Project 094 Probabilistic Unmanned Aircraft System (UAS) Trajectory and Noise Estimation Tool

Georgia Institute of Technology

Project Lead Investigator

Prof. Dimitri N. Mavris
Director, Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Phone: (404) 894-1557
Fax: (404) 894-6596
Email: dimitri.mavris@ae.gatech.edu

Dr. Holger Pfaender
Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Phone: (404) 385-2786
Fax: (404) 894-6596
Email: holger.pfaender@ae.gatech.edu

University Participants

Georgia Institute of Technology (Georgia Tech)

- P.I.s: Dr. Dimitri Mavris (P.I.), Dr. Holger Pfaender (Co-P.I.)
- FAA Award Number: 13-C-AJFE-GIT-143
- Period of Performance: January 1, 2024, to December 31, 2024
- Tasks:
  1. Investigate statistical sampling techniques
  2. Investigate surrogate noise modeling approaches
  3. Develop an integrated probabilistic noise computation methodology
  4. Extend existing prototype noise engine capabilities
  5. Coordinate with the Federal Aviation Administration (FAA) and Volpe National Transportation Systems Center (Volpe)
  6. Documentation

Project Funding Level

The ASCENT Project 094 is funded by the Federal Aviation Administration (FAA) at the following levels: Georgia Institute of Technology ($335,000).

The Georgia Institute of Technology has agreed to a total of $335,000 in matching funds. This total includes salaries for the project director, research engineers, and graduate research assistants and for computing, financial, and administrative support, including meeting arrangements. Georgia Tech has also agreed to provide tuition remission for students whose tuition is paid via state funds.

Investigation Team

Georgia Institute of Technology

Dimitri Mavris (P.I.)
Holger Pfaender (co-P.I.), Tasks 1-5
Raphaël Gautier (research faculty), Tasks 1-5
Jiacheng "Albert" Xie (research faculty), Tasks 1-5
Venkat Sai Chinta (graduate student), Task 5
Nada Himdi (graduate student), Task 3 and 4
Lloyd Teta (graduate student), Task 2 and 5
Xi Wang (graduate student), Task 1

Project Overview

Context and Motivation

The unmanned aircraft system (UAS) market is expected to grow rapidly in coming years, with projections estimating the civil UAS market at $121 billion in the next decade. Multiple operators are currently developing and testing various concepts of operations that fall within the umbrella of urban air mobility (UAM), with the two main use cases being drone delivery and electric-vertical takeoff and landing (e-VTOL) air taxis. Like traditional aircraft operations, these novel concepts are expected to have an impact on the environment in which they operate, particularly regarding noise. Just as noise assessments of traditional commercial and general aviation fixed wing and rotary aircraft operations are completed today, similar noise assessments for UAM operations will be necessary.

Problem Definition

UAM operations bring unique requirements. First, UAM operations are expected to be denser than current general or commercial aviation operations, possibly by orders of magnitude. Thus, a noise assessment method should be able to manage large vehicle densities. Second, UAM vehicles are expected to be smaller and therefore quieter (for example, small drones for deliveries or helicopter-sized vehicles for e-VTOL air taxis, benefiting from novel electric propulsion systems). As a result, the noise footprint of such vehicles is expected to be more localized. Therefore, noise exposure levels should be estimated with sufficient resolution. Third, instead of primarily following fixed trajectories dictated by approach and departure routes around airports, UAM vehicles are expected to operate point-to-point within populated areas. Departure and arrival locations are expected to vary from day to day; delivery drones may depart from warehouses and deliver goods to different customers every day, and e-VTOL air taxis may allow their customers to be picked up and dropped off all around an urban area. Thus, a noise assessment method should be flexible enough to accommodate changing flight paths, and the resulting noise assessment should account for the variability introduced by these changes.

Research Objective

The objective of this research is to develop a novel noise estimation method/tool that supports the computation of noise resulting from the stochastic operation of UASs and other upcoming vehicle concepts with irregular locations and operations in large numbers.

Research Approach

While the outcome of ASCENT Project 009 provided a solid base for the noise engine, it lacked several capabilities needed to realistically simulate UAS operations and account for the different sources of uncertainty that are the focus of ASCENT Project 094. Thus, in the first phase of this project, the effort primarily consisted of developing these capabilities, which include (1) modeling customer demand, (2) introducing more varied operational concepts beyond simple point-to-point flights, (3) generating realistic trajectories that account for obstacles found in urban areas, (4) updating the simulation logic to provide computational speed-ups and modeling flexibility, and (5) improving the visualizations of probabilistic quantities over a study area. These efforts led to an updated workflow for the integrated capability developed in this project, which is depicted in Figure 1. In addition to capability developments, two collaborations were initiated that aim at validating the approach proposed and implemented for this project.

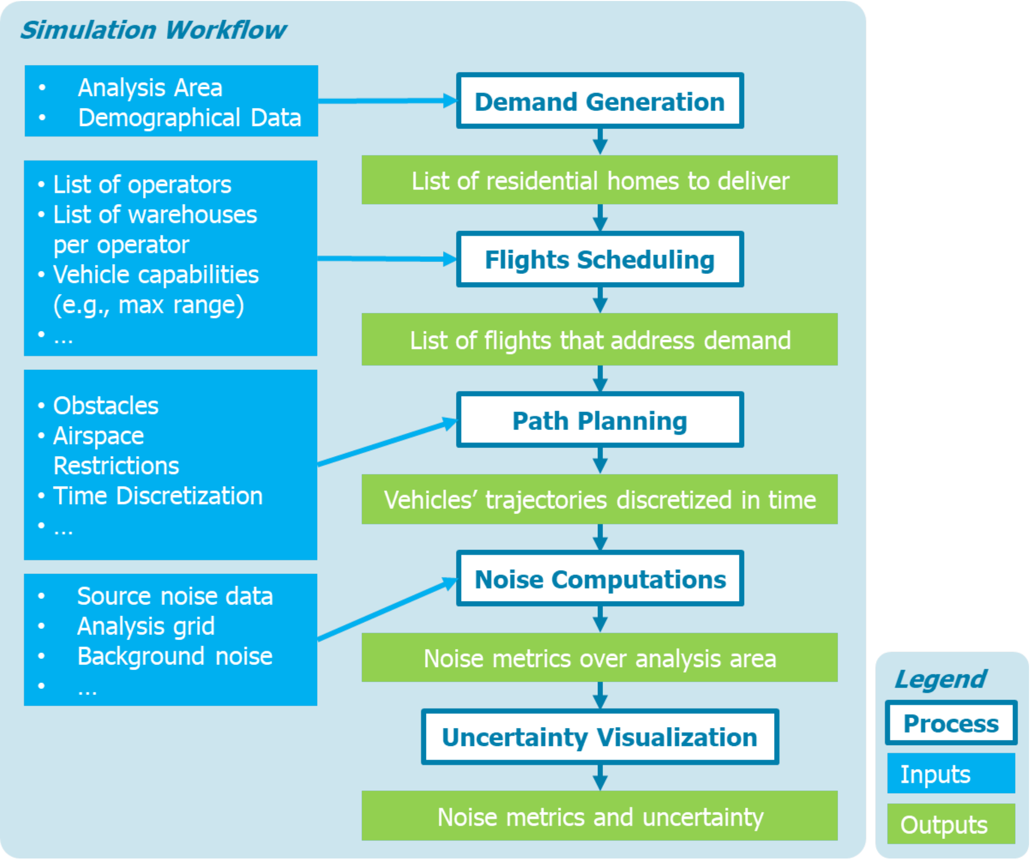


Figure 1. Workflow of the noise assessment logic in Phase 1.

Task 1 – Investigate Statistical Sampling Techniques

Georgia Institute of Technology

Objective

The objective of this task is to produce likelihood distributions for noise exposure maps at every receiver point, it is necessary to assign distributions to the variables that are the primary cause of the associated variability. In the case of UAS noise, these variables are thought to primarily be the day-to-day operations and trajectories. The first experiments performed in ASCENT Project 009 were conducted via Monte Carlo simulations; however, this brute force method is computationally expensive, requiring full reruns of the entire noise modeling with different inputs adjusted to the input variability. Other sampling methods are available that, if applied correctly, allow the computation of identical likelihood distributions but with a significantly reduced number of samples required. In this task, the ASCENT Project 094 team will investigate several sampling techniques and assess their applicability to the noise domain as well as their potential to reduce the computational time required to produce the desired probabilistic noise results.

The goal of the first task is to incorporate demand modeling as part of the analysis workflow such that the time and space distributions of flights match expected operations. In concrete terms, this means that instead of being sampled according to a uniform distribution within the analysis area under study, package delivery destinations are sampled according to the projected demand for online orders, which is estimated based on actual demographical information. To achieve this goal in an efficient manner, the proposed approach applies probabilistic and statistical computations and methods in an openly available demand model with openly available United States (U.S.) Census data.

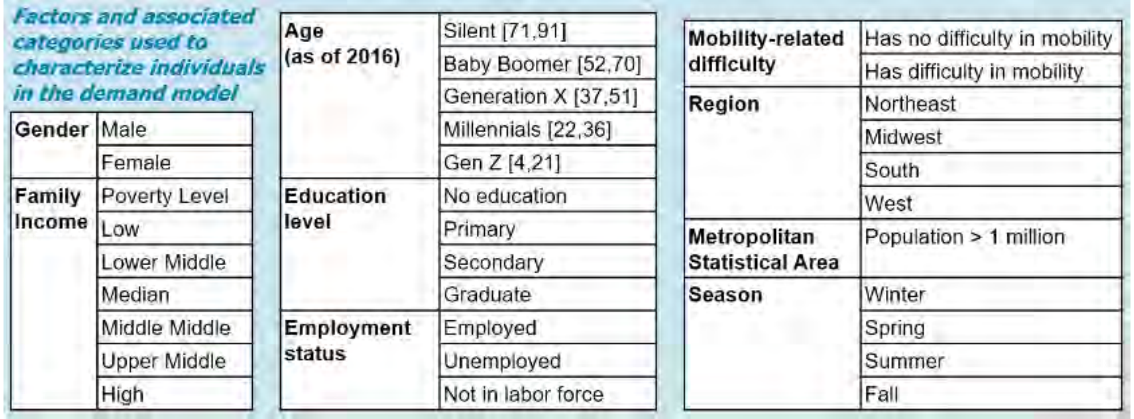
Research Approach

Background

This task utilizes existing multinomial regression models from a published article (Jaller & Pahwa, 2020) to determine the online shopping demand in the study area. In that study, Jaller and Pahwa (2020) delved into the environmental impact of online shopping, focusing on how it influenced ground vehicle miles traveled and emissions. A key question addressed was whether and to what extent online shopping effectively reduces emissions compared with traditional in-store shopping. The findings revealed that the most significant factor in determining the environmental impact is the degree to which orders are consolidated.

The approach taken in this research involved a three-step modeling process. Firstly, the team utilized publicly available U.S. Census data to select representative individuals from two distinct areas: San Francisco, California, and Dallas, Texas. This method ensures that the study is both relevant and reproducible, as it relies on data that are easily accessible. Next, they developed their own demand model, drawing on data from the American Time Use Survey (Hofferth et al., 2017). The relevance of this model is underscored by the fact that the authors have made the resulting demand model available for public use.

Finally, the team conducted a straightforward environmental impact analysis. This analysis aimed to calculate the total number of miles traveled, differentiating between personal vehicle trips and delivery truck journeys. It also estimated the resulting emissions from these trips. However, this particular aspect of the study was deemed irrelevant to our current research interests. Overall, the study provides valuable insights into the environmental consequences of online shopping, particularly in terms of vehicle usage and emissions. Only regression coefficients from their model were used in this subtask, with the required input shown in Figure 2.



Factors and associated categories used to characterize individuals in the demand model		**Age (as of 2016)**	Silent [71,91]	**Mobility-related difficulty**	Has no difficulty in mobility
			Baby Boomer [52,70]		Has difficulty in mobility
			Generation X [37,51]	**Region**	Northeast
Gender	Male		Millennials [22,36]		Midwest
	Female		Gen Z [4,21]		South
Family Income	Poverty Level	**Education level**	No education		West
	Low		Primary	**Metropolitan Statistical Area**	Population > 1 million
	Lower Middle		Secondary		
	Median		Graduate	**Season**	Winter
	Middle Middle	**Employment status**	Employed		Spring
	Upper Middle		Unemployed		Summer
	High		Not in labor force		Fall

Figure 2. Demand model inputs (Jaller & Pahwa, 2020).

Sampling Approach

The team developed a sampling method similar to that of Jaller and Pahwa (2020), as shown in Figure 3. In the demand model used for assessing the impact of online shopping, the Census data play a crucial role by providing the proportion of the population across various categories. These categories are used to create a synthetic population sample by sampling from these categorical distributions. However, it is important to note that this method does not account for dependencies between different factors, such as the relationship between age and family income level. Improvements to the model could be made by investigating ways to incorporate these dependencies.

The core of the demand model is a multinomial logit model, a type of classification model that predicts the probability of belonging to a certain class. In this context, the model considers four classes for a given individual on any given day, based on American Time Use Survey data: (1) no shopping ($k = 1$), (2) in-store shopping only ($k = 2$), (3) online shopping only ($k = 3$), and (4) both in-store and online shopping ($k = 4$).

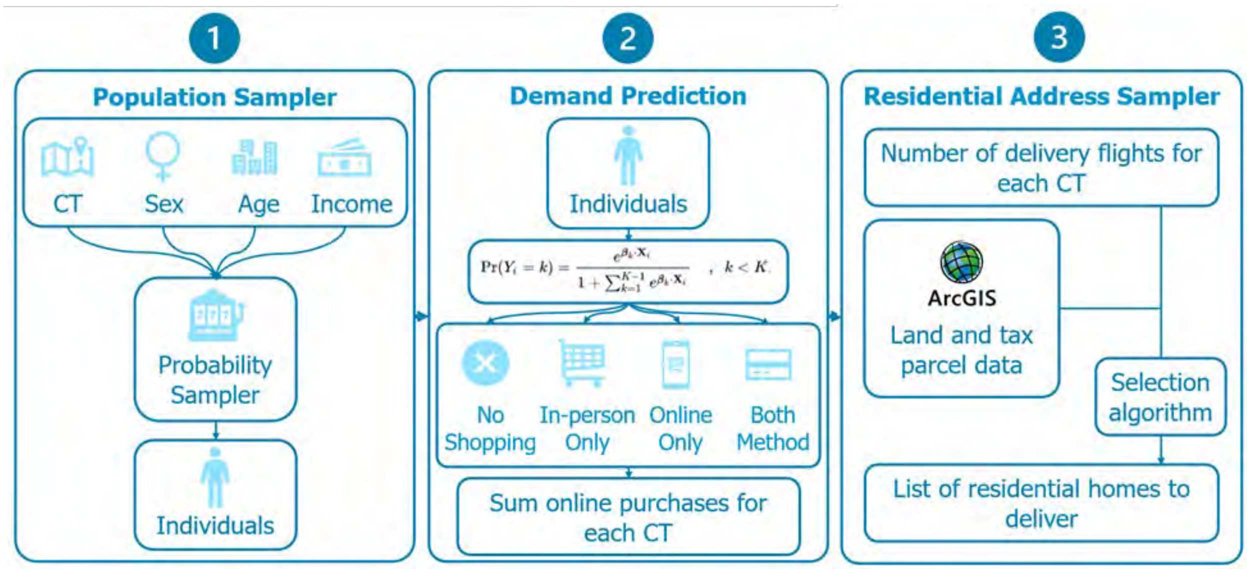

1
Population Sampler
CT
Sex
Age
Income
Probability Sampler
Individuals
2
Demand Prediction
Individuals
$\Pr(Y_i = k) = \frac{e^{\beta_k \cdot X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot X_i}}, \ k < K.$
No Shopping
In-person Only
Online Only
Both Method
Sum online purchases for each CT
3
Residential Address Sampler
Number of delivery flights for each CT
ArcGIS
Land and tax parcel data
Selection algorithm
List of residential homes to deliver

Figure 3. Sampling method workflow (Jaller & Pahwa, 2020). CT: Census tract.

The probability of an individual, characterized by their features $\boldsymbol{X}$**,** falling into one of these classes k is given by the following formula:

$$p(k|\boldsymbol{X}) = \frac{e^{\beta_k X}}{1+\sum_{i=1}^{4} e^{\beta_i X}} \qquad \text{(Eq. 1)}$$

In this equation, the individual's features $\boldsymbol{X}$ are derived from the Census categories. The $\boldsymbol{\beta_k}$ coefficients $(1 \leq i \leq 4)$ are the model coefficients provided by the authors of the study. For the purposes of this research, the primary interest lies in outcomes (3) online shopping only and (4) both in-store and online shopping. These outcomes are particularly relevant for understanding the shift in shopping behaviors toward online platforms and its subsequent impact on environmental factors such as vehicle miles traveled and emissions. A sampling method was first created following the process described in the published article, using Atlanta Regional Committee's census data (Atlanta Regional Committee, 2024).

Probabilistic Approach

To avoid the randomness of the sampling process for individuals in step 1, a probabilistic approach was developed as follows:

1. For every Census tract T_i, generate all possible combinations of household features, and for every combination of household features, evaluate the demand model to obtain the probability for that combination of household features, which allows us to evaluate the following formula:

$$p(household\ orders\ online|household\ in\ census\ tract\ T_i) = \Sigma_k^{N_{household\ info}} \big(p(household\ orders\ online|household\ info = H_k) \times p(household\ info = H_k|censu\ tract = T_i)\big) \qquad \text{(Eq. 2)}$$

There is one such probability per Census tract.

2. For every Census tract, compute $p(household\ in\ census\ tract\ T_i)$ using the numbers of households in each Census tract.

3. At this point, for every Census tract, the actual probability sought can be computed:

$$p(household\ in\ census\ tract\ T_i|\ household\ orders\ online) = \frac{p(household\ orders\ online|household\ in\ census\ tract\ T_i)\ p(household\ in\ census\ tract\ T_i)}{\Sigma_{k=1}^{N_{tracts}} p(household\ orders\ online|household\ in\ census\ tract\ T_k) p(household\ in\ census\ tract\ T_k)} \qquad \text{(Eq. 3)}$$

4. At this point, the information needed to sample from the multinomial distribution for which each category's probability is provided by $p(household\ in\ census\ tract\ T_i|\ household\ orders\ online)$. Sampling from the multinomial distribution gives us a number of deliveries per Census tract.

5. Within each Census tract for which there are deliveries, uniformly sample the corresponding number of homes among this tract's homes.

The strength of this approach lies in its reliance on publicly available nationwide statistics. By using data that are accessible and standardized across different regions, this methodology is not only relevant to the specific areas of study but also generalizable to other regions. This broader applicability is particularly valuable for elucidating and predicting the implications of online shopping and drone delivery in diverse contexts across the country.

Milestones

- Completed the sampling method in September 2023 and integrated with the flight scheduling task.
- Implement the probability approach (in process).

Major Accomplishments

From the implemented demand model, the online shopping demand for the State of Georgia was created using demographic data obtained from the census database (Atlanta Regional Committee, 2024), as presented in Figure 4. The demand was created with Census-tract resolution, corresponding to the smallest demographic data area, thus providing a more detailed and accurate representation of its population characteristics and therefore the online shipping demand.

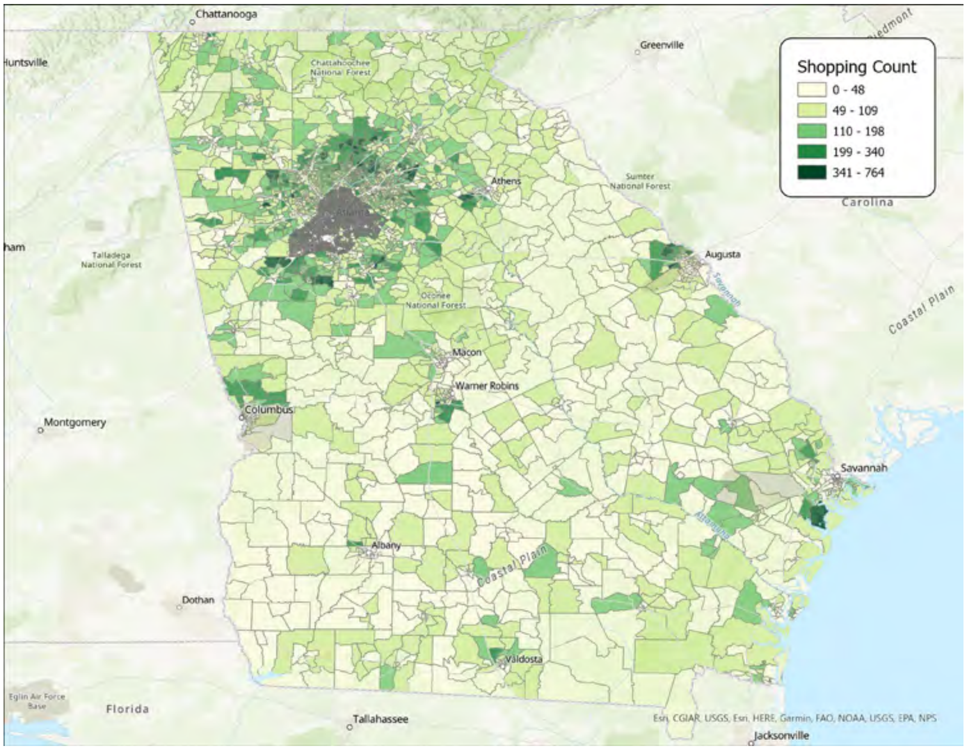


Figure 4. Online shopping demand for the state of Georgia.

Plans for Next Period

- Finalize the updated probabilistic approach for the demand model implementation.

Task 2 – Investigate Noise Surrogate Modeling Approaches

Georgia Institute of Technology

Objectives

In addition to the simplified sampling techniques, it is also possible to replace the full noise modeling with a reduced-order model. This surrogate model is a simplified mathematical representation of a high-fidelity noise study. To generate such a simplified model, a complete high-fidelity study will be created. This high-fidelity study can then be used to sample the input parameter space. Subsequently, the results can be applied to generate a much-simplified reduced-order model. This resulting surrogate model will serve as a stand-in for the full high-fidelity study with high statistical confidence. However, this surrogate model can also be used as a simplified transfer function to which the required input distributions, such as flight operations and trajectory variations, can be rapidly sampled in order to produce the desired output likelihood distributions with significantly reduced computational effort. This aspect is important for achieving rapid and efficient modeling of a potentially large number of vehicles and operations or a large area while still producing distributional results.

The first two efforts undertaken in the context of this task are (1) to reduce the geographical area over which the vehicle noise is propagated to the area in which the propagated sound level is noticeable instead of propagating over the complete analysis area (Subtask 2.a) and (2) to identify and compare a number of surrogate modeling approaches to replace the noise model (Subtask 2.b).

The potential for significantly decreasing the number of computations by reducing the geographical area over which noise is propagated, and therefore the number of receptors to which the noise must be propagated, was identified at the conclusion of ASCENT Project 009 and can be pursued in addition to surrogate modeling. To enable this, a systematic approach for computing a threshold distance around simulated vehicles must be developed that accounts for the fact that the noise level at one receptor location may originate from multiple vehicles.

Research Approach –

Subtask 2.a – Limit Range of Noise Computations

The granularity identified as appropriate in ASCENT Project 009 for obtaining sub-decibel accuracy leads to an analysis grid whose cells are separated by a distance on the order of 15 ft. To cover the geographical spread of a major U.S. city's urban area, this results in an analysis grid comprised of millions of points. In the simulation approach developed in ASCENT Project 009, the noise produced by every simulated vehicle at every time instant of the simulation was propagated to all cells in the analysis grid, resulting in a high computational burden and high computational times. This first effort aimed at assessing the feasibility and expected gains to be obtained by restricting the range around simulated vehicles over which noise would be propagated, on the basis that noise levels rapidly diminish with increasing distance from the noise source because of energy conservation and atmospheric absorption.

Figure 5 depicts the evolution of the sound pressure level as a function of distance from the noise source. The vehicle considered here is a small unmanned hexacopter manufactured by DJI® whose noise footprint was experimentally determined by Heutschi et. al. (2021). We observe that beyond approximately 1.1 km, the sound pressure level falls below typical background noise levels (30 dB). In this case, limiting the propagation distance to this threshold, instead of conducting propagation to all virtual receptors in the study area, would provide significant performance improvements.

® DJI is a registered trademark of Sz DJI Technology Co., Ltd., Shenzhen, China.

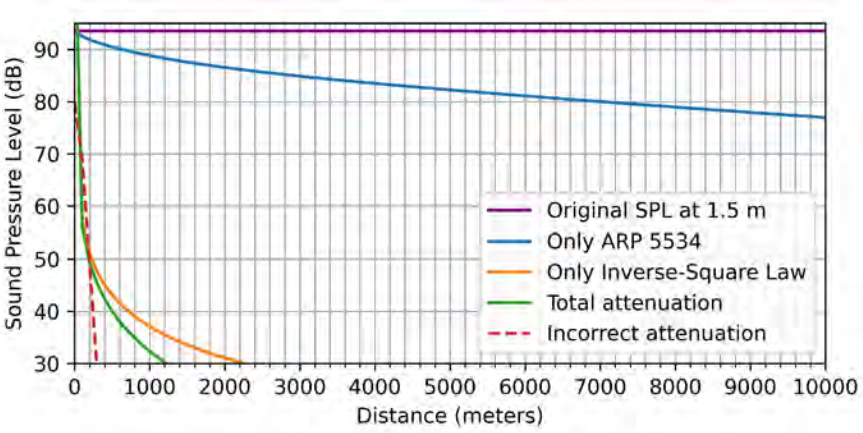


Figure 5. Evolution of sound pressure level (SPL) as a function of distance.

To generalize this approach to all vehicles considered in the simulation, it will be necessary to introduce a step in which this distance threshold is computed for each considered vehicle based on its specific source frequency distribution and amplitude. In addition, the possibility of interactions between multiple vehicles simultaneously located within the range of a given receptor will require additional logic to ensure that such situations do not lead to an underestimation of the resulting sound exposure level. Preliminary formulation work has shown that an analytical approach would allow the threshold distance to be updated based on the maximum number of vehicles concurrently operating within the simulation.

Subtask 2.b – Surrogate Modeling Study

1. *Objectives*

We aim to develop a surrogate model that replicates the output of a high-fidelity noise model with reduced computational complexity. The surrogate model should ensure a high degree of accuracy (high statistical confidence) while offering a considerable time improvement compared with the actual noise model.

2. *Approach*

The methodology involves three high-level steps, starting by identifying and fitting different types of surrogate models and then assessing these surrogate techniques based on computational time and accuracy. The final step is to select the most suitable approach. The following surrogate modeling techniques were considered in the modeling study.

Polynomial Response Surface Equations

The response surface equations (RSE) involve fitting a polynomial equation to the data points generated by the high-fidelity noise model. The general form of a second-order polynomial RSE is as follows:

$$\hat{f}(x) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} x_i x_j \qquad \text{(Eq. 4)}$$

where $\hat{f}$ is the response variable, x_i and x_j are predictor/weight basis functions, and the β terms are coefficients determined by training/fitting the model. The linear regression found in the “scikit-learn” Python® library was used to fit the model (Scikit-Learn Developers, 2023a).

Radial Basis Functions

Radial basis functions (RBFs) are used for interpolation in multidimensional space. They depend on the radius between two training points. Typically, the 2-norm Euclidian radial distance is used to develop an RBF of the following form:

$$\hat{f}(x) = \beta_0 + \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) \qquad \text{(Eq. 5)}$$

® Python is a registered trademark of Python Software Foundation, Beaverton, Oregon.

where λ_i are the weights, ϕ is the RBF, x is the input vector, and x_i are the centers of the RBF. The RBF was implemented by using SciPy's cdist, with a focus on multiquadric functional basis forms and a simple linear solver (Scikit-Learn Developers, 2023b).

Artificial Neural Networks

Artificial neural networks (ANN), inspired by biological neural networks, consist of layers of interconnected nodes. This model can learn nonlinear relationships and is thus capable of capturing complex patterns in data. A typical ANN has a structure with an input layer, one or more hidden layers, and an output layer. The neurons in these layers are interconnected with adjustable weights. The multi-layer perceptron regressor model from the “scikit-learn” Python package was employed (The Scipy Community, 2019).

3. *Interpolation*

Interpolation involves constructing new data points within the range of a discrete set of known data points. Linear interpolation is the simplest form, but higher-order polynomials can also be used for more complex datasets. The model was implemented through SciPy's interp1d (The Scipy Community, 2023).

4. *Surrogate Modeling Process*

Figure 6 depicts the surrogate modeling process.

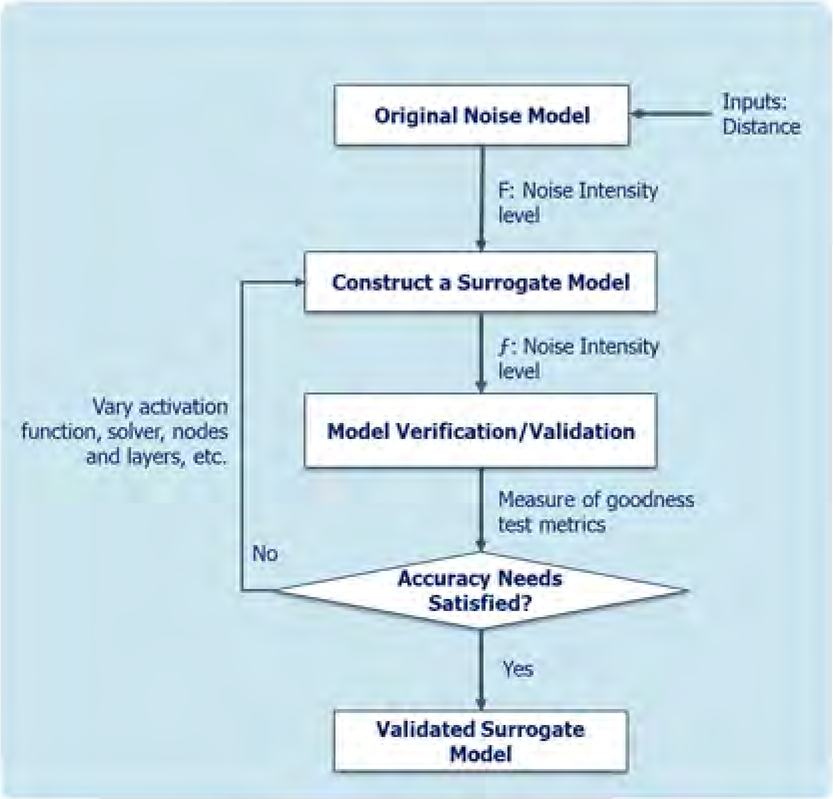

Original Noise Model
Inputs: Distance
F: Noise Intensity level
Construct a Surrogate Model
f: Noise Intensity level
Vary activation function, solver, nodes and layers, etc.
Model Verification/Validation
Measure of goodness test metrics
No
Accuracy Needs Satisfied?
Yes
Validated Surrogate Model

Figure 6. Surrogate modeling workflow.

Data Sampling

The high-fidelity study is used to sample the input parameter space. This step involves selecting a representative subset of the original model's input–output pairs.

Model Training and Validation

The surrogate models are trained on the sampled data. The training process involves adjusting the model parameters to minimize the difference between the model predictions and the actual data. Model validation is then performed on validation data, a set of model observations that were not used to train the surrogate model, thus ensuring the ability of the model to generalize.

5. *Surrogate Assessment*

The trained models were assessed based on two high-level metrics, surrogate accuracy and surrogate computational time. The accuracy of the model was verified based on validation metrics and validation plots.

Validation Metrics

- **Coefficient of determination (R^2):** This metric indicates the proportion of variance in the dependent variable predicted from the independent variables. A value closer to 1 suggests higher model accuracy. The lower the R^2 value, the greater the variance of the error compared with the response variance.
- **Mean absolute error:** This metric measures the average magnitude of the errors in a set of predictions, without considering their direction.

Validation Plots

- **Actual vs. predicted plots:** These plots compare actual values from the high-fidelity noise model with values predicted by the surrogate models. Ideally, the points should lie close to the 45° line, indicating high accuracy. Deviations from this straight diagonal line are due to inaccuracies introduced by surrogate modeling. Figure 7 presents examples of the actual vs. predicted plots used in the validation process.
- **Residuals vs. predicted plots:** These plots illustrate the difference (residuals) between actual and predicted values against the predicted values. The residuals should be randomly distributed and centered around zero.

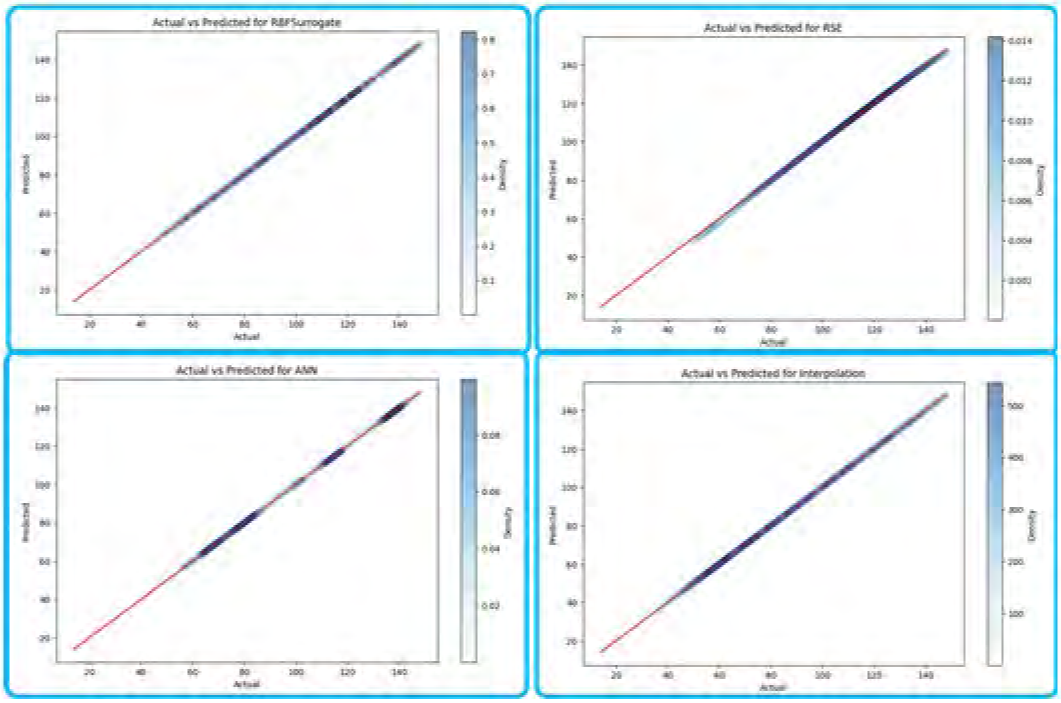


Figure 7. Actual vs. predicted plots for different surrogate modeling approaches.
ANN: artificial neural network; RBF: radial basis function; RSE: response surface equation.

6. *Computational Time Evaluation*

The surrogate computational time represents the time needed to run a trained model for predictions. Note that the time needed to train a model is not considered in the study because we are assessing the computational time required to make predictions: Are we going faster than the original model?

7. *Implementation*

It is important to note that different sizes of prediction samples result in different prediction times; that is, a prediction with many sample points requires more prediction time. However, the computational time may not be linear with respect to the number of prediction samples; thus, we also study the evolution of prediction time vs. the number of prediction points. Prediction batches of different sizes, ranging from 100 to 100,000, were considered. The time is measured using the “timeit” Python module to evaluate the computation time for each prediction batch size. The surrogate is trained once using the same number of points, with the training dataset fixed at 30,000 points.

8. *Results and Surrogate Model Selection*

To ensure surrogate accuracy, validation metrics and plots for different surrogates were assessed. Figure 7 and Figure 8 display the validation plots of actual vs. predicted and residuals vs. predicted, respectively, for each considered surrogate modeling approach. The actual vs. predicted present a 45˚ diagonal line as desired. As stated, any deviations from a straight diagonal line are due to inaccuracies in surrogate modeling. The R² values are also assessed to check for the best fit for each surrogate model. The results in Table 1 show that that RBF and interpolation surrogate models have R² values of 1, indicating an almost perfect fit. The RSE and ANN models also demonstrate high R² values, suggesting a high level of accuracy, as the amplitude of the error is insignificant compared with the total amplitude of the response variations.

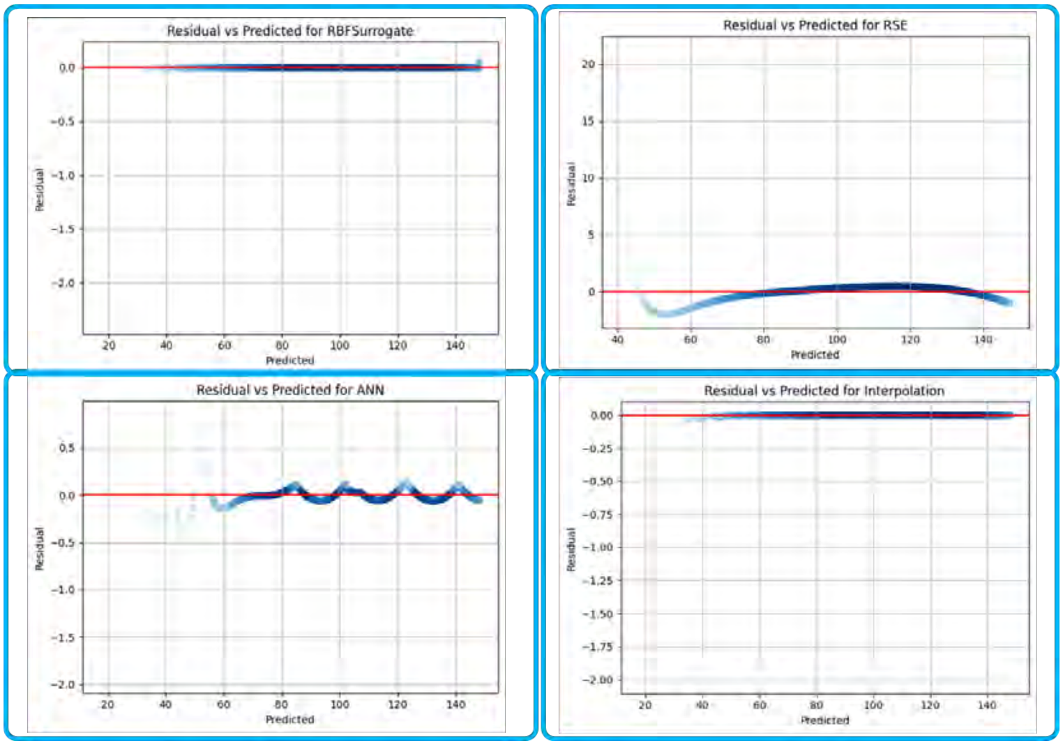


Figure 8. Residuals vs. predicted plots for different surrogate modeling approaches.
ANN: artificial neural network; RBF: radial basis function; RSE: response surface equation.

Table 1. Quality of surrogate models.

Surrogate model	R^2
Radial basis function	1
Response surface equation	0.99695
Artificial neural network	0.99998
Interpolation	1

The residual plot results are consistent with the actual vs. predicted plots: the residuals for the RBF and interpolation models show less deviation, indicating that these models have a higher accuracy in emulating the high-fidelity noise model. The spread of residuals in the RSE and ANN models is wider, indicating a lower degree of accuracy.

1. Impact of Prediction Points

Figure 9 shows two plots depicting the computational time and relative computational time as a function of the number of prediction points for each surrogate model and the actual model. Figure 9 (left) shows the computational time (in seconds, on a logarithmic scale) required by each model as the number of prediction points increases. The surrogate models all start with relatively low computational times for small numbers of prediction points, with the time increasing sharply as the number of prediction points grows. As the number of prediction points increases, the ANN and RBF models show a steady increase in computational time, but they remain significantly faster than the actual model. The RSE and interpolation models demonstrate the best performance among the surrogates, with interpolation being the most efficient. Their computational times increase more slowly and remain several orders of magnitude lower than that of the actual model, even at 100,000 prediction points.

Figure 9 (right) presents the relative computational time, which is a ratio of the computational time of each surrogate model to that of the actual model (on a logarithmic scale). The relative computational time for the ANN and RBF models is above 1, indicating that these models are both less efficient than the actual model, whereas the RSE and interpolation models have relative computational times of less than 1, thus reinforcing their status as being more computationally efficient than the actual model. It is also noted that the efficiency gains of the surrogate models are more pronounced at higher numbers of prediction points.

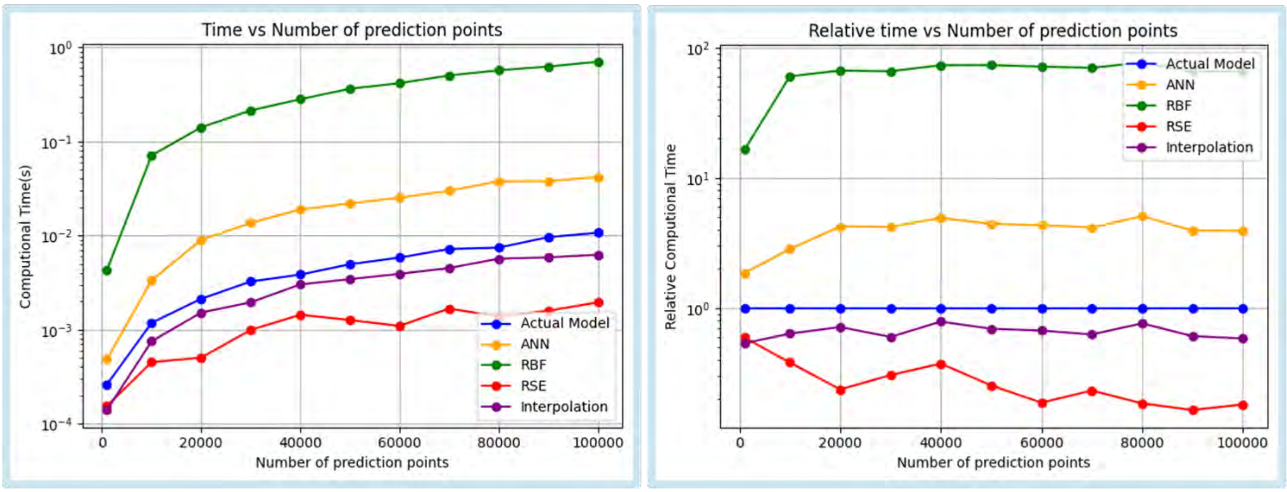


Figure 9. Left: Computational time vs. number of prediction points. Right: Computational times relative to the computational time of the actual model.

2. Surrogate Accuracy vs. Computational Time

Considering that the two attributes assessed for surrogate performance are surrogate accuracy and computational time, we provide a scatter plot for surrogate accuracy vs. computational time to determine whether any of the surrogate models dominate the other surrogates in all attributes, as shown in Figure 10.

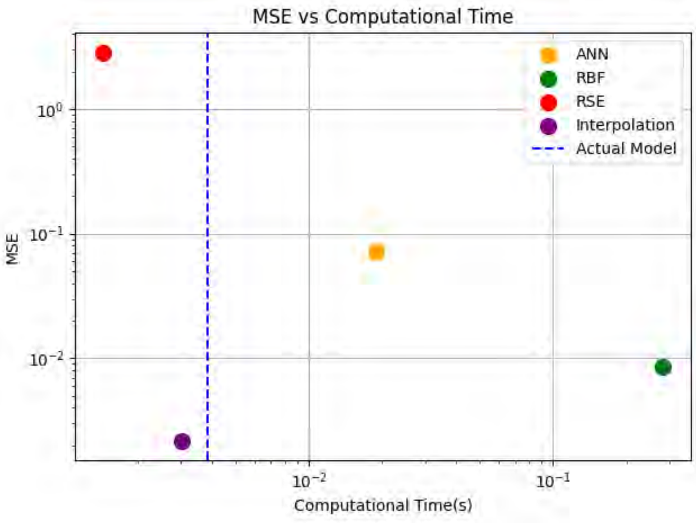


Figure 10. Mean squared error (MSE) vs. computational time.

3. Observations

Interpolation achieves the lowest mean squared error (MSE), indicating that it is the most accurate in making predictions among the surrogates evaluated. The ANN and RSE models have higher MSE values, suggesting that their predictions are less accurate compared with the interpolation and RBF models. The hierarchy of the testing error margin is $\epsilon_{RSE} > \epsilon_{ANN} > \epsilon_{RBF} > \epsilon_{Interpolation}$; thus, interpolation has the lowest error margin, followed by RBF, ANN, and finally RSE with the highest.

4. Actual Model Comparison

The blue dashed line in Figure 10 represents the computational time of the actual noise model. This serves as a benchmark for comparison. The computational times of both the interpolation and RSE models are to the left of this line, indicating that these models are faster than the actual model. Because we seek to accelerate the process, the interpolation and RSE models are the preferable options because of their lower computational times relative to the actual model.

5. Trade-off Analysis

There is an evident trade-off between accuracy and computational time among the surrogate models. Although the interpolation model is the most accurate, it has a higher computational time than the RSE model. In contrast, the RSE model offers a lower computational time at the cost of a higher MSE compared with the interpolation model.
The trade-offs highlighted by Figure 10 suggest that if the highest accuracy is required, interpolation is the best surrogate, given its lower MSE despite its slightly higher computational time compared with RSE. If the computational speed is more critical and some loss of accuracy can be tolerated, RSE might be the preferred choice.

Milestones

None.

Major Accomplishments

Identified and demonstrated multiple avenues for speeding up the noise assessment and making probabilistic assessments more practical. An initial study showed great potential for speeding up the noise assessment compared with the approach previously applied in ASCENT Project 009 by limiting the range of noise propagation. A comparison study of multiple surrogate approaches showed additional promise for speeding up this process.

Plans for Next Period

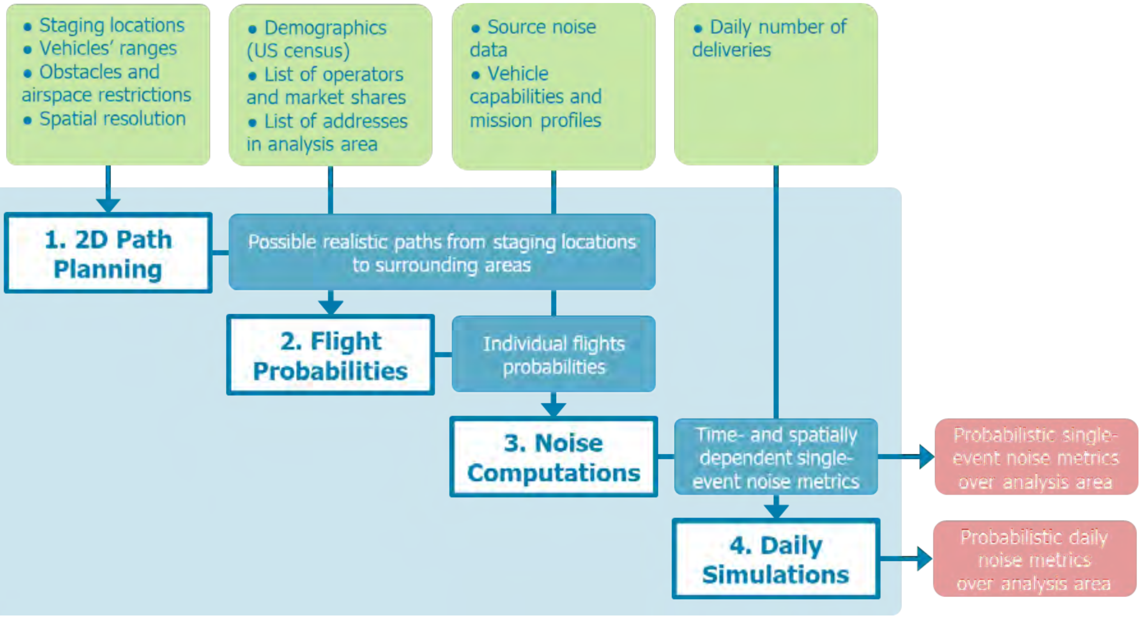
- Propagate the approach aimed at limiting the range over which noise is simulated.
- Implement and integrate the most promising surrogate modeling approaches into the noise assessment tool.

Task 3 – Develop an Integrated Probabilistic Noise Computation Methodology

Georgia Institute of Technology

Objectives

The objective of this task is to recommend a combined methodology that can quickly determine the statistical likelihood of where operations may occur, accompanied or integrated with a probabilistic determination of the resulting UAS noise exposure. ASCENT Project 009 provided a foundational noise engine but lacked several critical capabilities required to simulate UAS operations realistically and address the uncertainties emphasized in ASCENT Project 094. The first phase of this project concentrated on addressing these gaps by developing the ability to generate realistic two-dimensional (2D) path plans from staging locations to surrounding areas, compute flight probabilities to assess the likelihood of deliveries from specific staging locations and perform noise computations to produce time- and spatially dependent single-event noise metrics. Additionally, daily simulations were conducted to calculate probabilistic noise metrics over extended periods. These developments led to an updated workflow for the integrated capabilities, depicted in Figure 11.



• Staging locations
• Vehicles' ranges
• Obstacles and airspace restrictions
• Spatial resolution
• Demographics (US census)
• List of operators and market shares
• List of addresses in analysis area
• Source noise data
• Vehicle capabilities and mission profiles
• Daily number of deliveries
1. 2D Path Planning
Possible realistic paths from staging locations to surrounding areas
2. Flight Probabilities
Individual flights probabilities
3. Noise Computations
Time- and spatially dependent single-event noise metrics
Probabilistic single-event noise metrics over analysis area
4. Daily Simulations
Probabilistic daily noise metrics over analysis area

Figure 11. Updated workflow of drone delivery noise assessment.

This task will integrate the approach investigated in Tasks 1 and 2 and demonstrate the combined efficiency of the improved methodologies investigated to assess the wider noise implications of UAS operations as they expand and include larger operating areas, multiple operators, and multiple vehicles. More specifically, the dynamic nature of UAS operational drivers, including effects of market demand, service point locations (e.g., operating nests or delivery locations), airspace constraints, and possible operator/vehicle limitations, will require more flexible environmental analysis methods that can consider semi-randomization and statistically based quantification of unmanned aircraft activities.

The first effort of Task 3 was to develop the capability to model various operational concepts. Subtask 3.a aims to create a flight scheduling algorithm that produces a list of flights addressing the generated demand while accounting for the fact that multiple operational scenarios are possible, such as departures from multiple staging locations, multiple deliveries per flight, or the presence of multiple operators within the study area. Once the flights have been scheduled, their exact path can be planned, which is the focus of Subtask 4.a.

Research Approach –

Subtask 3.a – Modeling of Diverse Operational Concepts

1. Introduction

The input parameters for the system include (1) a comprehensive list of staging locations, typically warehouses, from where the UAS/drone will initiate the delivery journeys, and (2) a list of delivery locations, primarily residential homes, which will be the end points for these deliveries. Additionally, the system considers the maximum delivery radius for each drone, ensuring that deliveries are feasible given the drone's range. Lastly, the number of deliveries each drone can make per day is considered, which plays a vital role in scheduling and resource allocation.

The operational process of the drone delivery system involves two key steps. In the first step (i.e., the pairing step), each home within the service area is evaluated to determine whether it falls within the maximum delivery radius of any warehouse. If a warehouse is within range, the closest warehouse is assigned to that home. However, if no warehouse is within the maximum delivery radius, the home is not assigned a warehouse, and delivery to that location is deemed not possible. The second step involves the actual scheduling of deliveries. On each simulation day, delivery locations are randomly selected from the list of potential delivery locations. The corresponding warehouse for each delivery location is then identified using a lookup method, ensuring an efficient allocation of resources and timely deliveries.

2. Newly Implemented Operational Concepts

Per-Company List of Warehouses

The first improvement brought to the simulation is the ability to have multiple operators simultaneously operating in the study area, where each operator relies on its own staging locations. At this point, staging locations are assumed to be warehouses; thus, these two terms are used interchangeably in this discussion. This operational concept was implemented as follows, as presented in Figure 12:

- Each company has an attribute "list_of_warehouses" that consists of a list of warehouse identification numbers (ID).
- If warehouse W's ID is on the list, the company's delivery UAS can depart from that warehouse.
- When a company schedules a flight, its warehouses are located to determine from where the vehicle can take off.

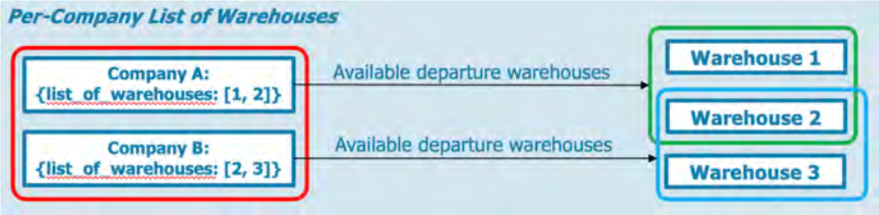

Per-Company List of Warehouses
Company A:
{list_of_warehouses: [1, 2]}
Company B:
{list_of_warehouses: [2, 3]}
Available departure warehouses
Available departure warehouses
Warehouse 1
Warehouse 2
Warehouse 3

Figure 12. Notional illustration of the operational concept of "per-company list of warehouses."

Flights Between Warehouses

The second newly implemented operational concept allows us to model flights between warehouses. Such flights could be used to rapidly balance stocks between warehouses. This ability was implemented by extending the existing scheduling logic to include warehouses as possible destinations, as presented in Figure 13.

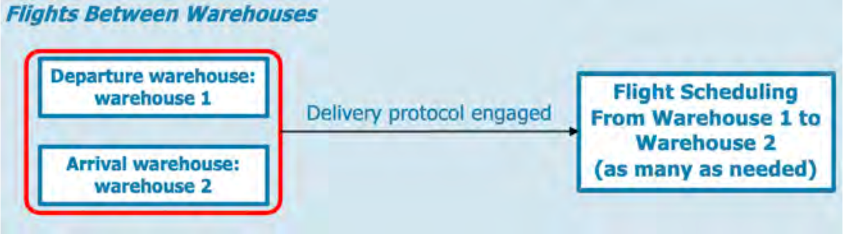


Figure 13. Notional illustration of the operational concept of "flights between warehouses."

Multi-Delivery Flights

The fourth and last additional operational concept is presented in Figure 14. This concept enables multiple deliveries within a single flight based on an optimization function that maximizes the number of houses associated with each drone and minimizes the number of drones used and the total distance flown by each drone. This feature would allow us to assign to each drone a list of houses that are relatively close for delivery.

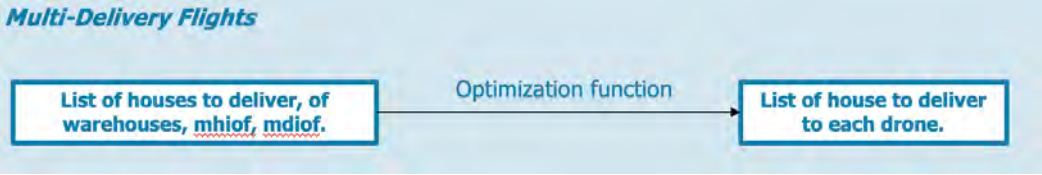

Multi-Delivery Flights
List of houses to deliver, of warehouses, mhiof, mdiof.
Optimization function
List of house to deliver to each drone.

Figure 14. Notional illustration of the operational concept of “multi-delivery flights.”

Examples of New Operational Scenarios

Figure 15 depicts the results obtained by applying the new operational concepts to two study areas: (1) the Memphis, Tennessee, study area (left), and (2) the Atlanta, Georgia, study area (right). All staging and delivery locations are notional at this point. We observe that the optimization algorithm efficiently places the mobile staging locations in order to maximize the number of reachable customers. On the right, it can be observed that the delivery radius of one of the warehouses (green) overlaps with the delivery radius of one of the staging locations (blue); this overlap occurs because the locations correspond to two different operators.

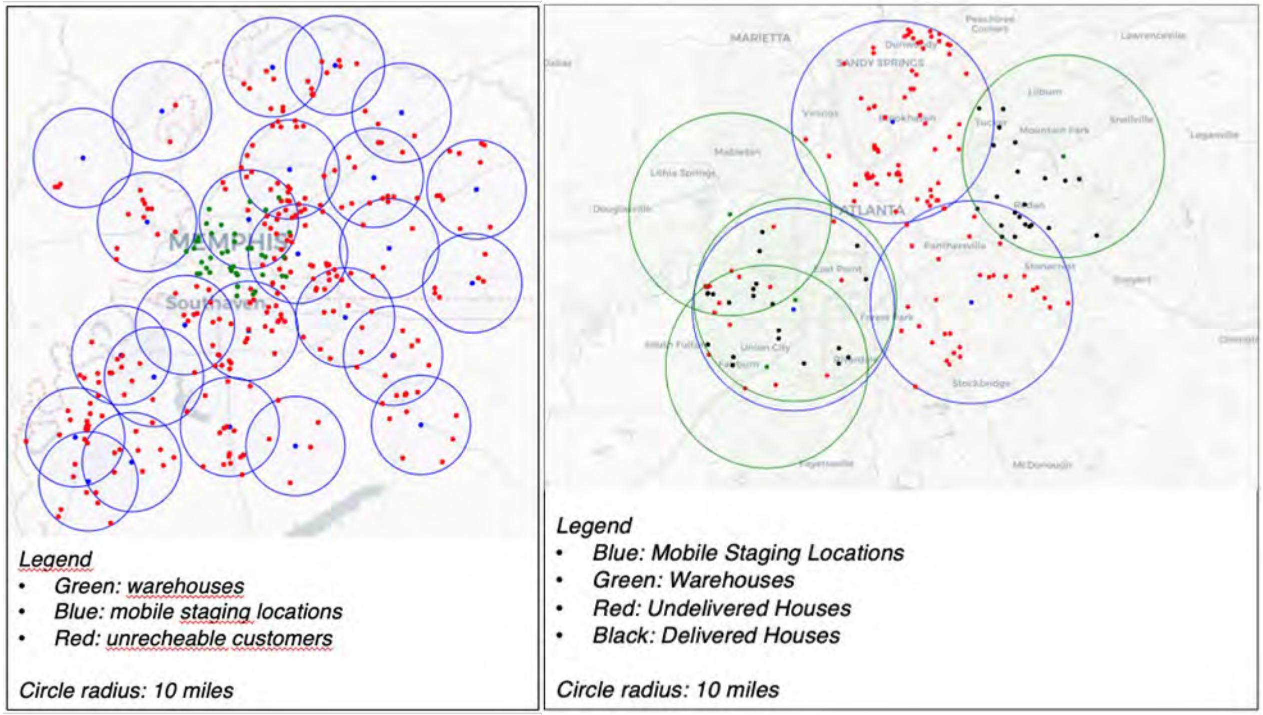


Figure 15. Example of a flight scheduling result over the Memphis, Tennessee, analysis area (left). Example of a flight scheduling result over the Atlanta, Georgia, analysis area (right).

3. *Probabilistic Approach*

To eliminate randomness in pairing warehouses and companies with delivery locations, a probabilistic framework was developed. Singular probabilities are computed at the household, company, and warehouse levels, considering factors such as Census tract demographics, market share, and proximity. These probabilities are then combined to determine the likelihood of a specific warehouse delivering to a particular household. The detailed process is outlined below.

Household-Level Delivery Probability

The household-level delivery probability leverages the approach developed in Task 1, which evaluates the demand model for all possible combinations of household features within each Census tract. By applying this probabilistic framework, the likelihood of a household generating an order is determined based on Census tract demographics and feature probabilities.

Figure 16 illustrates the probability distribution of addresses for potential daily orders. Addresses are color-coded based on the likelihood of an order being placed within a day, with all addresses within a given Census tract sharing the same probability. Order probabilities range between 1% and 6%, but absolute values are not applied directly, as the total number of deliveries is determined by user input. This setup allows flexibility in adjusting delivery volumes while maintaining spatial probability patterns.

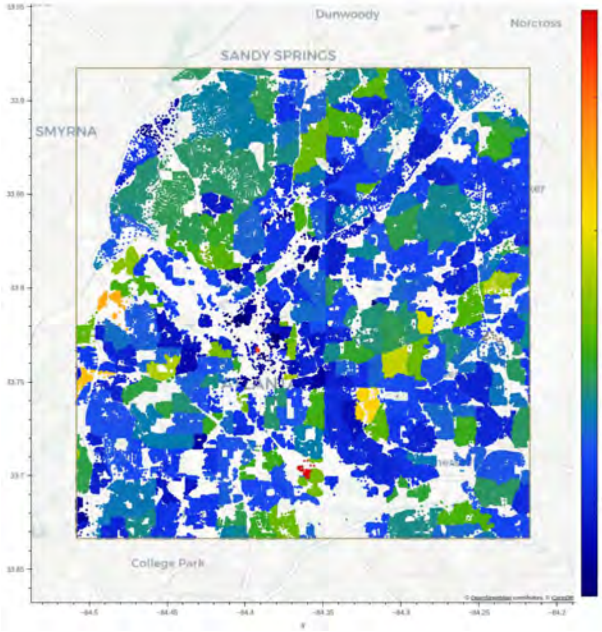


Figure 16. Probability distribution of daily UAS deliveries across addresses in Atlanta.

Company-Level Delivery Probability

This step was introduced to eliminate randomness in determining which operator performs the delivery, as the probability is heavily influenced by the market share of each operator. Operators with a larger market share have a higher likelihood of delivering to a household compared to others. To account for this, in the updated workflow, once the probability of a household receiving a delivery for being in a specific Census tract is determined, it will incorporate the market share of each company. Specifically, we multiply the household probability by the company's market share and then normalize it by dividing by the total market share of all companies delivering to that household:

$$p(household\ online\ order\ delivered\ by\ company\ C_j) = p(household\ in\ census\ tract\ T_i |\ household\ orders\ online) * \frac{Company\ C_j\ market}{Total\ market} \quad \text{(Eq. 6)}$$

Warehouse-Level Delivery Probability

This level was introduced to address the scenario where a company has multiple warehouses capable of delivering to the same household. Instead of randomly assigning a warehouse, we compute the distance between each warehouse and the delivery address, giving the highest probability of delivery to the closest warehouse. This probability is a user-defined parameter. Therefore, the probability of receiving a delivery from a specific warehouse is given by:

$$p(household\ delivered\ by\ warehouse\ W_k\ of\ C_j) = p(household\ online\ order\ delivered\ by\ company\ C_j) * p(warehouse\ w_k) \qquad \text{(Eq. 7)}$$

This final probability is the one we ultimately consider for each household to receive a delivery, as it is more precise and takes all relevant factors into account.

Summary of Flight Probability Computation Process

Figure 17 summarizes the methodology for calculating the probability of occurrence for each flight based on demand distribution. The demand from each address is allocated across the various flights that can reach it, allowing orders at a single address to be fulfilled by different companies or staging locations. Market share is incorporated to determine each company's relative probability of servicing the demand, with preference given to the closest staging location for a given company. In this example, a specific flight pairs address 3,568 with Company A’s staging location A2, demonstrating the assignment logic used to optimize fulfillment probability.

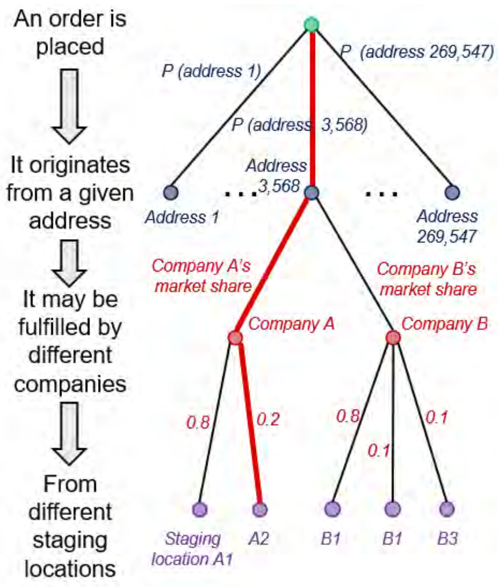


Figure 17. Logical breakdown of flight probability computation for UAS deliveries.

Figure 18 presents the results of applying the new approach to the households in the Atlanta, Georgia, area. Different colors in the figure indicate if a household’s online order can be delivered by multiple companies and/or multiple warehouses. The probability of each household’s online order being delivered by a specific warehouse of a specific company is calculated using the workflow illustrated in Figure 17.

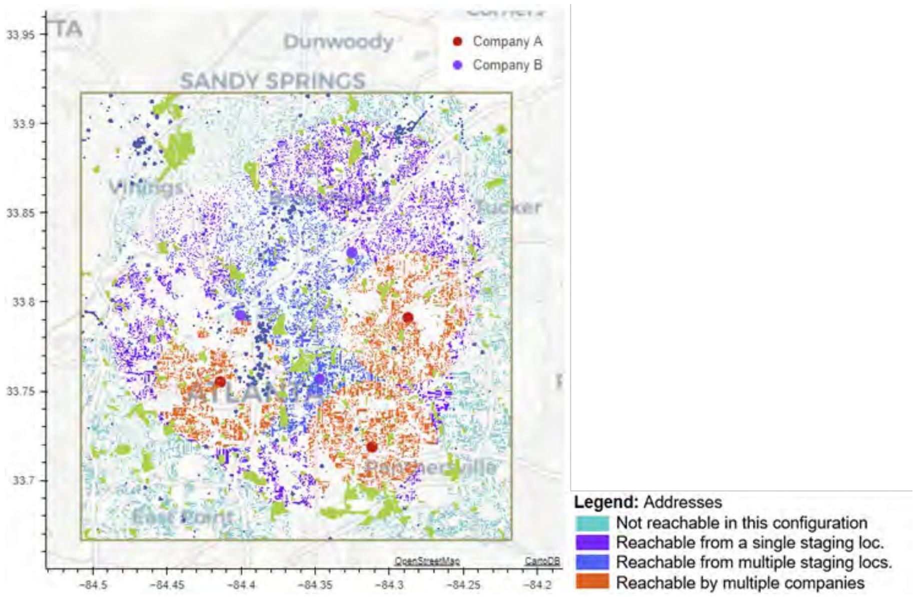


Figure 18. Spatial probability distribution of daily UAS delivery demand across addresses in Atlanta.

Subtask 3.b – Probabilistic Operational Noise Computation

1. *Three-dimensional (3D) Path Planning*

Before performing the noise computation, the 2D paths generated within the grid need to be transformed into 3D paths. To achieve this, optimal 2D paths that enable shortest distance while avoiding obstacles and restricted airspaces are selected and applied with vertical profiles. The details of the 2D generation are provided in Task 4. Two types of vertical profiles are considered, depending on the length of the path. The determination of the appropriate mission type involves evaluating both the horizontal distance of the path and the travel time, calculated using predefined inputs. The required inputs for the mission include cruise velocity, cruise altitude, climb/descent speed, and rate of climb/descent. Using these inputs, the horizontal distance traveled during each segment of the mission (i.e., climb, descent, cruise, etc.) is computed.

Two possible mission profiles are considered: (1) only vertical climb and vertical descent, and (2) vertical climb, cruise climb, vertical descent, and cruise descent. To determine the appropriate mission profile, the total horizontal distance of the path is compared against the requirements for the extended mission. Specifically, the horizontal distance covered during cruise climb and cruise descent is subtracted from the total path length. If the remaining distance is positive, then cruise climb, and descent will be applied in the mission profile. If not, the vehicle will only climb and descent vertically.

Once the vertical profile is determined, it is discretized into segments for further analysis. The discretized profile incorporates altitude variations and serves as the foundation for subsequent noise computations, enabling a detailed assessment of noise emissions along the selected path.

2. *Noise Computations*

Using the discretized 3D trajectories from the previous step, the actual noise computations are performed to analyze the impact of UAS operations on the surrounding environment. This process begins by propagating source noise from the vehicle's position at each time step along its trajectory to virtual receptors distributed across the analysis grid as shown in Figure 19. The propagation accounts for the spatial relationship between the vehicle and each receptor, ensuring precise modeling of noise exposure at different locations.

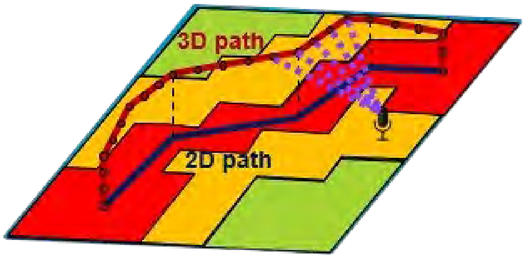


Figure 19. Noise computation along 3D trajectories.

For each flight, the propagated noise data are aggregated over time to calculate single-event noise metrics at each receptor. These metrics provide a detailed snapshot of noise exposure for a specific flight event. Additionally, probability distributions of these single-event noise metrics are generated at each receptor, incorporating the likelihood of different flight scenarios. This probabilistic approach ensures that variations in flight occurrences and trajectories are reflected in the noise analysis.

The outcomes of this step include pre-computed noise footprints for each flight and distributions of single-event noise metrics, such as sound exposure level (SEL), at the grid receptors' locations. These results are critical for understanding the spatial and temporal patterns of noise exposure and provide a foundation for further noise impact assessments.

3. *Daily Simulations*

Using the daily number of deliveries as input, the fourth and final step involves simulating the cumulative noise exposure over an entire day. This process begins by leveraging the pre-computed flight noise footprints, which contain detailed data such as grid cell locations, timestamps, and SPL in decibels. These footprints form the basis for constructing a realistic daily noise simulation.

A sample of flights is drawn to represent the expected number of daily deliveries, and these flights are scheduled across the simulated day. The schedule accounts for temporal distribution, ensuring that flights occur at realistic intervals. The simulation also considers potential interactions between flights, where overlapping noise events may occur due to simultaneous operations.

At each virtual receiver location, the sound levels from individual events are aggregated over time to capture the cumulative noise exposure throughout the day. This process is repeated multiple times using Monte Carlo simulations, where new samples of daily flights are drawn for each iteration. This approach enables the generation of probability distributions for daily noise metrics, such as the day-night average sound level (DNL), at every virtual receptor on the grid.

The outcome of this step is a detailed and probabilistic representation of daily noise exposure. The aggregated sound levels over time reveal the varying intensity of noise at each location, reflecting the effects of multiple single events within the day.

Subtask 3.c – Revised Computational Approach

The research conducted as part of ASCENT Project 009 identified a few shortcomings in the computational approach then adopted. In particular, that approach implicitly assumed that all vehicles independently contributed to the sound level received at receptors; however, it was demonstrated that when operations become sufficiently dense in a given geographical area, multiple vehicles may simultaneously emit noise to receptors at the same location, and the independence assumption may lead to a significant underestimation of the resulting sound level at the receptor. As a consequence, the computational approach was reformulated in order to account for instances in which multiple vehicles may simultaneously contribute to the sound level received at one given receptor location. The revised approach proceeds through time, timestep by timestep, instead of systematically vectorizing all computations.

The revised approach brings two additional benefits. First, because fewer operations are vectorized, a substantial reduction is expected for the memory requirements of the system on which the simulation is run. Second, stepping through time in the simulation is expected to allow for additional flexibility when identifying single events from the viewpoint of individual receptors, allowing us to clearly identify when an event starts and ends based on a pre-defined rule, such as the common "10 dB down" rule.

Milestones

- Completed Subtask 3.a. The different flight scheduling algorithms are fully functional.
- Partially completed Subtasks 3.b and 3.c. The noise analysis modules will be integrated with the flight path generation modules of Subtask 3.a.

Major Accomplishments

- Implement the newly formulated revised simulation approach that provides more flexibility in defining events and computing various noise exposure metrics. Multiple operational concepts have been identified and implemented.

Plans for Next Period

- Finish the integration between the flight trajectory generation modules with the noise analysis module.
- Complete single-event simulations and perform daily simulations with uncertainties using the computed flight probabilities.

Task 4 – Extend Existing Prototype Noise Engine Capabilities

Georgia Institute of Technology

Objectives

The objectives of this task are to (1) implement a path planning algorithm to generate more realistic vehicle trajectories that avoid obstacles found in the study area, such as buildings and other landmarks, and 2) provide the review and selection of a visualization method that would enable the depiction of noise exposure metrics of interest along with the uncertainty associated with these metrics on a geographic information system (GIS)-like dashboard.

Development and testing in Ascent Project 009 demonstrated the potential for possible speed-ups to accelerate the computation of noise grids by using novel computational technologies such as cluster computation. As part of this effort, a prototype was developed that also includes user-interactive visualizations of UAS noise for specific study areas on map displays. This task will be updated to include the estimation of probabilistic distributions of UAS operations based on scalable numbers and locations for service points and stochastic operational frequencies and trajectories over large geographic areas. This task will also extend the prototype to include the capability to integrate the methods proposed in Tasks 1 and 2 and combined in Task 3 to visually present probabilistic distributions of UAS operations and UAS noise exposure over the analyzed geographic areas.

Research Approach

Subtask 4.a – Modeling of Realistic Vehicle Trajectories

1. *Analysis Grid Definition*

Each company operates its own set of warehouses, from which its delivery UAS can depart. These warehouses are considered fixed staging locations. When a company schedules a flight, the company's list of warehouses is checked to determine the vehicle's take-off location.

To define the delivery areas, the range around each warehouse based on the drone's maximum delivery range is calculated, considering half of that range in all directions. Additionally, we factor in the maximum distance at which the drone can be heard, which is controlled by a user-defined noise threshold. Finally, by combining the delivery areas from all warehouses, the total coverage area for the entire delivery network can be established.

2. *Environment Definition*

Before generating the paths, the operational environment was defined to account for obstacles that the UAS must avoid, with these obstacles being user-provided and discretized over the grid. Leveraging the FAA obstacles database, we included critical structures such as buildings and antenna towers, as well as parks, which serve as examples of restricted airspace. Additionally, delivery addresses were sourced from county databases to accurately represent customer locations within the analysis area.

In the context of ASCENT Project 009, vehicles were expected to fly straight from the origin to their destination according to a fixed-altitude profile. Although this approach offered a simple and convenient starting point for the initial proof-of-concept approach, it does not account for the possibly trajectories that vehicles will eventually need to follow in order to avoid the many obstacles existing in an urban environment where package delivery is expected to take place.

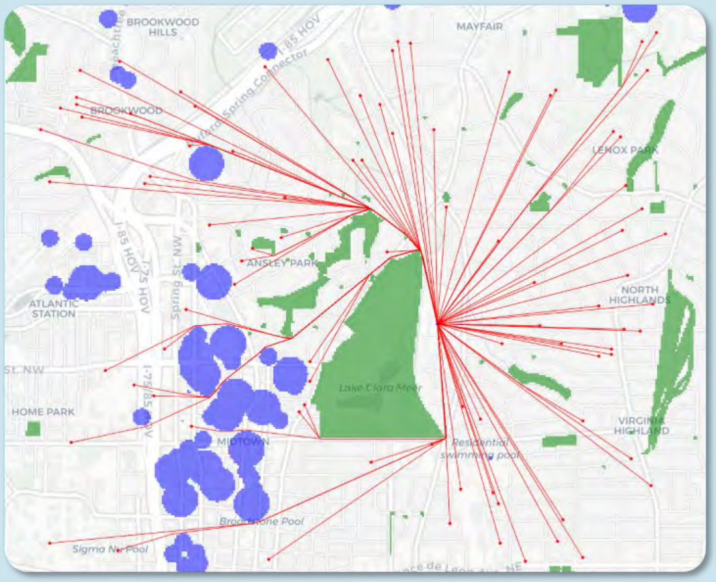
To account for this feature, we considered multiple path planning approaches from the literature. Because of its relative simplicity and maturity, the class of path planning over a discrete grid, including the well-known A* algorithm, was adopted. These algorithms operate over a discretized grid, which is already the case for the analysis grid used for the virtual receptors in the noise simulation. The grid cells obstructed by obstacles are identified and stored. The algorithm finds, if it exists, the shortest path between any starting cell to any destination cell in the grid, while avoiding all cells marked as obstacles.

In the context of ASCENT Project 094, the obstacles currently under consideration are buildings, which are obtained from the FAA obstacle list, and parks, which are obtained from openly available GIS databases. A custom approach was developed to effectively discretize these obstacles on the grid used for path planning. We note here that the developed approach is generalizable to all obstacles defined in the FAA obstacle database; therefore, new updates to the database can easily be imported in the simulation. Similarly, the logic developed for discretizing parks in the path planning grid relies on the definition of obstacles as a set of polygons, which is a generic method that can be generalized to other types of restricted geographical areas beyond parks.

The implementation effort started with a review of existing, off-the-shelf implementation of selected path planning algorithms. This initial review identified a lack of efficient Python implementations. Efficient implementation is crucial in the context of ASCENT Project 094 because path planning must be repeated at the beginning of every simulation run for every scheduled flight. Because probabilistic assessments require many simulation runs to generate distributional results, computational efficiency is a priority. Thus, we developed a custom implementation of the A* algorithm on a two-dimensional grid that does not require instantiation of the full grid, allowing the algorithm to run on a typical personal computer.

Because it only allows movements from one grid cell to its neighbors, the A* algorithm leads to unrealistic trajectories, such as “zig-zag”-shaped paths. A straightforward improvement to this is the Θ* (Theta) algorithm, which allows straight paths between any cells of the grid as long as the path is not obstructed by an obstacle. While conceptually simple, going from A* to Θ* requires the ability to check for obstructions between any two grid cells. This problem can be solved by ray-tracing algorithms. In this instance, computational efficiency is of utmost importance, as this process is repeated at every step of the path planning algorithm. Early testing indicated that a poor implementation could lead to a slowdown by orders of magnitude. As a result, a line-of-sight algorithm was efficiently implemented in Cython, a programming language and suite of tools that allows one to essentially write C code using a Python-like syntax. The resulting code can be compiled and easily called from the main Python program.

The resulting implementation allows one to generate a trajectory on the order of a tenth of a second. The actual duration depends on the granularity of the path planning grid and the distance between the departure and arrival locations; the further the distance between these locations, the greater the number of grid cells that must be explored to find the shortest path. An example of an outcome of path planning process is shown in Figure 20, where approximately 100 flights were generated over the Atlanta Midtown area. The green area in the center is Piedmont Park. Toward the west, multiple tall buildings are present that must be avoided by the vehicles. We observe that all red trajectories successfully reach their destination while avoiding obstacles.



Example trajectories generated by the implemented path planning approach over Atlanta's Midtown area.

- *Blue and green areas are obstacles to avoid (respectively buildings and parks)*
- *A single starting location was selected*
- *Delivery locations were randomly sampled within the area of interest (not actual homes)*

Figure 20. Example trajectories generated by the developed path planning logic. The study area under consideration is the Atlanta, Georgia, Midtown area.

Nevertheless, a critical issue with Θ* alone was that it required recalculating the path for every new flight, which is computationally expensive. This led us to adopt Dijkstra's algorithm[1] (further referred to as "Dijkstra'), which guarantees an optimal path for all grid cells from a given source without needing to recompute the entire path for every flight. Dijkstra allows us to precompute the shortest paths from a warehouse to all potential delivery locations, thus avoiding the need for repeated computations. This feature is particularly valuable when considering multiple delivery locations and obstacles across large urban environments.

The combination of Dijkstra and Θ* provided the optimal balance: Dijkstra ensures that the shortest path is computed once and reused, while Θ* smooths the paths by allowing direct, realistic trajectories as long as obstacles are avoided. This hybrid approach reduced computational overhead while ensuring realistic and efficient path planning in urban environments, where many obstacles like buildings and parks must be considered.

For example, Figure 21 showcases a study area of approximately 25 mi by 15 mi, divided into 50-ft cells (totaling over 5.6 million cells), Dijkstra efficiently planned paths for around 165,000 reachable grid cells marked in (in yellow) in about 5 min. The planned paths, marked in red, avoid obstacles (green) and extend the previously shown examples to longer and more complex paths.

[1] Dijkstra's algorithm is an algorithm for solving many single-source shortest path problems having non-negative edge weight in the graphs i.e., it is to find the shortest distance between two vertices on a graph. It was conceived by Dutch computer scientist Edsger W. Dijkstra in 1956.

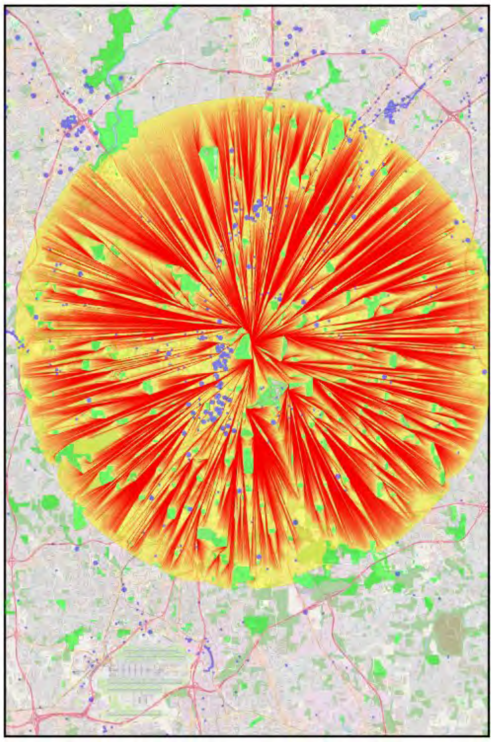


Figure 21. Reachable grid cells using Dijkstra/Theta*.

Upon identifying the reachable grid cells for each warehouse, the next step involves mapping the list of delivery addresses to these grid cells. Each address is checked against the reachable cells to determine if it falls within the delivery range of any warehouse. Addresses that do not correspond to any reachable grid cell are filtered out, ensuring that only addresses within the operational delivery zones are considered for planning. This step is crucial for optimizing the delivery process, as it removes unreachable locations, focusing the path planning on viable delivery targets within the grid.

On a larger scale, a notional example of path planning over the Atlanta, Georgia, area using Dijkstra, starting from six different staging locations is showcased in Figure 22. In this illustration, 1,000 paths are generated from each staging location to visualize potential routes toward customer addresses. The example includes a total of 285,789 addresses within the analysis area, resulting in 303,118 unique paths. This configuration highlights the variability in reachability, as some addresses are accessible from multiple staging locations while others remain unreachable due to environmental constraints and mapped obstacles.

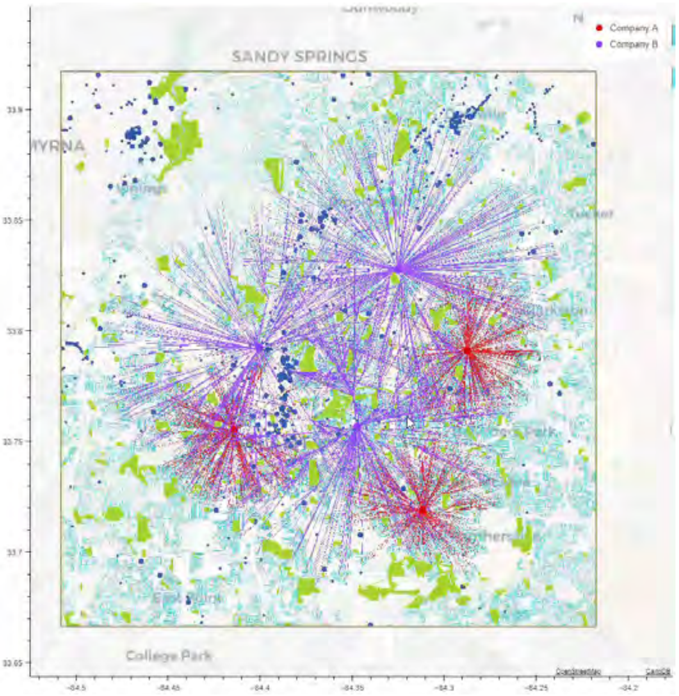


Figure 22. UAS path planning in Atlanta, Georgia, using Dijkstra/Theta.

Subtask 4.b – Visualization of Probabilistic Noise Exposure Metrics Over a Geographical Area
This task commenced with an intense literature review on different frameworks for visualizing uncertainty. Based on the review, we selected the Perlin noise model to visualize the uncertainty for our noise tool.

1. *Perlin Noise Model*

To visualize uncertain data fields, the Perlin noise model integrates color-scale visualization methods by controlling the model parameters. Based on the uncertainty of the data, the model effectively visualizes the uncertain scalar data fields in a combined manner, showing local data value and quality. The color-scale lookup is biased by a Perlin noise weighted by uncertainty data, allowing for a visualization of both the mean value and uncertainty information.

2. *Algorithmic Principle*

The objective of this study is to propose a universal method for visualizing both scalar data and associated uncertainty information, such as the error or standard deviation. The proposed method adheres to the following constraints:

- The approach should extend traditional one-dimensional color maps, ensuring that the resulting visualization remains readily comprehensible to individuals familiar with color maps.
- The representation of uncertainty data should not mask the fundamental data, nor should it introduce complexities, inaccuracies, or challenges in interpretation.

In a conventional 3D color-scale visualization, the data value $V(\vec{x})$ is mapped linearly to a color map coordinate and then to the corresponding red–green–blue color value, which is subsequently displayed in the 3D visualization. The proposed method introduces a modulation to this visualization by applying an uncertain value $U(\vec{x})$*.* The values generated by the Perlin noise model $n_{f_o,p}(\vec{x})$ are linearly scaled to ensure a null mean and to fall within the range of [-1,1]. Subsequently, the scaled noise is multiplied by the uncertainty value and added to the base value $V = (\vec{x})$.

The result is $V(\vec{x})\ +\ n_{f_o,p}(\vec{x})\ *\ U(\vec{x})$, which spans the range $[V(\vec{x}) + U(\vec{x})\ , V(\vec{x})$ - $U(\vec{x})]$. The range of the above resulting distribution will have a mean of $V(\vec{x})$ and serves as a color map coordinate in the visualization.

We implemented the Perlin noise model on a simple normal distribution with varying standard deviation to show the level of increasing uncertainty in the dataset. We employed the same algorithm as detailed above, with the results of the use case shown in Figure 23. The uncertainty of the data is shown to increase from left to right based on our Perlin noise texture without a masking of scalar data. A detailed level of pixel displacement can be achieved by tweaking the parameters of the Perlin noise model that will be employed for our noise computational model.

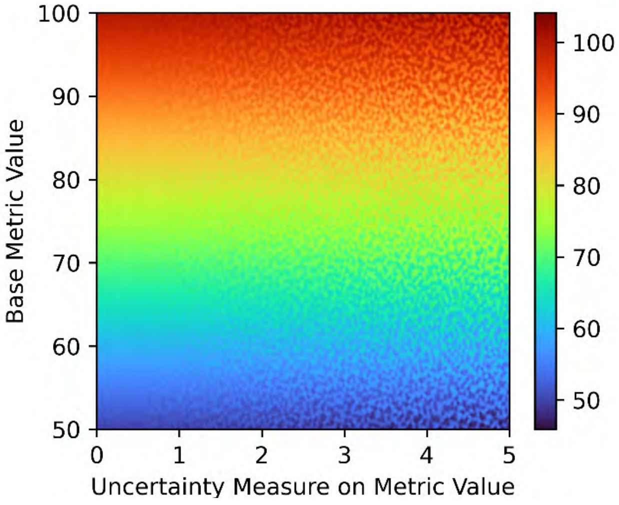


Figure 23. Test case result obtained by using the Perlin noise model.

Milestones

- Completed Subtask 4.a.
- Completed Subtask 4.b (stand-alone). This subtask will be integrated into the entire workflow.

Major Accomplishments

- Implemented and test a first version of the path planning algorithm.
- Selected the Perlin noise model as a method for visualizing noise assessment results.

Plans for Next Period

- Extend the path planning algorithm to the 3D case and integrate all steps of the noise assessment.

Task 5 – Coordinate with the FAA and Volpe

Georgia Institute of Technology

Objectives

The research team will utilize key noise computation methods for which experts reside at the FAA and Volpe. Likewise, to identify potential UAS noise source data and sample study data, it may also be necessary to coordinate with experts at the FAA and Volpe. This includes consultations regarding which noise metrics (e.g., A-weighted, equivalent continuous sound level [LA_{eq}], maximum noise level [LA_{max}], SEL, NA60[2], and potentially others) are the most appropriate for UAS noise and are amenable to probabilistic representations.

Two key collaborations were identified to enhance and validate the noise assessment tool under development. First, through collaboration with the FAA aeroacoustics team, flight test data for UAS vehicles of the relevant size class were obtained. These measurements should (1) enable the introduction of realistic source noise spheres in the simulation and (2) validate the noise propagation technique used in the noise engine. Second, through collaboration with the National Aeronautics and Space Administration (NASA) aeroacoustics team, input data were obtained for a UAM study conducted

[2] Number of flights above the 60 dB (A) per time period (NA = number above)

using the Aviation Environmental Design Tool (AEDT). For this case, replicating this study using the capabilities developed as part of ASCENT Project 094 would enable a higher-level validation.

Research Approach

Subtask 5.a – Leveraging of Causey Airport Measurements

1. *Objectives*

This subtask aims to validate the in-house noise analysis tool (ASCENT Project 094 approach) using Causey Airport UAS acoustics measurements. In 2021, an FAA-led team conducted a set of noise measurements at Causey Airport to characterize the noise produced by UASs during ground and airborne operations. In this experiment, the noise from three vehicles with different weights, speeds, and flight conditions was measured by two sets of microphone arrays: (1) for the flyover and (2) for the hover. Both ground-based and cable-suspended microphones were utilized in both layouts. The measurements were presented in terms of pressure time-history acoustic data stored in American Standard Code for Information Interchange (ASCII) files. This subtask uses the Causey Airport measurements to validate the in-house noise analysis used in this project, which was developed in ASCENT Project 009.

2. *Approach*

The validation involves two major steps. First, vehicle noise source data based on the Causey Airport measurements are generated by using the noise analysis tools developed by Volpe. Second, the in-house noise analysis code is used to re-generate the measured acoustics data based on the generated noise source data and the vehicle trajectories used in the Causey Airport measurements.

Noise Sphere Generation

Generating noise source data for UASs relies on two analysis tools from Volpe: (1) the Advanced Acoustics Model (AAM) and (2) the Acoustic Repropagation Technique (ART). The atmospheric conditions, microphone locations, and flight trajectories used in the Causey Airport measurements are first input to the AAM. Using time- and geometric-based analyses, the AAM calculates the propagation physics and geometric relationship between the noise source (i.e., UAS aircraft) and microphones; it then outputs a time-history file, including the mapped relationship on a sphere for each trajectory point. Along with the aircraft specifications and microphone measurement data, the generated time-history file is given as input to the ART to generate noise source data for the vehicle in terms of a noise sphere. This noise sphere is obtained by applying the propagation corrections determined by the AAM to the measured microphone spectral time histories.

Simulation of Noise Propagation

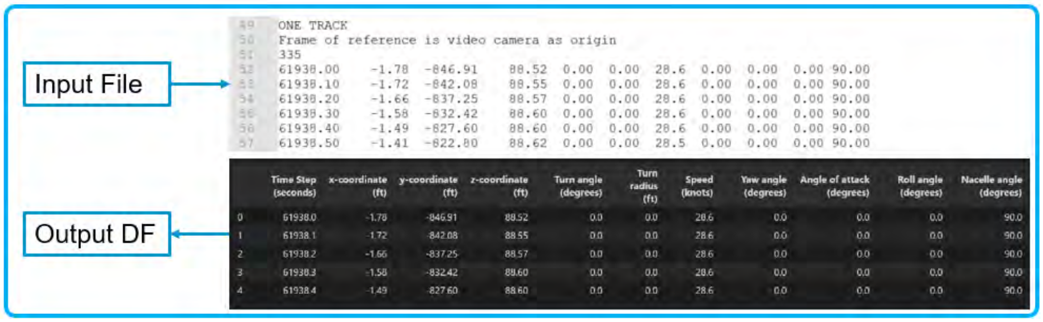
The generated noise sphere is then provided to the in-house noise analysis code. Using the same flight trajectory applied in the Causey Airport measurements, the noise levels at different points of interest are computed by the analysis code based on the noise sphere. In this validation, the points of interest are set to the microphone locations used in the Causey Airport measurements. If the computed noise levels match the measured acoustic data, the accuracy of the in-house noise analysis code will be validated. Running the in-house noise analysis tool for validation consists of five high-level steps. More details about each step are described in the following section.

3. *Replace point noise sources with noise spheres*

To ensure an apples-to-apples comparison between our results (ASCENT Project 094 tool) and results generated using the ART, we needed to provide the same inputs as such as the source noise, grid definition (receptor positions), and series of vehicle positions. Previously, the ASCENT Project 009 tool assumed that the noise source is isotropic, hence additional development was done to add the capability to take a noise source sphere as an input. In addition to computing the distance from a vehicle (source) position to each grid point (receptor), the extra steps added include computing the 3D angles between vehicle location/direction and grid receptor points. The computed angles are then used to interpolate the noise source data for noise propagation.

4. *File Parsing*

The inputs: grid definition, vehicle trajectory and noise source used for AAM propagation are provided as ART file formats that cannot be directly implemented in our Python-based noise analysis tool. The ART input(.INP) file contains Causey Airport trajectories, receptor points are in the comma-separated values (CSV) file and the noise source data are stored as a NetCDF (.NC) format. As such, file parsing algorithms were developed to seamlessly load the input(.INP) file, CSV file and NetCDF (.NC) file in the format expected by the tool. Figure 24 shows an example of extracting flight trajectory data, coordinates and time step from the input file to a data frame.



```
ONE TRACK
Frame of reference is video camera as origin
335
61938.00    -1.78  -846.91    88.52  0.00  0.00  28.6  0.00  0.00  0.00 90.00
61938.10    -1.72  -842.08    88.55  0.00  0.00  28.6  0.00  0.00  0.00 90.00
61938.20    -1.66  -837.25    88.57  0.00  0.00  28.6  0.00  0.00  0.00 90.00
61938.30    -1.58  -832.42    88.60  0.00  0.00  28.6  0.00  0.00  0.00 90.00
61938.40    -1.49  -827.60    88.60  0.00  0.00  28.6  0.00  0.00  0.00 90.00
61938.50    -1.41  -822.80    88.62  0.00  0.00  28.5  0.00  0.00  0.00 90.00
```

	Time Step (seconds)	x-coordinate (ft)	y-coordinate (ft)	z-coordinate (ft)	Turn angle (degrees)	Turn radius (ft)	Speed (knots)	Yaw angle (degrees)	Angle of attack (degrees)	Roll angle (degrees)	Nacelle angle (degrees)
0	61938.0	-1.78	-846.91	88.52	0.0	0.0	28.6	0.0	0.0	0.0	90.0
1	61938.1	-1.72	-842.08	88.55	0.0	0.0	28.6	0.0	0.0	0.0	90.0
2	61938.2	-1.66	-837.25	88.57	0.0	0.0	28.6	0.0	0.0	0.0	90.0
3	61938.3	-1.58	-832.42	88.60	0.0	0.0	28.6	0.0	0.0	0.0	90.0
4	61938.4	-1.49	-827.60	88.60	0.0	0.0	28.6	0.0	0.0	0.0	90.0

Figure 24. Parsing the .inp file to a Pandas DataFrame

5. Distance and Sphere Computation

The sound sphere angles were computed based on the description in the AAM technical manual (U.S. Department of Transportation Volpe Center, 2020). Figure 25 shows the description of the polar angle (theta) and azimuth (phi).

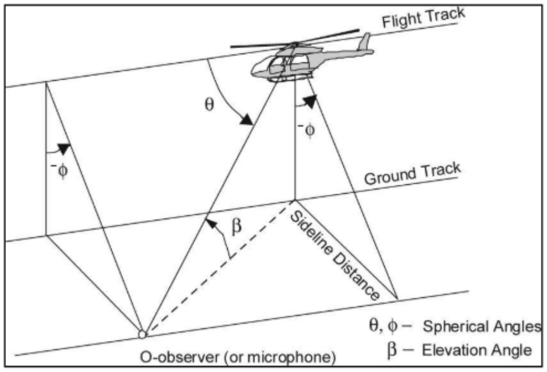


Figure 25. Sound sphere angles, adapted from U.S. Department of Transportation Volpe Center (2020).

For each point on the grid, the tool computes the:

- Distance from a vehicle (source) position to each grid point. This is computed as the length of the line segment between them$d(p,q) = \sqrt{\sum_{i=1}^{3}(q_i - p_i)^2}$ where p, q are two points (receptor and source) in the 3D space.
- The 3D angles; the elevation angle (angle with the z-axis) and the azimuth angle (angle in the x-y plane with respect to vehicle direction) from the source to each point in the grid. The vehicle direction at the point (x_i, y_i, z_i) is calculated as $\vec{v}_i = (dx, dy, dz) = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)$. The vector is then normalized to give a unit direction of the vehicle at each point along the trajectory.

As an example, Figure 26 depicts a noise source as a red dot and the grid of virtual receivers as a scatter plot colored according to the distance from the source. The source coordinates and the angles between the source and one of the receivers are also shown for illustration.

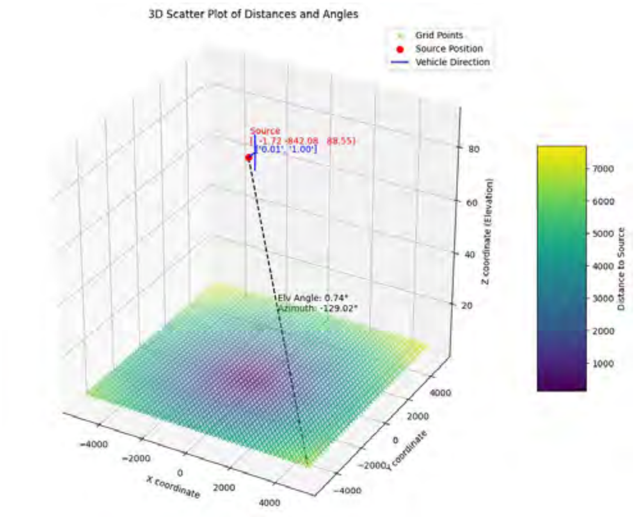


Figure 26. A 3D scatter plot illustrating distances and angles in a spatial context.

6. *Validation Distance and Sphere Computation*

As stated, ensuring the input fed into our tool is the same as for the AAM is important. Thus, after computing the distances and angles, the values were compared with the corresponding values in the time history file generated by the AAM. The scatter plot shown in Figure 27 was utilized to check the evolution of the results with time. This helped identify discrepancies in the azimuth computation which later updated.

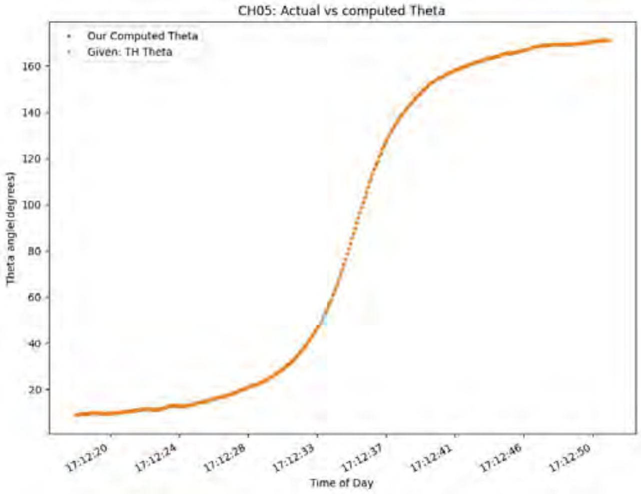


Figure 27. Actual vs. computed polar angle (theta).

7. *Noise Sphere interpolation*

The goal is to interpolate the noise source data (frequency and amplitudes) based on the input noise source angles. The loaded NetCDF noise source has 'PHI', 'THETA', 'FREQUENCY,' and 'AMPLITUDE' which are of shapes (37,), (37,), (31,), (37, 37, 31) respectively. AMPLITUDE data are in the shape of (37, 37, 31) suggests that for each (phi, theta) pair, there are 31 amplitude values, one for each frequency. The Scipy's RectBivariateSpline is used for interpolation due to its ability for vectorization to manage multiple (phi, theta) pairs of inputs.

8. Validating Interpolation

Figure 28 presents the contour plots of the noise levels with and without the interpolation. The left subplot displays the contours for the original amplitudes at the given 37×37 (phi, theta) pair values. A new finer grid of 100×100 (phi, theta) pair values were generated, and the amplitude corresponding to these pairs were interpolated and plotted in the right subplot. These two contour plots are almost the same, indicating that the interpolator is working correctly.

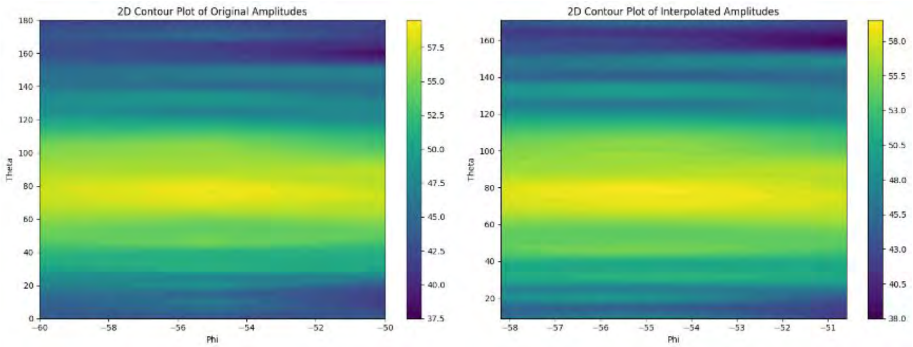


Figure 28. Comparison of 2D contour plots of original and interpolated amplitudes

9. Implementation and Results

Once all the inputs were validated to match the AAM inputs, they were fed into ASCENT Project 094 noise analysis tool for acoustic regeneration. The noise source sphere currently being used for noise propagation is the “FTREX094.nc” for Causey RUN #94. Data from the Causey RUN #94, such as the trajectory, were utilized for the initial validation efforts.

Loading Vehicle Trajectories

Figure 29 showcases the trajectory points for Causey RUN #94. Although the trajectory seems to be changing on the left plot, the vehicle change in the x and z are insignificant as compared to the y-axis. The trajectory contains 335 coordinates.

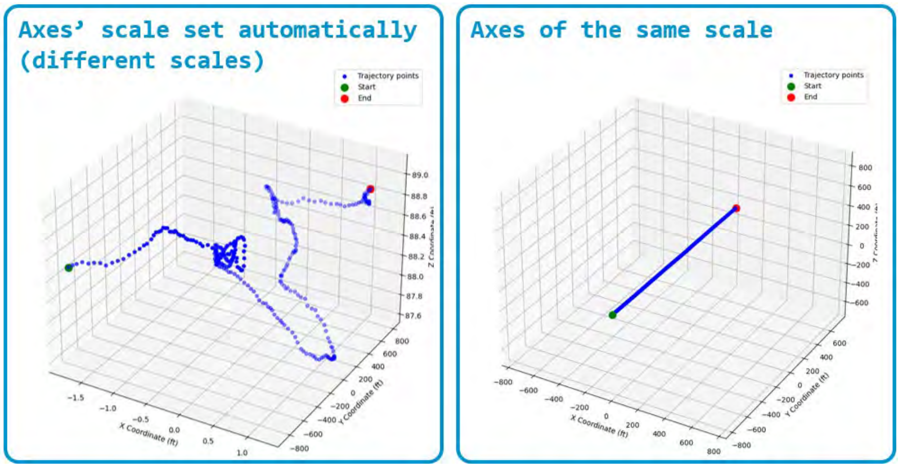


Figure 29. Trajectory points for Causey run #94.

Loading Receptor Locations

The file parsing algorithm is used to load the Causey Airport measurements receptor points. These consist of five receptor points at an altitude z>0 (crane microphones) and nine receptor points on the ground (z=0). In Figure 30, the picture on the left shows the Causey Airport measurement receptor points whereas the plot on the right is the 3D visualization of the loaded points in coordinate system.

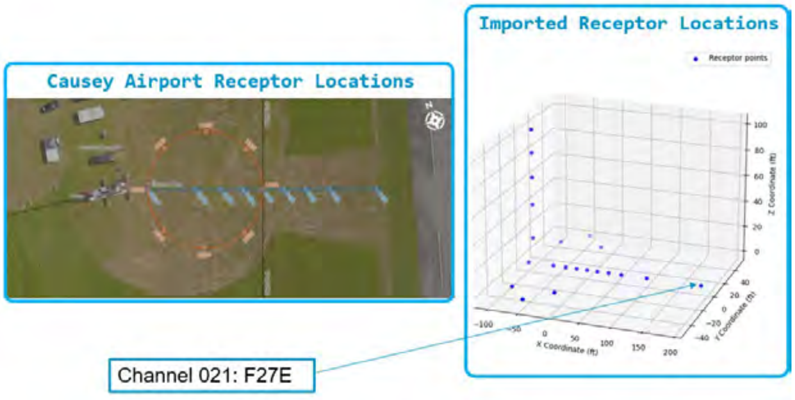


Figure 30. Location of the receivers used in the Causey Airport measurements.

A-Weighting Factors

The acoustic data/results generated by the AAM are A-weighted decibels (dBA). As such A-weighting factors were implemented in the noise analysis tool for both frequency SPL and total SPL. These were implemented based on the standard weighting function which is applied to the amplitude spectrum of unweighted sound level.

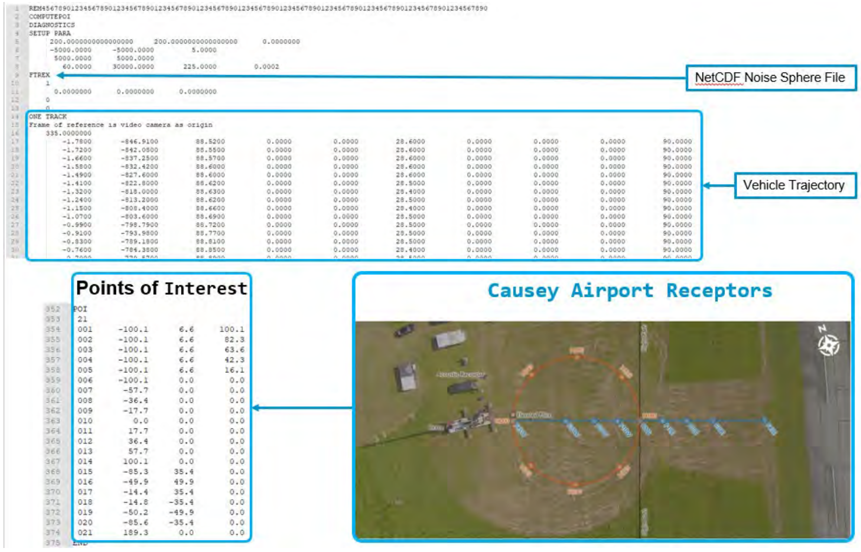
$$R_A(f) = \frac{12193^2 f^4}{(f^2+20.6^2)\sqrt{(f^2+107.7^2)(f^2+737.9^2)}(f^2+12194^2)} \qquad \text{(Eq. 8)}$$

The weighting factor becomes:

$$A(f) = 20\log_{10}\left(R_A(f)\right) - 20R_A(1000) \qquad \text{(Eq. 9)}$$

AAM Propagation

To compare the AAM and ASCENT Project 094 tool results at the receptor points of interest, “Detailed Time History at Points of Interest (poitime)” study was adapted. This requires the “COMPUTEPOI” input file, with the Points of Interest, and Vehicle Trajectory NetCDF Noise Sphere File clearly defined. Figure 31 shows the “COMPUTEPOI” input file required to run the “Detailed Time History at Points of Interest (poitime).”



```
REM4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
COMPUTEPOI
DIAGNOSTICS
SETUP PARA
    200.0000000000000000    200.0000000000000000      0.0000000
    -5000.0000      -5000.0000          5.0000
     5000.0000       5000.0000
       60.0000      30000.0000        225.0000         0.0002
FTREX
     1
     0.0000000       0.0000000       0.0000000
     0
     0
ONE TRACK
Frame of reference is video camera as origin
     335.0000000
        -1.7800     -846.9100     88.5200      0.0000      0.0000     28.6000      0.0000      0.0000      0.0000     90.0000
        -1.7200     -842.0800     88.5500      0.0000      0.0000     28.6000      0.0000      0.0000      0.0000     90.0000
        -1.6600     -837.2500     88.5700      0.0000      0.0000     28.6000      0.0000      0.0000      0.0000     90.0000
        -1.5800     -832.4200     88.6000      0.0000      0.0000     28.6000      0.0000      0.0000      0.0000     90.0000
        -1.4900     -827.6000     88.6000      0.0000      0.0000     28.6000      0.0000      0.0000      0.0000     90.0000
        -1.4100     -822.8000     88.6200      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -1.3200     -818.0000     88.6300      0.0000      0.0000     28.4000      0.0000      0.0000      0.0000     90.0000
        -1.2400     -813.2000     88.6200      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -1.1500     -808.4000     88.6600      0.0000      0.0000     28.4000      0.0000      0.0000      0.0000     90.0000
        -1.0700     -803.6000     88.6900      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -0.9900     -798.7900     88.7200      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -0.9100     -793.9800     88.7700      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -0.8300     -789.1800     88.8100      0.0000      0.0000     28.5000      0.0000      0.0000      0.0000     90.0000
        -0.7600     -784.3800     88.8500      0.0000      0.0000     28.4000      0.0000      0.0000      0.0000     90.0000
```

```
POI
 21
 001     -100.1      6.6    100.1
 002     -100.1      6.6     82.3
 003     -100.1      6.6     63.6
 004     -100.1      6.6     42.3
 005     -100.1      6.6     16.1
 006     -100.1      0.0      0.0
 007      -57.7      0.0      0.0
 008      -36.4      0.0      0.0
 009      -17.7      0.0      0.0
 010        0.0      0.0      0.0
 011       17.7      0.0      0.0
 012       36.4      0.0      0.0
 013       57.7      0.0      0.0
 014      100.1      0.0      0.0
 015      -85.3     35.4      0.0
 016      -49.9     49.9      0.0
 017      -14.4     35.4      0.0
 018      -14.8    -35.4      0.0
 019      -50.2    -49.9      0.0
 020      -85.6    -35.4      0.0
 021      189.3      0.0      0.0
END
```

Figure 31. Required data to run the detailed time history at the points of interest.

Comparison Between the ASCENT Project 094 Noise Analysis Tool and AAM Results

Our first approach started by comparing total SPL (dBA) for a single Causey run (94). Receiver #5 was randomly chosen for initial comparison. Initially, the raw measurements for the Causey Airport measurements were included to determine how the ASCENT Project 094 tool and AAM results compare to the actual data. Figure 32 displays the discrepancies between SPL propagated using the ASCENT Project 094 tool and AAM especially at the maximum SPL.

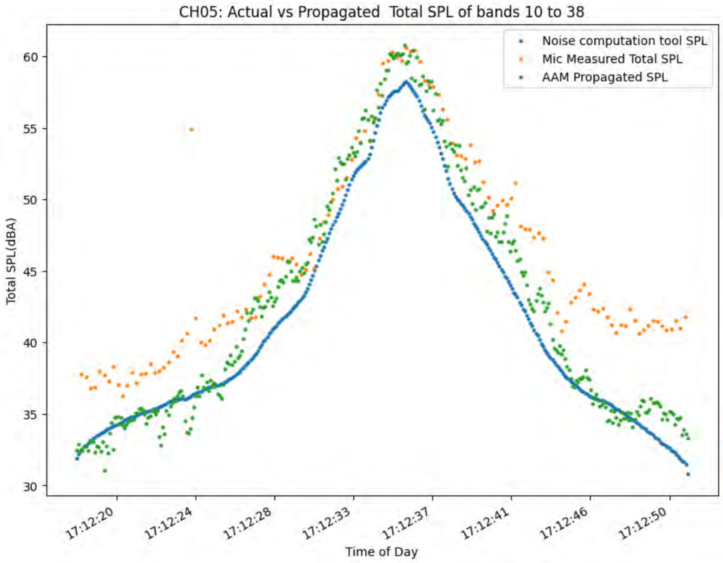


Figure 32. Discrepancies in propagated SPL between the A94 tool and AAM.

ASCENT Project 094 Tool Modifications to Match AAM Results

Studies were conducted to investigate the cause of the discrepancies between the ASCENT Project 094 tool and AAM. There appears to be a shift in time between the tool results, hence sound propagation delay was implemented. Also, as stated, the ASCENT Project 094 tools underestimates the maximum SPL in the middle, as such, reflected sound was added.

The receiver effectively receives sound from two sources at any given time: the “direct” noise and the “reflected noise” as depicted in Figure 33. Also accounted for is the difference in time.

- t_{direct} comes from trajectory information.
- $t_{received}$ comes from the sound propagation delay.
- $t_{reflected}$ computed such that reflected sound arrives to the receiver at $t_{received}$.

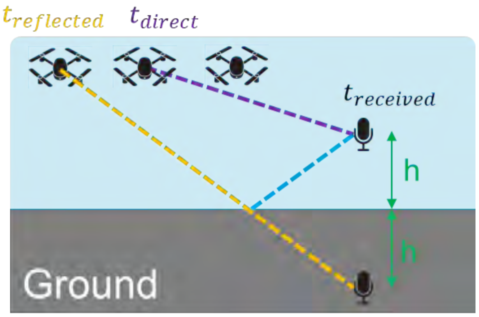


Figure 33. Representation of direct and reflected noise contributions to the receiver.

Notional View of the Reflected Sound Implementation

The SPL from the ASCENT Project 094 tool matches the AAM well in the region of highest SPL. It can be observed in Figure 34 that there is a good match in the 10dB down region which is used to compute integrated noise metrics. However, there is some overestimation of the sound level at higher distances between the vehicle and receiver. This is likely due to other phenomena not modeled in the tool. Note that the goal is not to perfectly match the AAM, but to have reasonable accuracy. Since this is for a single receptor, the next step will be to extend the validation to more Causey runs and receivers, to get a more global view of the tool’s performance.

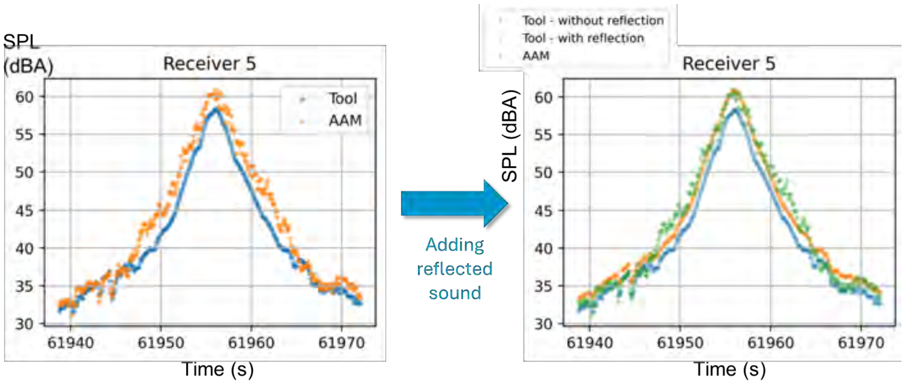


Figure 34. Impact of reflected sound on SPL prediction: A94 tool vs. AAM at receiver #5.

Global View of the Tool on All Receptor Points

Recalling that there are 14 receptor points, 5 receptors at an altitude and 9 receptors on the ground. Figure 35 shows the plots for 3 randomly chosen receptors on the crane (first row) and 3 ground receptors. The results for elevated (Crane) receptors better match the AAM results as compared to ground receptors (non-elevated receptors).

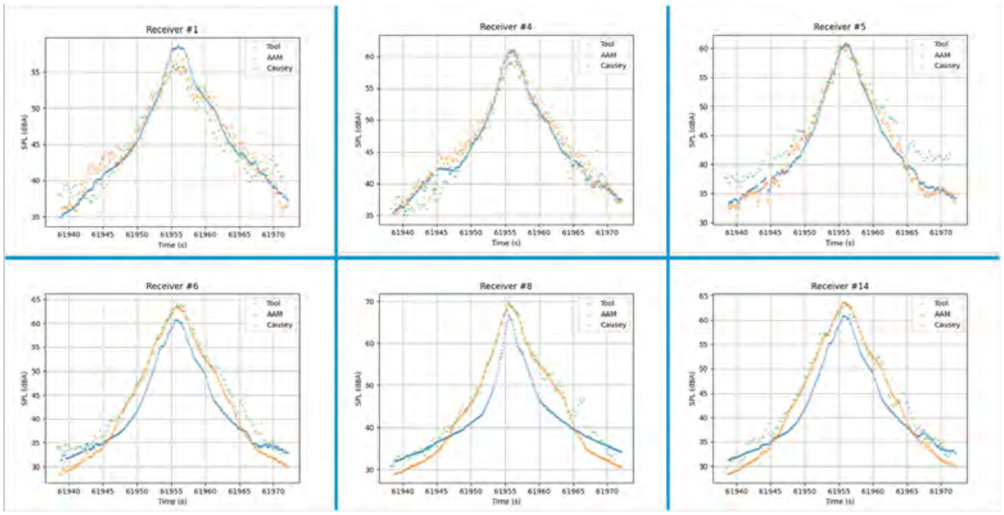
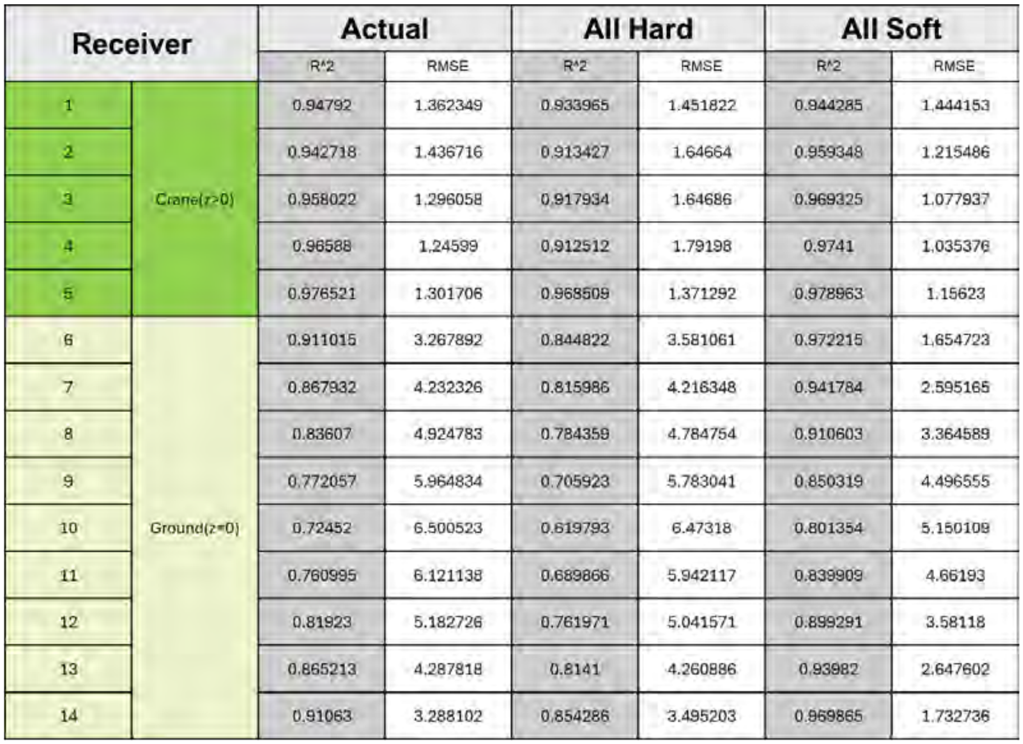


Figure 35. Results on all elevated receptors vs. ground receptors.

To investigate other phenomena not modelled, we suspected that it could be due to the terrain data implementation (e.g., hard impedance vs soft impedance). AAM terrain settings were varied from the original terrain data to ALL SOFT and ALL HARD impedances. The results were compared with the ASCENT Project 094 tool results to see which one matches better. Comparison for all the microphones are shown in Table 2. R^2 and root mean squared error (RMSE) were utilized to determine how the SPL results vary from each other.

Table 2. R^2 and RMSE of validation cases at different terrain settings.



Receiver		Actual		All Hard		All Soft	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
1	Crane(z>0)	0.94792	1.362349	0.933965	1.451822	0.944285	1.444153
2		0.942718	1.436716	0.913427	1.64664	0.959348	1.215486
3		0.958022	1.296058	0.917934	1.64686	0.969325	1.077937
4		0.96588	1.24599	0.912512	1.79198	0.9741	1.035376
5		0.976521	1.301706	0.968509	1.371292	0.978963	1.15623
6	Ground(z=0)	0.911015	3.267892	0.844822	3.581061	0.972215	1.654723
7		0.867932	4.232326	0.815986	4.216348	0.941784	2.595165
8		0.83607	4.924783	0.784359	4.784754	0.910603	3.364589
9		0.772057	5.964834	0.705923	5.783041	0.850319	4.496555
10		0.72452	6.500523	0.619793	6.47318	0.801354	5.150109
11		0.760995	6.121138	0.689866	5.942117	0.839909	4.66193
12		0.81923	5.182726	0.761971	5.041571	0.899291	3.58118
13		0.865213	4.287818	0.8141	4.260886	0.93982	2.647602
14		0.91063	3.288102	0.854286	3.495203	0.969865	1.732736

Comparing results obtained when running the AAM and ASCENT Project 094 tool with (1) the actual terrain data file, (2) all hard, and (3) all soft terrain data, shows that the AAM results with all soft terrain data appear to match the ASCENT Project 094 tool results better than compared to the actual and all hard impedance terrain data. Results are worse for all hard impedances which may suggest that the ASCENT Project 094 tool is not implemented to consider hard impedance phenomenon. The same observations were observed on other Causey runs (91, #340 and #342) as depicted in Figure 36.

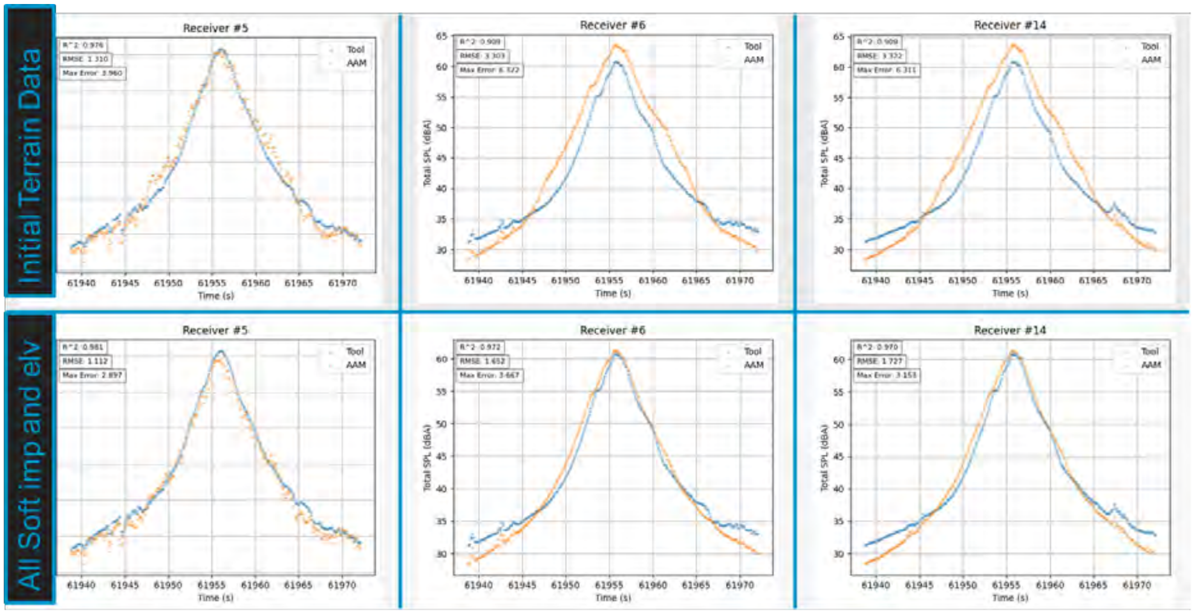


Figure 36. Graphical representation of the previous results.

Single-Event Metrics

$\boldsymbol{R^2}$, $\boldsymbol{RMSE}$ are not proper metrics for 10dB down comparison because the time/points events do not match. As such, the three single event metrics were implemented.

- $\boldsymbol{L_{AE}}$: Sound exposure level (SEL)
  - L_{AE} is generally preferable because it accounts for the duration of the event as well as its intensity.
  - L_{AE} measurements are more susceptible to interference from background noise.
  - Sometime difficult to implement because L_{AE} usually exceeds L_{max} numerically, typically by around 10dB for the same event.
  - $\boldsymbol{L_{AE} = 10\log_{10}\left(\frac{1}{t_0}\int_{t_1}^{t_2} 10^{\frac{L(t)}{10}}\,dt\right)}$, t_1 and t_2 are the start and end times.
- $\boldsymbol{L_{max}}$: Max A-weighted Sound level
- $\boldsymbol{L_{EPN}}$: Effective Perceived Noise Level (EPNL)
  - metric for aircraft noise certification limits laid down by International Civil Aviation Organization.
  - $\boldsymbol{L_{EPN} = L_{max} + 10\log_{10}\left(\frac{t_e}{t_0}\right)}$, t_0 is a reference time, usually 1 s.

Figure 37 shows the heatmap for the difference between the ASCENT Project 094 tool and AAM results across receivers and metrics. Note that, on the one hand, negative values mean that the value obtained with the tool is smaller than the value obtained with the AAM. In other words, the ASCENT Project 094 tool underestimates the noise level compared to the AAM. On the other hand, positive values indicate that the ASCENT Project 094 tool overestimates the noise level compared to AAM. Metrics were also computed for 10dB down. As shown previously, results almost match for non-elevated (crane) receptors, where the difference between the ASCENT Project 094 tool and AAM results is the largest for on-ground receptors (numbered 8, 9, and 10).

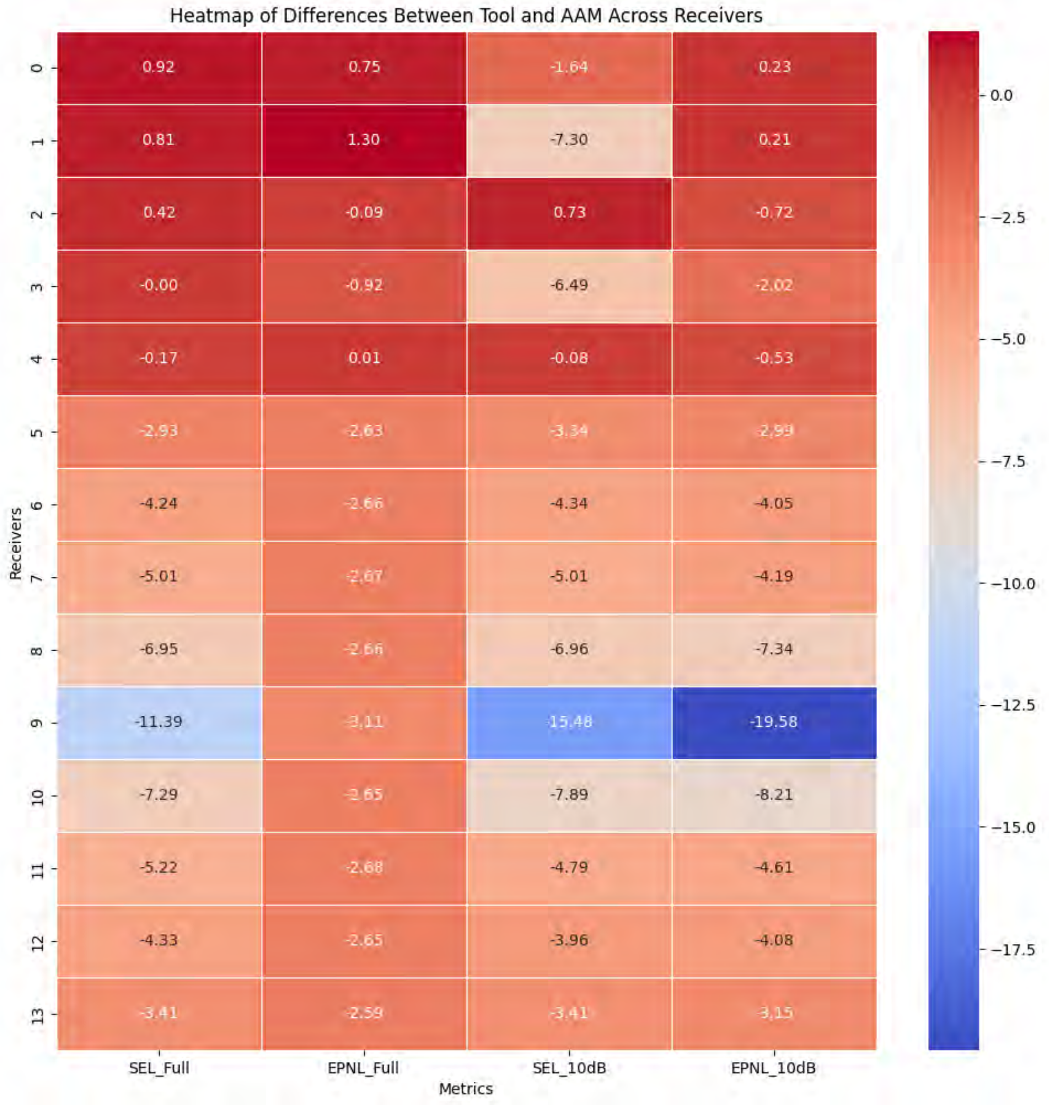


Figure 37. Heatmap for the difference between the A94 tool and AAM results across receivers.

Subtask 5.b – Validation of the Approach for Leveraging the NASA AEDT UAM Study

1. *Objectives*

The objective of this subtask is to validate the in-house noise analysis tool (ASCENT Project 094 approach) using a NASA UAM study conducted using AEDT (Rizzi and Rafaelof, 2023). Given the unique operational and acoustic characteristics of UAM vehicles, the study aims to overcome limitations in existing AEDT frameworks by generating noise-power-distance (NPD) data based on advanced simulation tools like NASA's Aircraft Noise Prediction Program (ANOPP2). The research focuses on segmenting point-to-point UAM operations into distinct departure, overflight, and approach phases, comparing noise exposure levels for fixed-wing, helicopter, and hybrid modeling approaches. The input data provided by the NASA UAM study group, along with the data generated by the AEDT, are utilized to replicate the study within the ASCENT Project 094 tool. This implementation serves as a critical component of the validation study, ensuring consistency and accuracy in noise modeling methodologies.

2. Approach

The validation process comprises two primary steps. The first step involves replicating the NASA UAM study within the FAA's AEDT, utilizing the input data provided by the NASA acoustics team. The output generated by the AEDT plays a crucial role in the second step of the validation process. In this step, the NASA UAM study is implemented within the ASCENT Project 094 tool, leveraging both the input data from NASA and the processed data from the AEDT. This phase requires extensive parsing and integration of multiple data files into the ASCENT Project 094 environment to ensure accurate and reliable implementation.

Replication of NASA UAM Study in the AEDT

The first phase of the validation study was conducted using the AEDT to ensure the accurate replication of results produced by the NASA acoustics group with the same input data. The input data, provided by the NASA acoustics team, consisted of CSV and structured query language (SQL) files, including information on receptor locations, vertiports, flight profiles (e.g., fixed-wing [FW] and helicopter), operations, and annualization settings for FW, helicopter, and hybrid models. These data files were imported into the AEDT using SQL scripts that facilitated the integration into specific modules, streamlining the replication process, Figure 38 compares the results from the NASA UAM study and replicated data. Additionally, terrain files for the corresponding UAM study location were loaded into the AEDT prior to running the simulations. All tasks specified in the input files were executed within the AEDT, and a comparative analysis was performed to verify the accuracy of the replication process and ensure consistency with the original UAM study results as shown in Figure 39.

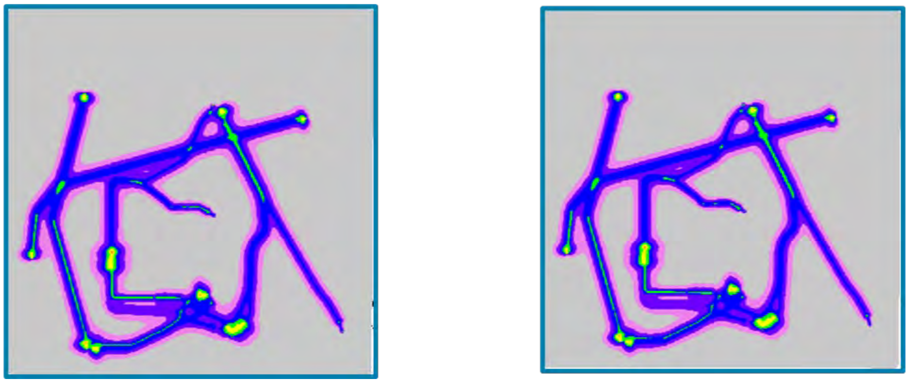


Figure 38. Results of NASA UAM study (left) and replicated UAM study (right).

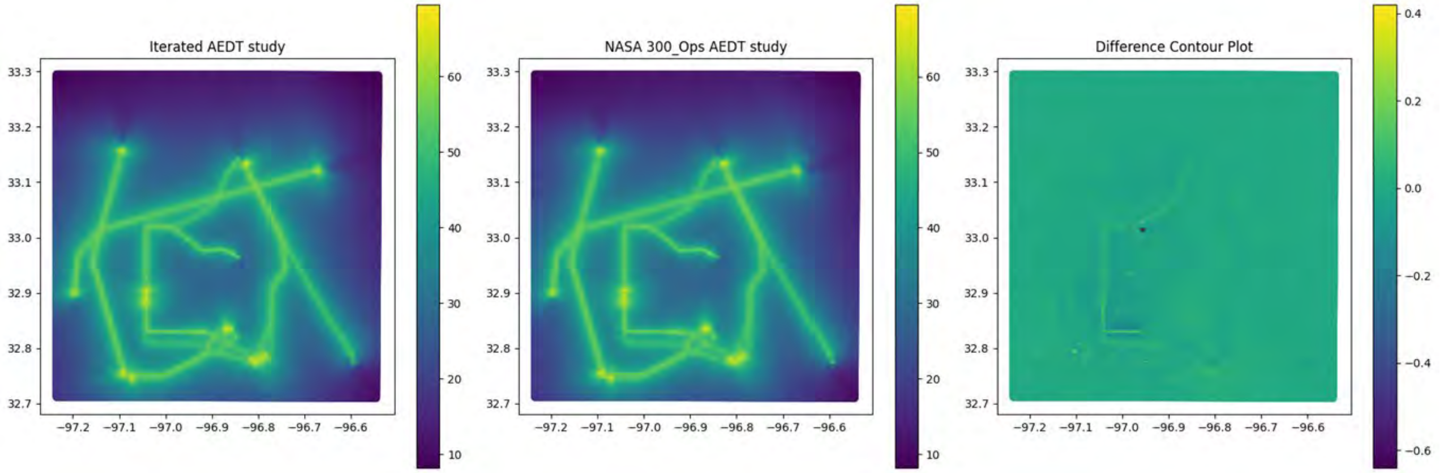


Figure 39. Results of the comparison study between NASA UAM study and replicated study.

Implementation of the AEDT Study in A94 Tool

The second phase of the validation study focuses on extracting trajectory data from the replicated UAM study in the AEDT, parsing the data files, and running noise simulations in the ASCENT-Project 094 tool. This phase adopts the noise sphere approach, as utilized in the Causey UAS validation study, in place of the traditional NPD approach. The noise spheres for FW and quadrotor configurations were provided by the NASA acoustics team and were specifically tailored for velocity and climb angle combinations corresponding to the vehicle profiles in the UAM study. Using the trajectory data generated in the AEDT, the noise levels at the receivers (virtual grid points from the AEDT) were calculated within the ASCENT Project 094 tool as part of the validation process. The implementation required substantial data extraction from the AEDT, parsing of the files for integration into the ASCENT Project 094 environment, computation of noise levels for various operations, and a comparative analysis between the AEDT results and outputs from the ASCENT Project 094 tool. The detailed process is depicted in Figure 40.

- **Loading Vehicle Locations/Source Points:** The vehicle coordinates along the trajectory were imported.
- **Loading Receiver Locations:** The coordinates of the points where noise levels were measured were defined.
- **Loading Noise Spheres:** Preconfigured noise spheres were loaded for specific velocity and climb angle combinations.
- **Noise Calculation for Each Vehicle Location and Receiver Point:**
  - o **Velocity and Climb Angle Determination:** For each trajectory point, velocity and climb angle were computed.
  - o **Noise Sphere Selection:** The nearest noise sphere corresponding to the computed velocity and climb angle was identified.
  - o **Distance and Angle Computation:** The distances and angles (polar, θ, and azimuth, φ) between source and receiver were calculated.
  - o **Source Noise Retrieval:** Noise values were interpolated from the noise sphere data based on θ and φ.
  - o **Noise Propagation:** The source noise was propagated to the receptor locations.

Detailed explanations of each step and its sub-processes are provided in the Causey UAS validation study (Subtask 5.a). This comprehensive implementation ensures consistency and accuracy of the ASCENT Project 094 tool in replicating and validating the results of the UAM noise study.

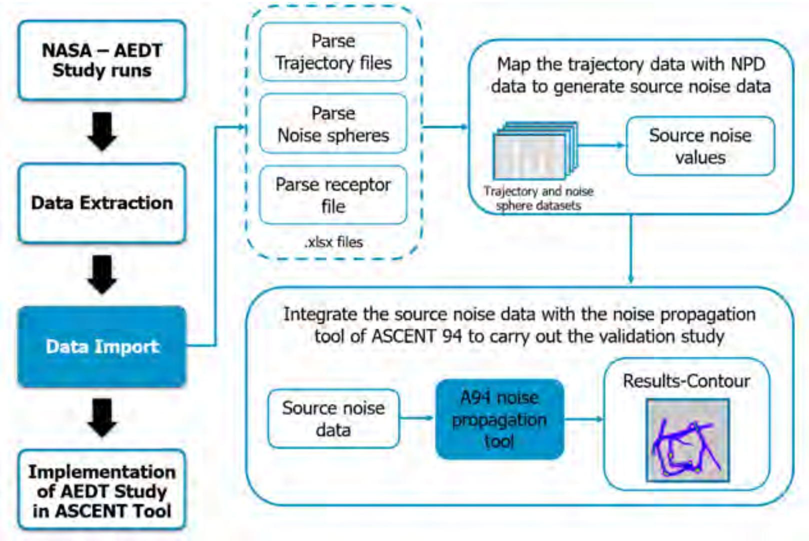

NASA – AEDT Study runs
Data Extraction
Data Import
Implementation of AEDT Study in ASCENT Tool
Parse Trajectory files
Parse Noise spheres
Parse receptor file
.xlsx files
Map the trajectory data with NPD data to generate source noise data
Trajectory and noise sphere datasets
Source noise values
Integrate the source noise data with the noise propagation tool of ASCENT 94 to carry out the validation study
Source noise data
A94 noise propagation tool
Results-Contour

Figure 40. Detailed validation process using Causey Airport measurements. A94: ASCENT Project 094.

Implementation and Results

Following the formatting of input data from the NASA UAM study and the extraction of receptor locations and trajectory data for all operations from the AEDT, the parsed data files were imported into the ASCENT Project 094 tool to calculate noise levels at the receptor locations. To initiate the validation process, a single-event analysis was conducted as a test case, focusing on the trajectory from the KADD airport to the KDF5 airport with helicopter annualization. Subsequently, the trajectory data for all annualization settings across all operations, including both single-event and multiple-operation scenarios, were extracted and parsed.

Loading Vehicle Trajectories

The trajectory of the quad vehicle from KADD airport to KDF5 is visualized in Figure 41, illustrating the three distinct phases of flight: (1) departure, (2) overflight, and (3) approach. Each waypoint in the trajectory represents the transition between segments, marking the start of one segment and the end of another. Initially, the trajectory data extracted from the AEDT is in global positioning system (GPS) coordinates and converted into the Web Mercator coordinate system for consistency. The trajectory comprises 188 segments, corresponding to 188 waypoints, with unique timestamps and a total flight time of 809 seconds for the vehicle to complete its journey. To standardize the data for implementation, the trajectory was converted into 1-s interval segments using linear interpolation. This approach facilitated a more efficient and consistent processing of the trajectory data during the analysis.

Loading Noise spheres

The noise spheres provided by the NASA team were integrated into the ASCENT Project 094 environment using the NetCDF package. These noise spheres were accessed based on the calculated velocity and climb angle of the vehicle at specific instances, as shown in Figure 41. To ensure accuracy and avoid potential errors, the nearest value approach was employed instead of interpolation, Figure 42 compares the trajectory obtained with original and interpolated data points. This method was chosen to eliminate the risk of extrapolation caused by outliers in the trajectory data and to prevent miscalculations during transitions in hybrid models.

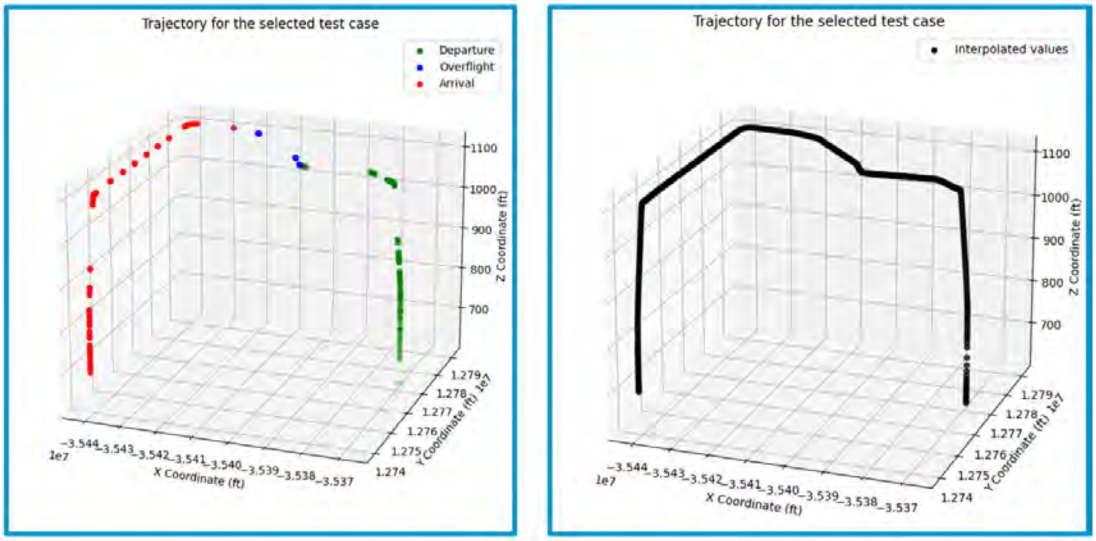


Figure 41. Comparison of the original trajectory (left) and trajectory interpolated at 1-second time intervals (right).

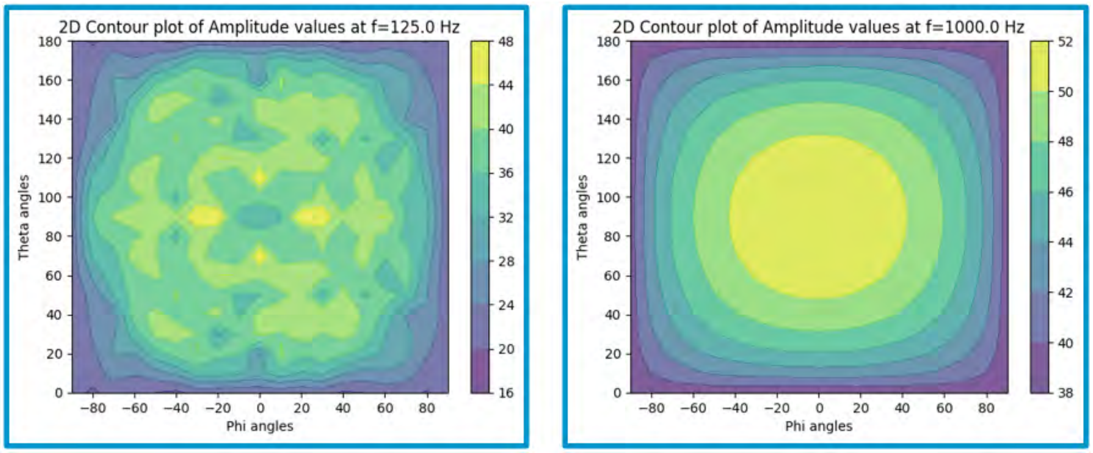
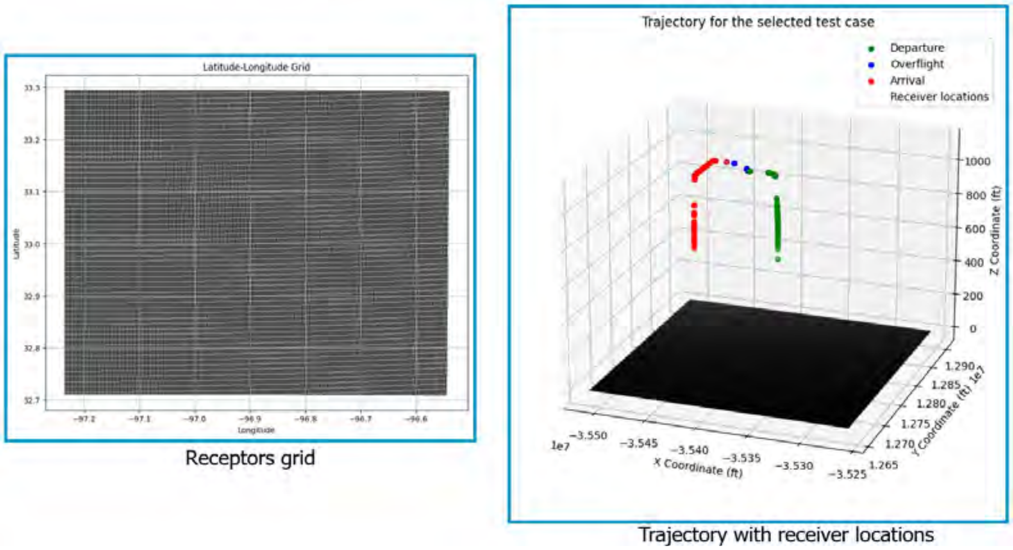


Figure 42. Noise sphere of quad vehicle (zero velocity and zero climb angle).

Loading Receptor Locations

In the AEDT, receptors are virtual grid points where the noise levels of the vehicle are calculated. The precise locations of these receptors are extracted from the AEDT in GPS coordinates. To ensure consistency in unit measurements, the GPS coordinates were converted into the Web Mercator coordinate system. A virtual grid comprising 501 × 501 points was established, with the elevation set to mean sea level (MSL), corresponding to 0 ft above ground. Figure 43 illustrate the virtual grid in relation to the trajectory of the vehicle.



Receptors grid

Trajectory with receiver locations

Figure 43. Receptors and flight trajectory

Milestones

- Set-up and configure the AAM and ART analysis tools. (Subtask 5.a)
- Received and translated Causey Airport UAS measurement data into a format that can be processed by the AAM and ART. (Subtask 5.a)
- Modified microphone measurement files in terms of the maximum number of bands to account for the limitation of ART. (Subtask 5.a)
- Decoded and executed the AAM and ART input files created for the Causey Airport UAS measurements. (Subtask 5.a)

- Replicated the NASA UAM study in the AEDT using the provided input data, and conducted a comparative analysis to evaluate the accuracy of the replicated study. (Subtask 5.b)
- Imported and parsed the trajectory data, receptor locations, and noise spheres into the ASCENT-Project 094 environment for the implementation of the AEDT study. (Subtask 5.b)
- Formatted the imported data to align with the requirements of the in-house tool, streamlining the implementation process and ensuring compatibility. (Subtask 5.b)

Major Accomplishments

- Implemented run cases 15 and 20 from the Causey Airport measurements were successfully implemented in the AAM and ART. (Subtask 5.a)
- Obtained Noise spheres for run cases 15 and 20 by using the terrain files provided in the AAM tutorial. (Subtask 5.a)
- Conducted an iterative AEDT study using the input data files provided by the NASA study group. (Subtask 5.b)
- Performed a comprehensive comparison study to validate the results to ensure accuracy. (Subtask 5.b)

Plans for Next Period

- Generate elevation and impedance files using the software BaseOps based on the experimental conditions applied in Causey Airport measurements. (Subtask 5.a)
- Select test cases from the Causey Airport measurements for validation purposes. (Subtask 5.a)
- Generate noise spheres for selected test cases using updated terrain files. (Subtask 5.a)
- For each test case, run the in-house noise analysis code using the generated noise sphere. (Subtask 5.a)
- Implement single-event analyses and multiple operations from the AEDT study in the ASCENT-Project 094 tool to further validate its functionality. (Subtask 5.b)
- Conduct a comprehensive comparison study between the results generated by the AEDT and ASCENT-Project 094 tool to assess the accuracy and reliability of the validation process. (Subtask 5.b)

References

Heutschi, K., Ott, B., Nussbaumer, T., & Wellig, P. (2021). Virtual microphone signals of flying drones (pp. 27–29). *Proceedings of the NATO STO MSG-SET-183 Specialists' Meeting on Drone Detectability: Modelling the Relevant Signature*, Wakefield, Massachusetts.

Jaller, M., & Pahwa, A. (2020). Evaluating the environmental impacts of online shopping: A behavioral and transportation approach. *Transportation Research. Part D, Transport and Environment, 80*, 102223-. https://doi.org/10.1016/j.trd.2020.102223

Hofferth, S., Flood, S.M., & Sobek, M. (2017). *American Time Use Survey Data Extract System: Version 2.6* [Machine-readable database]. In: College Park, M.U.o.M.a.M., MN: University of Minnesota. (Ed.).

Atlanta Regional Committee. (2024). *Census Data Manifest 2019-2023*. Georgia Association of Regional Commissions. Retrieved January, 2024, from https://opendata.atlantaregional.com/pages/census-data-arc

Volpe Center. (2020). *Advanced Acoustic Model (AAM) Technical Reference and User's Guide*. U.S. Department of Transportion.

Rizzi, S. A., & Rafaelof, M. (2023). On the modeling of UAM aircraft community noise in AEDT helicopter mode. In *AIAA AVIATION 2023 Forum* (p. 3363).

Scikit-Learn Developers. (2023a). *1.1. Linear models — user guide*. https://scikit-learn.org/stable/modules/linear_model.html

Scikit-Learn Developers. (2023b). 1.17. *Neural network models (supervised) — user guide*. https://scikit-learn.org/stable/modules/neural_networks_supervised.html

The Scipy Community. (2019). *Scipy.Spatial.Distance.Cdist — SciPy v1.3.2 reference guide*. https://docs.scipy.org/doc//scipy-1.3.2/reference/generated/scipy.spatial.distance.cdist.html

The Scipy Community. (2023). *Scipy.interpolate.intErp1d — SciPy v1.11.4 manual*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html