2



(c) Top level

Fig. 19 Gating network weights

VII. Observations, Conclusions and Future Work

This section focuses on some general observations and conclusions that can be drawn after the aforementioned case studies and analyses of variations in independent variables.

The required accuracy of object localization for Urban Air Mobility with APNT techniques is between 1m and 3m if one aims to achieve similar precision levels as traditional GPS methods. In cruise, these precision levels are expected to be adequate, but higher accuracy (1m and lower) may be needed for specific phases of flight, such as takeoff and landing. Especially for higher density operations around the vertiports, where GPS signals degrade in an urban environment, APNT techniques have the potential to become enabling technologies.

Based on lessons learned from previous experiments, an average fitting error of around 3m was achieved in this particular case study. More improvements can still be achieved by focusing on the most important factors that contribute to the fitting errors.

Based on the aforementioned case studies, the most important factors that influence the precision of the localization results are the following:

- **Precision of the location and alignment of the ground nodes:** In WGS-84 coordinates, latitude and longitude coordinate values should be defined down to tenths of arcseconds, vertical precision down to centimeter level, and the viewing angles need a precision down to less than a tenth of a degree. Table 2 shows the values and the corresponding precision levels that were used in this study. Ref. [18] shows the potential of bundle adjustment for this purpose.
- **Field of view of the sensors:** Aligning the viewing angles of the sensors to maximize coverage, while taking into account their field of view limitations, is also important. This can be mitigated by increasing the number of GNs so that the relative impact of the loss of one GN is not as significant.
- **Sensor noise:** The sensor noise levels based on state of the art sensor characteristics (specified in Table 1) and observed noise levels after image processing of field test imaging do not prevent achieving the aforementioned precision levels. An important observation here is that images at closer ranges tend to become more noisy because the image processing software spreads the target detection over more available pixels. More advanced target detection techniques have the potential to bring benefits here.
- **Sensor delay:** Typical sensor delay values based on state of the art sensor characteristics (specified in Table 1) have a more significant impact on precision levels than noise, but do not prevent achieving the aforementioned precision levels. However, image processing and network time delays are not yet taken into account in these numbers. Ref. [19] investigates the impact of communication delays in the context of distributed sensing. The impact on the precision is always a combination of the total time delay and the speed of the vehicle. Higher accuracy positioning in takeoff and landing areas will take place at lower speeds and shorter distances compared to cruise flight conditions, which will help improving the precision levels.
- **Density and positioning of the ground nodes and airborne observers:** Distances between ground nodes and airborne observers on one hand and the tracked object on the other hand, together with the intersection angles of the observation vectors have a major impact on the precision levels, as previously shown in [5].

As a final remark, another important aspect is cross-validation of the sensor measurements, like for example by means of hierarchical mixture of experts. These provide time varying weightings for each ground node as an indication of the reliability of each individual ground node. In future work, information about unreliable ground nodes could be used in the Kalman Filter to avoid their measurements for calculating the estimated location of the target.

Appendix: Kalman Filter based Object Tracker

The object tracker is a further development which is based on a conventional Kalman Filter as elaborated in detail in Ref. [2–4, 20]. This filter estimates 3D position and velocity components in ENU coordinates (East–North–Up) and assumes constant speed in the prediction step, and updates the estimate in the correction step based on the available measurements. The filter gain calculation is modified in order to incorporate valid flags for the different sensor measurements, which makes this method adaptive and robust against invalid data, loss of signal between the different distributed sensors, outliers and lower sampling rates. The filter gain is also rescaled to prevent overcompensation in this new implementation. This modular architecture is fairly straightforward to extend so that other sensors and sensor locations can be incorporated as well, such as radar and others. Fig. 20 illustrates the general overview of how the Kalman Filter is set up, including the possibility to add radar. The prediction step assumes constant velocity for calculating the predicted state $\hat{\mathbf{x}}_k$ based on the previous state $\mathbf{x}_{k-1}$. The state consists of six elements, namely the three position coordinates and the three velocity coordinates, all in ENU with respect to a predefined fixed referenced point.

Predicted measurements $\hat{\mathbf{z}}_{k,i}$ are calculated for each sensor location i, which are a function of the predicted state $\mathbf{h}_i\ (\hat{\mathbf{x}}_k)$ and the sensor location coordinates. These predicted measurements consist of the measured variables that are provided by the relevant sensors for each location, namely range, azimuth and elevation from the radar (when available) as well as azimuth and elevation from the EO camera. By comparing the actual measurement $\mathbf{z}_{k,i}$ with the predicted measurement $\hat{\mathbf{z}}_{k,i}$ for each sensor location i, one can calculate the innovation $\varepsilon_i = \mathbf{z}_{k,i} - \hat{\mathbf{z}}_{k,i} = \mathbf{z}_{k,i} - \mathbf{h}_i\ (\hat{\mathbf{x}}_k)$. Main drivers for the innovation are errors in the predicted state (due to the constant speed assumption) and sensor disturbances. Next, the information from all individual sensor locations and sensors is stacked together in one large innovation array ε. As last but one, the Kalman gain matrix $\mathbf{W}$ is calculated by means of the predicted covariance $\hat{\mathbf{P}}_k$, measurement matrix $\mathbf{H} = \frac{d\mathbf{h}(\mathbf{x})}{d\mathbf{x}}$, and the innovation covariance matrix $\mathbf{S}$. This Kalman gain matrix $\mathbf{W}$ is weighted with valid flags to deal appropriately with the sensor measurements as explained previously. The corrected state $\mathbf{x}_k$ is calculated by means of predicted state $\hat{\mathbf{x}}_k$, the innovation ε and the Kalman gain matrix $\mathbf{W}$ via the following equation: $\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{W} \cdot \varepsilon$. Finally, the corrected covariance $\mathbf{P}_k$ is calculated. The complete procedure is described in more detail below.

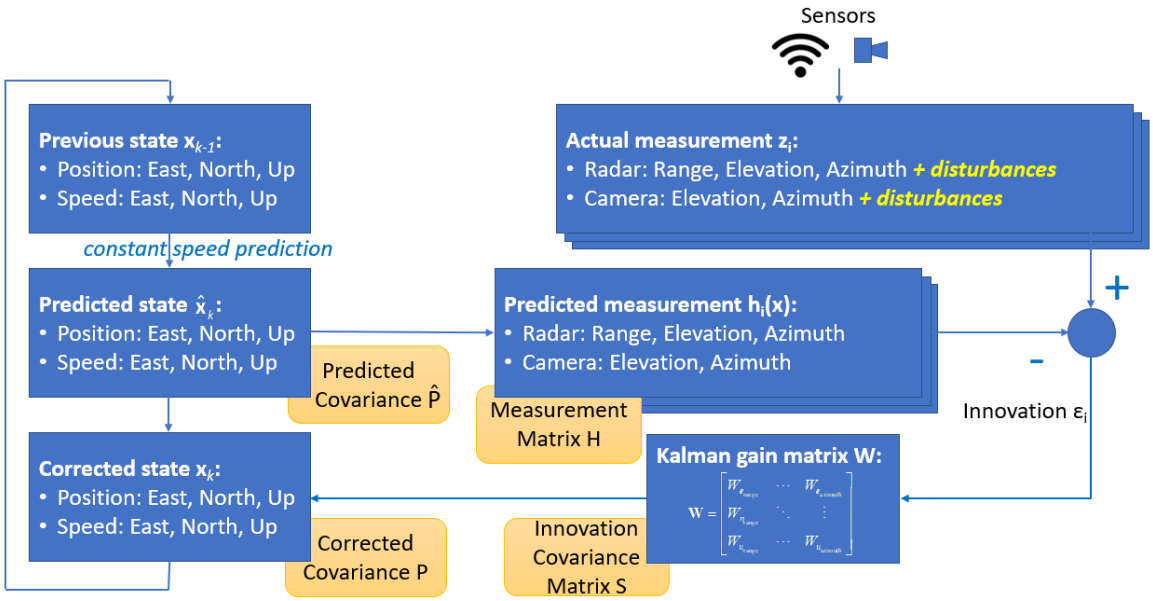


Fig. 20 General overview of the Distributed Sensor Kalman Filter setup

A. State Prediction $\hat{\mathbf{x}}_k$

The one step ahead prediction of the state vector is split up as follows:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{k_1} \\ \hat{\mathbf{x}}_{k_2} \end{bmatrix} \tag{7}$$

With the position in the upper half of the vector:

$$\hat{\mathbf{x}}_{k_1} = \begin{bmatrix} \hat{E} \\ N \\ U \end{bmatrix}_k = \begin{bmatrix} E \\ N \\ U \end{bmatrix}_{k-1} + \Delta t \cdot \begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_{0_{k-1}} \tag{8}$$

And the velocity in the lower half of the vector:

$$\hat{\mathbf{x}}_{k_2} = \begin{bmatrix} \hat{\dot{E}} \\ N \\ U \end{bmatrix}_k = \begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_{0_{k-1}} \tag{9}$$

Note that the coordinates are East–North–Up with respect to a predefined fixed reference point and that the velocity is assumed to be close to constant. Moreover:

$$\begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_{0_{k-1}} = \mathbf{L}^{\top} \cdot \begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_{T_{k-1}} \tag{10}$$

where $\mathbf{L}^{\top}$ is the rotation matrix from the target ENU-frame to the reference point ENU-frame, as elaborated in the Appendix. Next step is to rotate from ENU to ECEF reference frame:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{0_{\text{ECEF}}} = \mathbf{T} \begin{bmatrix} E \\ N \\ U \end{bmatrix}_{0_{k-1}} \tag{11}$$

where $\mathbf{T}$ is the rotation matrix from the ENU reference frame to the ECEF reference frame, as defined in the Appendix. The predicted target position is converted straight from ECEF to WGS-84 (i.e. latitude, longitude and altitude) via the transformation elaborated in the next appendix, resulting in the predicted target position coordinates $\begin{bmatrix} \phi_t & \lambda_t & h_t \end{bmatrix}^{\top}_{\text{pred}}$.

B. State Error Covariance Prediction $\hat{\mathbf{P}}_k$

$$\hat{\mathbf{P}}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^{\top} + \mathbf{Q} \tag{12}$$

with:

$$\mathbf{F} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ \hline & & & 1 & 0 & 0 \\ & \mathbf{0}_{3\times 3} & & 0 & 1 & 0 \\ & & & 0 & 0 & 1 \end{array}\right] \qquad \mathbf{Q} = \mathbf{G}\boldsymbol{\Gamma}\mathbf{G}^{\top} \tag{13}$$

and:

$$\mathbf{G} = \left[\begin{array}{ccc} 0.5\Delta t^2 & 0 & 0 \\ 0 & 0.5\Delta t^2 & 0 \\ 0 & 0 & 0.5\Delta t^2 \\ \hline \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{array}\right] \qquad \boldsymbol{\Gamma} = \begin{bmatrix} \sigma_e^2 & 0 & 0 \\ 0 & \sigma_n^2 & 0 \\ 0 & 0 & \sigma_u^2 \end{bmatrix} \tag{14}$$

C. Predicted Measurement Calculation $\hat{\mathbf{z}}_k$

The total predicted measurement vector $\hat{\mathbf{z}}_k$ consists of the predicted measurement vectors $\hat{\mathbf{z}}_{k,i}$ of each individual sensor location i. The predicted measurement vectors contain the relevant equipment-dependent information per individual sensor location and include only EO camera information in this specific study: elevation α and azimuth γ (no range due to the lack of stereo view and depth perception). The predicted measurement vector per sensor location is described as follows with only the hardware-relevant parts present in order to reduce vector size and computational load:

$$\hat{\mathbf{z}}_{k,i} = \hat{\mathbf{z}}_{k,i_{\text{EOcam}}} \tag{15}$$

$$\hat{\mathbf{z}}_{k,i_{\text{EOcam}}} = \begin{bmatrix} \hat{\alpha}_i & \hat{\gamma}_i \end{bmatrix}^{\top} \tag{16}$$

The individual measured variables are defined as follows:

$$\hat{\alpha}_i = \arctan \frac{-\hat{Z}_{S_i}}{\hat{\rho}_{i_{\text{hor}}}} \tag{17}$$

$$\hat{\gamma}_i = \arctan \frac{\hat{Y}_{S_i}}{\hat{X}_{S_i}} \tag{18}$$

where:

$$\hat{\rho}_{i_{\text{hor}}} = \sqrt{\hat{X}_{S_i}^2 + \hat{Y}_{S_i}^2} \tag{19}$$

and:

$$\begin{bmatrix} \hat{X_{S_i}} \\ Y_{S_i} \\ Z_{S_i} \end{bmatrix} = \mathbf{M}_i^\top \cdot \begin{bmatrix} \hat{E_i} \\ N_i \\ U_i \end{bmatrix}_k \tag{20}$$

where $\mathbf{M}_i^\top$ is the rotation matrix from the i^{th} sensor-based ENU reference frame to the sensor fixed reference frame of the i^{th} sensor, as defined in the Appendix. It rotates the East-North-Up coordinates for each sensor location towards the sensor orientation (viewing direction of camera (and radar if applicable)). These local ENU_i coordinates are calculated from the aforementioned predicted state in WGS-84 coordinates $\begin{bmatrix} \phi_t & \lambda_t & h_t \end{bmatrix}_{\text{pred}}^\top$ via the transformation elaborated in the appendix, resulting in the predicted target position coordinates in ECEF, and a subsequent rotation from ECEF to the local sensor's ENU as follows:

$$\begin{bmatrix} \hat{E_i} \\ N_i \\ U_i \end{bmatrix}_k = \mathbf{T}_i^\top \begin{bmatrix} \hat{X} \\ Y \\ Z \end{bmatrix}_{0_{\text{ECEF}}} \tag{21}$$

where $\mathbf{T}_i^\top$ is the rotation matrix from the ECEF reference frame to the local sensor ENU reference frame, as defined in the Appendix.

D. Kalman Filter Gain W

The following steps are performed in order to calculate the gain of the filter:

1) Calculate measurement matrix **H**
2) Calculate innovation covariance matrix **S**
3) Calculate gain matrix **W**

Each step will be elaborated in detail next:

1. Measurement matrix **H**

The measurement matrix **H** is basically the Jacobian of the predicted measurement vector:

$$\hat{\mathbf{z}}_k = \begin{bmatrix} \hat{\mathbf{z}}_{k,1} & \dots & \hat{\mathbf{z}}_{k,i} & \dots & \hat{\mathbf{z}}_{k,N} \end{bmatrix}^\top$$

with N the number of sensor locations and each individual sensor location based component defined as follows with only the hardware-relevant parts present in order to reduce matrix size and computational load:

$$\hat{\mathbf{z}}_{k,i} = \hat{\mathbf{z}}_{k,i_{\text{EOcam}}} = \hat{\mathbf{h}}_{\text{EOcam}_i}\left(\hat{\mathbf{x}}_k\right)$$

with respect to the state vector $\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{k_1} & \hat{\mathbf{x}}_{k_2} \end{bmatrix}^\top$, with $\mathbf{x}_{k_1}$ being the position components and $\mathbf{x}_{k_2}$ the velocity components, both in the ENU reference frame. The measurement matrix **H** basically consists of the blocks $\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \dots & \mathbf{H}_i & \dots & \mathbf{H}_N \end{bmatrix}^\top$ with N the number of sensors and decomposing each $\mathbf{H}_i$ in 8 blocks results in the following structure:

$$\mathbf{H}_i\left(\mathbf{x}_k\right) = \frac{d\hat{\mathbf{h}}_i\left(\hat{\mathbf{x}}_k\right)}{d\hat{\mathbf{x}}_k} = \begin{bmatrix} \frac{d\hat{\mathbf{h}}_{\text{EOcam}_i}\left(\hat{\mathbf{x}}_{k_1}\right)}{d\hat{\mathbf{x}}_{k_1}} & \frac{d\hat{\mathbf{h}}_{\text{EOcam}_i}\left(\hat{\mathbf{x}}_{k_1}\right)}{d\hat{\mathbf{x}}_{k_2}} = \mathbf{0}_{2\times 3} \end{bmatrix} \tag{22}$$

The measurements depend only on the position components and thus $\mathbf{x}_{k_1}$. All nonzero blocks in Eq. (22) are elaborated next:

$$\frac{d\hat{\mathbf{h}}_{\mathrm{EOcam}_i}\left(\hat{\mathbf{x}}_{k_1}\right)}{d\hat{\mathbf{x}}_{k_1}} = \begin{bmatrix} \frac{d\hat{\alpha}_i}{dE_i} & \frac{d\hat{\alpha}_i}{dN_i} & \frac{d\hat{\alpha}_i}{dU_i} \\ \frac{d\hat{\gamma}_i}{dE_i} & \frac{d\hat{\gamma}_i}{dN_i} & \frac{d\hat{\gamma}_i}{dU_i} \end{bmatrix} = \begin{bmatrix} \frac{d\hat{\alpha}_i}{dX_{S_i}} & \frac{d\hat{\alpha}_i}{dY_{S_i}} & \frac{d\hat{\alpha}_i}{dZ_{S_i}} \\ \frac{d\hat{\gamma}_i}{dX_{S_i}} & \frac{d\hat{\gamma}_i}{dY_{S_i}} & \frac{d\hat{\gamma}_i}{dZ_{S_i}} \end{bmatrix} \cdot \mathbf{M}^\top = \begin{bmatrix} \frac{\hat{X}_{S_i}\cdot\hat{Z}_{S_i}}{\hat{\rho}_{\mathrm{hor}_0}\hat{\rho}_0^2} & \frac{\hat{Y}_{S_i}\cdot\hat{Z}_{S_i}}{\hat{\rho}_{\mathrm{hor}_0}\hat{\rho}_0^2} & \frac{-\hat{\rho}_{i_{\mathrm{hor}}}}{\hat{\rho}_0^2} \\ \frac{-\hat{Y}_{S_i}}{\hat{\rho}_{\mathrm{hor}_0}^2} & \frac{\hat{X}_{S_i}}{\hat{\rho}_{\mathrm{hor}_0}^2} & 0 \end{bmatrix} \cdot \mathbf{M}^\top \tag{23}$$

The denominators $\hat{\rho}_0$ and $\hat{\rho}_{\mathrm{hor}_0}$ in Eq. (23) are singularity protected as follows:

$$\hat{\rho}_0 = \max\left(\hat{\rho}_i, \varepsilon_\rho\right) \tag{24}$$

$$\hat{\rho}_{\mathrm{hor}_0} = \max\left(\hat{\rho}_{i_{\mathrm{hor}}}, \varepsilon_{\rho_{\mathrm{hor}}}\right) \tag{25}$$

where ε_ρ and $\varepsilon_{\rho_{\mathrm{hor}}}$ are fairly small positive values to prevent singularities.

2. Innovation covariance matrix $\mathbf{S}$

The innovation covariance matrix is calculated as follows:

$$\mathbf{S} = \mathbf{H}\hat{\mathbf{P}}_k\mathbf{H}^\top + \mathbf{R}_{\mathrm{sensors}} \tag{26}$$

where: $\mathbf{R}_{\mathrm{sensors}} = \mathrm{diag}\left[\mathbf{R}_{\mathrm{sensor}_1} \ldots \mathbf{R}_{\mathrm{sensor}_i} \ldots \mathbf{R}_{\mathrm{sensor}_N}\right]$ with the diagonal elements per sensor location defined as follows:

$$\mathbf{R}_{\mathrm{sensor}_i} = \mathrm{diag}\left[\sigma^2_{\alpha_{\mathrm{EOcam}_i}} \quad \sigma^2_{\gamma_{\mathrm{EOcam}_i}}\right]$$

where only the hardware relevant elements per sensor location are kept. A diagonal matrix of sensor valid flags is defined similarly, $\mathbf{M}_{\mathrm{sensors}_{\mathrm{valid}}} = \mathrm{diag}\left[\mathbf{M}_{\mathrm{sensors}_{1_{\mathrm{valid}}}} \ldots \mathbf{M}_{\mathrm{sensors}_{i_{\mathrm{valid}}}} \ldots \mathbf{M}_{\mathrm{sensors}_{N_{\mathrm{valid}}}}\right]$ with:

$$\mathbf{M}_{\mathrm{sensors}_{i_{\mathrm{valid}}}} = \mathrm{diag}\left[\mu_{\mathrm{EOcam}_i} \quad \mu_{\mathrm{EOcam}_i}\right]$$

These valid flags $\mu_{\bullet_i}$ are one for valid sensor readings or zero for invalid ones. These flags allow to take into account lower sampling rates for specific sensors, signal dropouts (e.g. object is outside field of view), sensor malfunctions, invalid signals, etc. This setup assumes that there is a monitoring algorithm in place that drives the valid flag values accordingly. This valid flag matrix is needed for calculating the inverse of the innovation covariance matrix:

$$\mathbf{S}^{-1}_{\mathrm{valid}} = \mathbf{M}_{\mathrm{sensors}_{\mathrm{valid}}} \cdot \mathbf{S}^{-1} \cdot \mathbf{M}_{\mathrm{sensors}_{\mathrm{valid}}} \tag{27}$$

3. Gain matrix $\mathbf{W}$

The Kalman filter gain matrix is calculated with all the aforementioned information as follows:

$$\mathbf{W} = \hat{\mathbf{P}}_k\mathbf{H}^\top\mathbf{S}^{-1}_{\mathrm{valid}} \tag{28}$$

$$\mathbf{W}_{\mathrm{valid}} = \mathbf{W}\mathbf{M}_{\mathrm{sensors}_{\mathrm{valid}}} \tag{29}$$

Invalid flags in $\mathbf{M}_{\mathrm{sensors}_{\mathrm{valid}}}$ will result in zero columns in $\mathbf{W}_{\mathrm{valid}}$. As a consequence, that sensor information will not be used for the correction step. This means that the prediction step will have a higher relative importance for the states that are the main drivers of these invalid measurements, except when there are other valid sensor measurements that contain significant information of these states. When valid measurements are sparse, the Kalman Filter will extrapolate a trajectory that will gradually diverge from reality. This growing uncertainty is indicated by the rank and/or condition number of the observability matrix, as well as by the standard deviation of the estimates.

To prevent overcompensation, the gain matrix $\mathbf{W}_{\mathrm{valid}}$ is rescaled in case one of the matrix entries exceeds a certain upper threshold $\varepsilon_{\mathrm{limit}}$: If $\mathbf{W}(i,j)_{\mathrm{valid}} > \varepsilon_{\mathrm{limit}}$ for any i and j, then define the scaling factor $\sigma_{\mathrm{scale}} = \frac{\varepsilon_{\mathrm{limit}}}{\max \mathbf{W}(i,j)_{\mathrm{valid}}}$ and rescale the weighting matrix as follows: $\mathbf{W}(i,j)_{\mathrm{valid}_{\mathrm{scaled}}} = \sigma_{\mathrm{scale}}\mathbf{W}(i,j)_{\mathrm{valid}}$. Similarly, if $\mathbf{W}(i,j)_{\mathrm{valid}} < -\varepsilon_{\mathrm{limit}}$, define the scaling factor $\sigma_{\mathrm{scale}} = \frac{-\varepsilon_{\mathrm{limit}}}{\max \mathbf{W}(i,j)_{\mathrm{valid}}}$ and rescale the weighting matrix as follows: $\mathbf{W}(i,j)_{\mathrm{valid}_{\mathrm{scaled}}} = \sigma_{\mathrm{scale}}\mathbf{W}(i,j)_{\mathrm{valid}}$.

E. State Update $\mathbf{x}_k$

The state update step relies on the total sensor measurements vector covering all sensor locations, which will give only measurement values for the locations where the tracked object is in the field of view.

The innovation is calculated first, which is the difference between the predicted measurements $\hat{\mathbf{z}}_k$ and the actual sensor measurements $\mathbf{z}_k$*, including sensor limitations and perturbations:

$$\varepsilon = \mathbf{z}_k - \hat{\mathbf{z}}_k = \mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k\right) \tag{30}$$

This innovation ε is then used to calculate the state update by means of the previously computed Kalman Gain matrix:

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{W}_{\text{valid}} \cdot \varepsilon \tag{31}$$

F. Geodetic Coordinates Calculation

Next step is calculating the absolute position of the target in WGS-84 coordinates. This is achieved by combining the known sensor position and the relative position of the target with respect to the sensor position, which is the first half of the corrected state vector $\mathbf{x}_{k_1}$. Since both are defined in different ENU reference frames, they are first converted to the ECEF (earth centered earth fixed) reference frame, then added together and converted back to WGS-84.

The sensor position is converted straight from WGS-84 (i.e. latitude, longitude and altitude) to ECEF via the transformation elaborated in the appendix, resulting in the position coordinates $[X \;\; Y \;\; Z]^\top_{\text{ECEF}}$. The relative position of the target with respect to the sensor position in ECEF coordinates is calculated as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(S\rightarrow T)_{\text{ECEF}}} = \mathbf{T} \begin{bmatrix} E \\ N \\ U \end{bmatrix}_k = \mathbf{T}\mathbf{x}_{k_1} \tag{32}$$

where $\mathbf{T}$ is the rotation matrix from the ENU reference frame to the ECEF reference frame, as defined in the Appendix.

The absolute position of the target in ECEF coordinates is then calculated as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{T_{\text{ECEF}}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{ECEF}} + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(S\rightarrow T)_{\text{ECEF}}} \tag{33}$$

As the last step, this absolute target position is converted back to WGS-84 (i.e. latitude, longitude and altitude) via the reverse transformation elaborated in the Appendix.

G. Velocity Update

The velocity components of the target, which are the second half of the corrected state vector $\mathbf{x}_{k_2}$ are defined with respect to the sensor ENU reference frame, these are transformed to the target ENU reference frame as follows:

$$\begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_{k_T} = \mathbf{L}^\top \begin{bmatrix} \dot{E} \\ N \\ U \end{bmatrix}_k = \mathbf{L}^\top \mathbf{x}_{k_2} \tag{34}$$

H. State Error Covariance Update $\mathbf{P}_k$

By first defining:

$$\mathbf{W}_H = \mathbf{I}_{6\times 6} - \mathbf{W}_{\text{valid}}\mathbf{H} \tag{35}$$

where $\mathbf{I}_{6\times 6}$ is a 6-by-6 identity matrix, the state error covariance matrix is then updated in the reference point reference frame:

$$\mathbf{P}_k = \mathbf{W}_H \hat{\mathbf{P}}_k \mathbf{W}_H^\top + \mathbf{W}_{\text{valid}}\mathbf{R}_{\text{sensors}}\mathbf{W}_{\text{valid}}^\top \tag{36}$$

*In the numerical calculations, an 'empty' measurement scalar in $\mathbf{z}_k$ is usually represented by a 'NaN'. In order to prevent numerical issues these have to be replaced by zeros together with their respective valid flag being set to zero in $\mathbf{W}_{\text{valid}}$ so that they are ignored.

which is finally converted to the target reference frame via the aforementioned rotation matrix $\mathbf{L}$:

$$\mathbf{P}_{k_T} = \begin{bmatrix} \mathbf{L} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{L} \end{bmatrix} \mathbf{P}_k \tag{37}$$

Appendix: Reference Frame Transformations for Kalman Filter

Throughout this paper, five different reference frames are used: geodetic (WGS-84), Earth-Centered Earth-Fixed (ECEF), East–North–Up (ENU), body fixed and sensor spherical reference frames. They are all illustrated in Fig. 21.

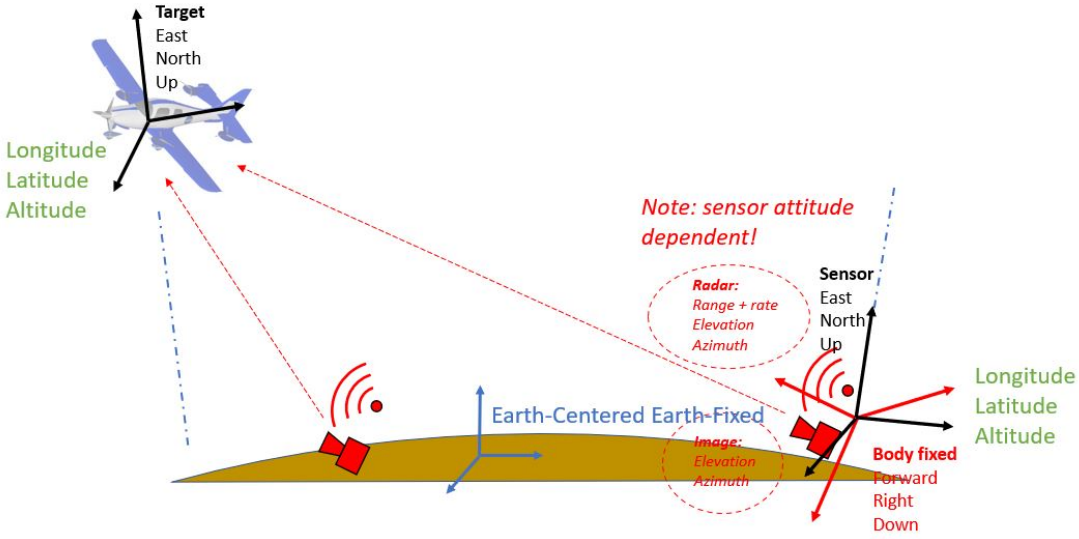


Fig. 21 Reference frames used by Kalman Filter

The geodetic reference frame expresses the spherical coordinates as longitude λ_s with respect to the Greenwich meridian, latitude ϕ_s with respect to the equator and altitude h_s above the average sea-level. The ECEF reference frame is Cartesian with the origin in the Earth center and its coordinates are as follows: the Z_{ECEF} axis extends through true North, X_{ECEF} points through the earth surface where the equator and Greenwich meridian intersect, and Y_{ECEF} is finally perpendicular on the two previous axes. The ENU Cartesian reference frame is sensor centered with the axes pointing East–North–Up. As illustrated in Fig. 21, the 'up' direction is location dependent, as well as the others. As a consequence, the ENU frames for different sensors and target are all differently oriented. This difference increases for increasing distances between all of them. This must be reflected in the calculations. Next is the body fixed reference frame, which is Cartesian and centered at the sensor's center of gravity. The X_B axis points forward, Y_B axis points to the right and the Z_B axis is downwards. Finally, the sensor centered spherical reference frame expresses the coordinates as range ρ (distance from center of gravity), elevation α (vertical angle with respect to the $X_B - Y_B$-plane) and azimuth γ (horizontal angle with respect to the forward axis). The sensor measurements from radar and camera are expressed in this spherical reference frame. Note that both latter sensor fixed reference frames are sensor attitude dependent, i.e. the viewing direction.

The transformations between the different aforementioned reference frames are illustrated in Fig. 22. The transformations between body-fixed, ENU, ECEF as well as between different ENU's of different sensors and target at different locations are pure angular rotations by means of the respective rotation matrices $\mathbf{M}(\phi, \theta, \psi)$ over the three Euler angles between body fixed and ENU, $\mathbf{T}(\lambda_s, \phi_s)$ over longitude and latitude between ENU and ECEF and $\mathbf{L}(\lambda_s, \phi_s, \lambda_t, \phi_t)$ over longitude and latitude of both locations between different ENU's at different locations. The mappings between body fixed and sensor spherical as well as between geodetic and ECEF are more complex nonlinear transformations. Each of them is elaborated next.

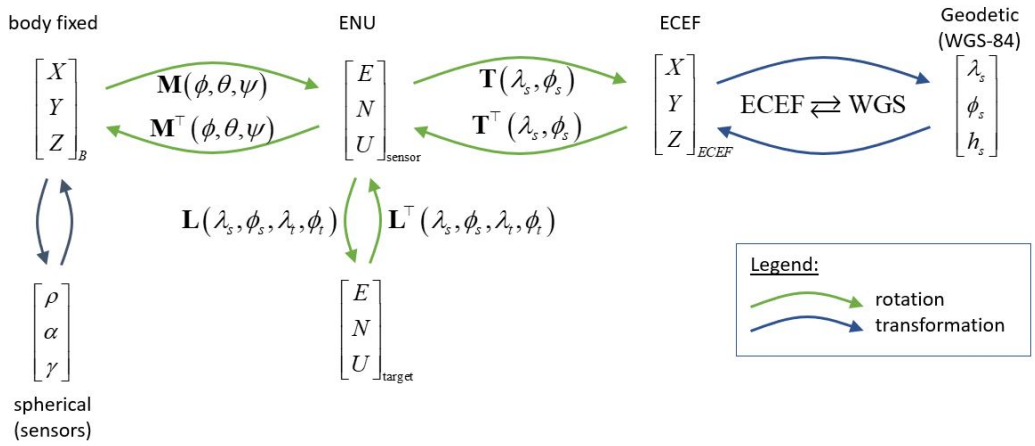


Fig. 22 Transformations between the different reference frames

Transformation from body-fixed to spherical and backwards

The sensor spherical coordinates are defined as a function of the body-fixed coordinates as follows:

$$\rho = \sqrt{X_B^2 + Y_B^2 + Z_B^2} \tag{38}$$

$$\alpha = \arctan \frac{-Z_B}{\rho_{\text{hor}}} \tag{39}$$

$$\gamma = \arctan \frac{Y_B}{X_B} \tag{40}$$

$$\tag{41}$$

where:

$$\rho_{\text{hor}} = \sqrt{X_B^2 + Y_B^2} \tag{42}$$

And the other way around:

$$X_B = \rho \cos\alpha \cos\gamma \tag{43}$$

$$Y_B = \rho \cos\alpha \sin\gamma \tag{44}$$

$$Z_B = \rho \sin\alpha \tag{45}$$

Rotation from body-fixed to ENU and backwards

From body-fixed to ENU is a sequential rotation over the three Euler angles roll ϕ, pitch attitude θ and yaw ψ[21]:

$$\mathbf{M}(\phi,\theta,\psi) = \begin{bmatrix} \sin\psi\cos\theta & \cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta & -\sin\phi\cos\psi + \cos\phi\sin\psi\sin\theta \\ \cos\psi\cos\theta & -\cos\phi\sin\psi + \sin\phi\cos\psi\sin\theta & \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ \sin\theta & -\sin\phi\cos\theta & -\cos\phi\cos\theta \end{bmatrix} \tag{46}$$

The transpose is used for the backwards direction.

Rotation from ENU to ECEF and backwards

From the vehicle centered ENU to the earth centered ECEF is a sequential rotation over the longitude λ_s and latitude ϕ_s of the current position[21]:

$$\mathbf{T}(\lambda_s,\phi_s) = \begin{bmatrix} -\sin\lambda_s & -\cos\lambda_s\sin\phi_s & \cos\lambda_s\cos\phi_s \\ \cos\lambda_s & -\sin\lambda_s\sin\phi_s & \sin\lambda_s\cos\phi_s \\ 0 & \cos\phi_s & \sin\phi_s \end{bmatrix} \tag{47}$$

The transpose is used for the other way around.

Rotation between ENU's

A rotation between ENU's is basically a rotation from the first ENU to ECEF and subsequently from ECEF back to the second ENU:

$$\mathbf{L}\left(\lambda_s, \phi_s, \lambda_t, \phi_t\right) = \mathbf{T}^{\top}\left(\lambda_t, \phi_t\right) \cdot \mathbf{T}\left(\lambda_s, \phi_s\right) \tag{48}$$

Substituting Eq. (47) results in:

$$\begin{aligned}
\mathbf{L} \quad & \left(\lambda_s, \phi_s, \lambda_t, \phi_t\right) = \\
= \quad & \mathbf{T}^{\top}\left(\lambda_t, \phi_t\right) \cdot \mathbf{T}\left(\lambda_s, \phi_s\right) \\
= \quad & \begin{bmatrix} -\sin\lambda_t & -\cos\lambda_t \sin\phi_t & \cos\lambda_t \cos\phi_t \\ \cos\lambda_t & -\sin\lambda_t \sin\phi_t & \sin\lambda_t \cos\phi_t \\ 0 & \cos\phi_t & \sin\phi_t \end{bmatrix}^T \begin{bmatrix} -\sin\lambda_s & -\cos\lambda_s \sin\phi_s & \cos\lambda_s \cos\phi_s \\ \cos\lambda_s & -\sin\lambda_s \sin\phi_s & \sin\lambda_s \cos\phi_s \\ 0 & \cos\phi_s & \sin\phi_s \end{bmatrix} \\
= \quad & \begin{bmatrix} \cos\left(\lambda_t - \lambda_s\right) & \sin\left(\lambda_t - \lambda_s\right)\sin\phi_s & -\sin\left(\lambda_t - \lambda_s\right)\cos\phi_s \\ -\sin\left(\lambda_t - \lambda_s\right)\sin\phi_t & \cos\phi_t\cos\phi_s + \sin\phi_t\sin\phi_s\cos\left(\lambda_t - \lambda_s\right) & \cos\phi_t\sin\phi_s - \sin\phi_t\cos\phi_s\cos\left(\lambda_t - \lambda_s\right) \\ \sin\left(\lambda_t - \lambda_s\right)\cos\phi_t & \sin\phi_t\cos\phi_s - \cos\phi_t\sin\phi_s\cos\left(\lambda_t - \lambda_s\right) & \sin\phi_t\sin\phi_s + \cos\phi_t\cos\phi_s\cos\left(\lambda_t - \lambda_s\right) \end{bmatrix}
\end{aligned} \tag{49}$$

Transformation from ECEF to geodetic and backwards

ECEF ⇒ geodetic

The target position in WGS-84 geodetic coordinates is obtained following the 15-step process presented below (extracted from RTCA DO-317A)[20]:

$$r_{\text{ecef}} = \sqrt{X_t^2 + {Y_t}^2} \tag{50}$$

$$E^2 = a^2 - b^2 \tag{51}$$

$$F = 54 b^2 Z_t^2 \tag{52}$$

$$G = r_{\text{ecef}}^2 + \left(1 - \varepsilon^2\right) Z_t^2 - \varepsilon^2 E^2 \tag{53}$$

$$C = \frac{\varepsilon^4 F r_{\text{ecef}}^2}{G^3} \tag{54}$$

$$S = \sqrt[3]{1 + C + \sqrt{C^2 + 2C}} \tag{55}$$

$$P = \frac{F}{3\left(S + \frac{1}{S} + 1\right)^2 G^2} \tag{56}$$

$$Q = \sqrt{1 + 2\varepsilon^4 P} \tag{57}$$

$$r_o = \frac{-P\varepsilon^2 r_{\text{ecef}}}{1+Q} + \sqrt{\frac{a^2}{2}\left(1 + \frac{1}{Q}\right) - \frac{P\left(1-\varepsilon^2\right)Z_t^2}{Q(1+Q)} - \frac{P r_{\text{ecef}}^2}{2}} \tag{58}$$

$$U = \sqrt{\left(r_{\text{ecef}} - \varepsilon^2 r_o\right)^2 + Z_t^2} \tag{59}$$

$$V = \sqrt{\left(r_{\text{ecef}} - \varepsilon^2 r_o\right)^2 + \left(1 - \varepsilon^2\right) Z_t^2} \tag{60}$$

$$Z_o = \frac{b^2 Z_t}{aV} \tag{61}$$

$$\phi_t = \tan^{-1}\left(\frac{Z_t + \varepsilon'^2 Z_o}{r_{\text{ecef}}}\right) \tag{62}$$

$$\lambda_t = \tan^{-1}\left(\frac{Y_t}{X_t}\right) \tag{63}$$

$$h_t = U\left(1 - \frac{b^2}{aV}\right) \tag{64}$$

where:
$a = 6.378137 \cdot 10^6 m$, semimajor axis of the Earth
$b = 6.3567523142 \cdot 10^6 m$, semiminor axis of the Earth
$\varepsilon = 0.081819190842622$, first eccentricity of the Earth
$\varepsilon' = 0.082094437935831$, second eccentricity of the Earth
ϕ_t = target latitude [rad]
λ_t = target longitude [rad]
h_t = target geodetic altitude [m]

geodetic $\Rightarrow$ *ECEF*

The ownship's Earth-Centered, Earth-Fixed (ECEF) position is calculated by[20]:

$$\begin{bmatrix} X_S \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} (N_s + h_s)\cos\phi_s \cos\lambda_s \\ (N_s + h_s)\cos\phi_s \sin\lambda_s \\ \left(N_s\left(1-\varepsilon^2\right) + h_s\right)\sin\phi_s \end{bmatrix} \tag{65}$$

where:
ϕ_s = ownship geodetic latitude [rad]
λ_s = ownship geodetic longitude [rad]
h_s = ownship geodetic altitude [m]
$N_s = \frac{a}{\sqrt{1-\varepsilon^2 \sin^2\phi_s}}$
$a = 6.378137 \cdot 10^6 m$, semimajor axis of the Earth
$\varepsilon = 0.081819190842622$, first eccentricity of the Earth

If the altitude of the ownship is measured as height above Mean Sea Level (MSL) from a barometric sensor, it should be converted to WGS-84 Height Above Ellipsoid (HAE) using the following steps:

$$\xi(\phi_s) = g_{0_{\text{eq}}} \frac{1 + k\sin^2\phi_s}{\sqrt{1 - \varepsilon^2 \sin^2\phi_s}} \tag{66}$$

$$R(\phi_s) = \frac{a}{1 + f + m_r - 2f\sin^2\phi_s} \tag{67}$$

$$h_S = \frac{R(\phi_s)^h}{\frac{R(\phi_s)\xi(\phi_s)}{\xi\left(\frac{\pi}{4}\right)} - h} \tag{68}$$

where:
ϕ_s = ownship geodetic latitude [rad]
$\varepsilon = 0.081819190842622$, first eccentricity of the Earth
$g_{0_{\text{eq}}} = 9.7803253359 m/s^2$, theoretical gravity at the equator
$g_{0_{\text{pl}}} = 9.8321849378 m/s^2$, theoretical gravity at the poles
$k = \frac{b \cdot g_{0_{\text{pl}}}}{a \cdot g_{0_{\text{eq}}}} - 1$
$\xi(\phi_s)$ = normal gravity on the surface of an ellipsoid at $\phi_s [m/s^2]$
$a = 6.378137 \cdot 10^6 m$, semimajor axis of the Earth

$b = 6.3567523142 \cdot 10^6 m$, semiminor axis of the Earth
$f = \frac{a-b}{a}$, flattening of the Earth
$m_r = 0.003449787$, gravity ratio
$R(\phi_s)$ = radius of the Earth at ϕ_s [m]
h = measured ownship height above MSL
h_s = ownship geodetic altitude [m]

Acknowledgments

This work was performed under the NASA Aeronautics Research Mission Directorate (ARMD), Transformative Tools and Technologies (TTT) Project. The authors would like to thank Zach Roberts, Jonas Jonsson, Matt Gregory, Edgar Torres and the other colleagues of Smart Mobility at NASA Ames Research Center for the successful drone flights, as well as NASA Armstrong colleague Nelson Brown, for their invaluable contributions to this project.

References

[1] Aubuchon, V. V., Hashemi, K. E., Shively, R. J., and Wishart, J. M., "Multi-Vehicle (m:N) Operations in the NAS - NASA's Research Plans," *AIAA AVIATION 2022 Forum*, American Institute of Aeronautics and Astronautics, 2022. doi:10.2514/6.2022-3758.

[2] Lombaerts, T., Shish, K., Keller, G., Stepanyan, V., Cramer, N., and Ippolito, C., "Adaptive Multi-Sensor Fusion Based Object Tracking for Autonomous Urban Air Mobility Operations," *AIAA SciTech Forum*, 2022. doi:10.2514/6.2022-0362.

[3] Lombaerts, T., Kannan, K., Kawamura, E., Dolph, C., Stepanyan, V., Gorospe, G. E., and Ippolito, C. A., "Distributed Ground Sensor Fusion Based Object Tracking for Autonomous Advanced Air Mobility Operations," *AIAA SCITECH 2023 Forum*, American Institute of Aeronautics and Astronautics, 2023. doi:10.2514/6.2023-0896.

[4] Lombaerts, T., Kawamura, E., Kannan, K., Dolph, C., Stepanyan, V., Gorospe, G. E., and Ippolito, C. A., "Development and Field Test Results of Distributed Ground-Based Sensor Fusion Object Tracking," *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024. doi:10.2514/6.2024-1389.

[5] Lombaerts, T., Kawamura, E., Kannan, K., Ippolito, C. A., Stepanyan, V., Gorospe, G., and Dolph, C., "Considerations for Optimal Sensor Placement for Higher Accuracy Object Localization for Urban Air Mobility," *AIAA SCITECH 2025 Forum*, American Institute of Aeronautics and Astronautics, 2025. doi:10.2514/6.2025-0099.

[6] Ippolito, C. A., Martin, R. A., Kawamura, E., Gorospe, G. E., Holforty, W., Kannan, K., Stepanyan, V., Lombaerts, T., Brown, N., and Dolph, C., "Enabling Smart Urban Airspaces through Distributed Sensing Technologies," *AIAA SCITECH 2025 Forum*, American Institute of Aeronautics and Astronautics, 2025. doi:10.2514/6.2025-0344.

[7] Kannan, K., Baculi, J. E., Lombaerts, T., Kawamura, E., Gorospe, G. E., Holforty, W., Ippolito, C. A., 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*, American Institute of Aeronautics and Astronautics, 2023. doi:10.2514/6.2023-0895.

[8] Stepanyan, V., Ippolito, C., Kannan, K., Kawamura, E., Lombaerts, T., Holforty, W., and Dolph, C., "Flight Test Design and Implementation for Airspace Independent Surveillance through a Distributed Ground Based Sensor Network," *AIAA SciTech 2024 Forum*, 2024. doi:10.2514/6.2024-1387.

[9] Kannan, K., Ippolito, C., and Holforty, W., "Flight Test Configuration of the Sensor Payload and the Ground Nodes in Distributed Sensing Frameworks," *AIAA SciTech 2024 Forum*, 2024. doi:10.2514/6.2024-2007.

[10] 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*, American Institute of Aeronautics and Astronautics, 2024. doi:10.2514/6.2024-2010.

[11] Glover, H., Wengrove, M., and Holman, R., "Measuring hydrodynamics and exploring nearshore processes using distributed sensing of fiber-optic cable strain," *Coastal Engineering*, Vol. 190, 2024, p. 104487. doi:10.1016/j.coastaleng.2024.104487.

[12] Karapanagiotis, C., Hicke, K., Wosniok, A., and Krebber, K., "Distributed humidity fiber-optic sensor based on BOFDA using a simple machine learning approach," *Optics Express*, Vol. 30, No. 8, 2022, p. 12484. doi:10.1364/oe.453906.

[13] Mu, S., and Lin, S., "A Comprehensive Survey of Mixture-of-Experts: Algorithms, Theory, and Applications," , 2025. doi:10.48550/ARXIV.2503.07137.

[14] Kawamura, E., Lombaerts, T., Stepanyan, V., Kannan, K., Dolph, C., and Ippolito, C. A., "Hierarchical Mixture of Experts for Advanced Air Mobility Flight Phase Classification," *AIAA SCITECH 2024 Forum*, 2024, p. 1388. doi:10.2514/6.2024-1388.

[15] 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. doi:10.1115/1.4056935.

[16] 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. doi:10.1016/j.robot.2023.104369.

[17] Shish, K. H., Cramer, N. B., Gorospe, G., Lombaerts, T., Stepanyan, V., and Kannan, K., "Survey of Capabilities and Gaps in External Perception Sensors for Autonomous Urban Air Mobility Applications," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021. doi:10.2514/6.2021-1114.

[18] Kawamura, E., Kannan, K., Lombaerts, T., Stepanyan, V., Dolph, C., Brown, N., and Ippolito, C. A., "Vision-Based Distributed Sensing at Vertiports for Advanced Air Mobility and Urban Air Mobility Approach and Landing," *AIAA SCITECH 2025 Forum*, American Institute of Aeronautics and Astronautics, 2025. doi:10.2514/6.2025-0343.

[19] Stepanyan, V., Ippolito, C. A., Kannan, K., Kawamura, E., and Lombaerts, T., "Optimal Communication Topology Determination and Sensor Selection for Independent Airspace Surveillance," *AIAA SCITECH 2025 Forum*, American Institute of Aeronautics and Astronautics, 2025. doi:10.2514/6.2025-0345.

[20] *Minimum Operational Performance Standards for Detect and Avoid Systems*, DO-365, RTCA Inc., 2019, techreport Appendix F: Example Functional Description For Airborne Surveillance Data Processor (ASDP) Subsystem. URL `https://standards.globalspec.com/std/14281907/rtca-do-365`.

[21] Mohinder, S., Grewal, P., Lawrence, R., Weill, A. P., and Andrews, "Appendix C: Coordinate Transformations," *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, Inc., 2006, pp. 456–501. doi:10.1002/9780470099728.app3.