Machine Learning Methods for Estimating Propeller Source Noise Spheres

Andrew Patterson* and Naira Hovakimyan†
University of Illinois Champaign-Urbana, Champaign, IL, 61820

Kyle A. Pascioni‡ and Irene Gregory§
NASA Langley Research Center, Hampton, VA, USA, 23666

In this work, several neural network function approximations are compared for interpolating, storing, and sampling acoustic source spheres with applications to propeller noise estimation. These methods are compared using an acoustic model of the three-bladed GL-10 propeller at different flight conditions, with training data generated using NASA's ANOPP-PAS module. The source spheres used to train the networks capture the tonal propeller noise due to both the blade thickness and loading. This tonal noise prediction method allows the vehicle noise to be estimated for auralization and acoustic control. Three radial basis function neural network architectures are compared in this work. The first two networks directly estimate the parameters of the source sphere at different flight conditions but differ in the number of layers used. The third network estimates the parameters of the source sphere using a weighted combination of spherical basis functions. These networks are trained on numerically generated source spheres, with operating points given in terms of the propeller rotation rate, freestream speed, and propeller angle of attack. The performance of the neural network is determined using a validation dataset of withheld data points. This performance is quantified in terms of the approximation error, training time, and sample time. The third network, which estimates the weights of the spherical basis functions, performs the best in both average and maximum approximation errors in all cases. This network's worst case performance is 5.6 % relative difference of a model parameter associated with acoustic pressure. The direct estimation network with a single layer has the worst approximation error in all cases. Additionally, the spherically defined network has the slowest sample time at 0.05 seconds per thousand points. Both direct estimation methods produce one thousand sample points in approximately 0.001 seconds.

I. Introduction

New air vehicle configurations are being designed to improve transportation options in urban environments. These vehicles are smaller than those seen in traditional aviation and are eventually expected to be autonomous, allowing for large scale deployment [1]. Many of these newly designed vehicles employ distributed electric propulsion (DEP) systems, which use electrical motors to drive propellers. Along with airspace management and contingency management methods, new vehicle noise reduction methods are expected to overcome barriers to public acceptance [2, 3].

Noise reduction techniques are often aimed at reducing the propulsor noise of these vehicles. Many noise reduction techniques focus on physical modifications. These methods are associated with the design of the aircraft, such as changes to the blade geometry. These techniques are considered passive and do not require energy to operate. The limitation of these design changes is that they are fixed and can have an effect on the vehicle capabilities, such as reducing the produced thrust [4]. In contrast, active methods use power to change the acoustic profile of the vehicle. Active methods can be implemented by changing the flight behavior of the aircraft, such as flight speed or altitude [5]. Combining the benefits of both of these methods, phase control requires little extra power and can be used to dynamically adjust the acoustic profile of the vehicle. Phase control methods synchronize propeller blades to cause their emitted sound to destructively interfere [6, 7].

*Doctoral Student, Dept. of Mechanical Science and Engineering, AIAA Member.

†Professor, Dept. of Mechanical Science and Engineering, AIAA Fellow.

‡Research Aerospace Engineer, AIAA member.

§NASA Senior Technologist for Advanced Control Theory and Application, Dynamic Systems and Control Branch, AIAA Fellow.

Noise control methods that can dynamically adjust the sound profile of the vehicle require noise prediction methods to operate. The simplest models treat the vehicle as a point source without considering the interaction of the many propeller noise sources [8]. Recent auralization and tonal noise prediction work has focused on the acoustic interaction of the independently modeled noise sources. This more complicated model treats each of the propellers as a source sphere with a different pressure amplitude emitted in each direction, rather than a uniform sound source in all directions [9].

The construction of these source spheres is therefore central for tonal noise prediction methods that are used for auralization and acoustically aware mission planning. These source spheres encode the directional acoustic pressure of the propeller, allowing vehicle acoustics to be simulated without performing a full numerical simulation of the propeller acoustics for each vehicle configuration. Instead, the propellers are treated as independent sources that are combined to determine the sound in the far field. Though the full vehicle does not need to be simulated, the propeller acoustic profile must be provided for the vehicle operating conditions. To this end, hundreds of source spheres may be simulated to cover the flight envelope of the vehicle. Then, approximation methods are used to find the sphere values at operating points that have not been calculated. Natural neighbor and spline interpolation have been used in Refs. [6] and [10] for interpolation on a single sphere and estimating ground noise, respectively. The purpose of this work is to compare machine learning methods for interpolating *between* the source spheres to estimate noise at a new operating point. Previous related work on quantifying aircraft noise has focused on helicopters, using databases, machine learning, and semianalytical models [11, 12]. This paper will instead consider a source sphere model for small Unmanned Aerial Vehicle (UAV) propellers. The proposed method will allow a set of computed or measured source spheres to be used to analyze a large number of vehicle configurations and flight conditions. This reduction in the number of source spheres will reduce the computational and experimental burden of propeller source sphere generation for use by auralization and acoustic control methods.

This paper is structured in four additional sections. In Section II, the dataset is introduced, and the interpolation performance metrics are presented. The methods of processing and interpolating between flight conditions are presented in Section III. The performance of each of the methods is then demonstrated in Section IV. Finally, concluding remarks are provided in Section V.

II. Background

A. Dataset Generation

To demonstrate propeller source noise sphere estimation, we use a dataset generated using the method found in Ref. [6]. The propeller considered has three blades, as found on the GL-10 50% scale vehicle concept developed by NASA to combine long endurance with vertical takeoff and landing technology. The propellers are composed of AeroNaut 16x8 blades, which results in a propeller diameter of 0.4064 m. The spherical coordinate frame in which we consider the noise generated by these propellers is centered on the propeller hub, as shown in Figure 1. The propeller rotates about the z-axis, and the freestream is aligned with the positive z axis at zero degrees angle of attack. The source sphere dataset is generated using the Propeller Analysis System (PAS) module of the NASA Aircraft NOise Prediction Program (ANOPP) [13, 14] to estimate the tonal acoustic signature, at a range of emission angles, of a single propeller under different flight conditions. An emission vector is shown as a red vector in Figure 1. This vector is defined in terms of azimuth and polar angles, θ and ϕ, respectively. The tonal noise estimate is predicted using Farassat's F1A full blade formulation of the Ffowcs Williams-Hawkings equation [15]. The thickness and loading noise are the deterministic components of the tonal noise. Thickness noise is computed given the blade's geometry and motion relative to each observer on the sphere. Loading noise is estimated by projecting the surface pressure, derived from Blade Element Momentum Theory (BEMT) [14], to the farfield. The freestream velocity vector is adjusted for any nonzero propeller angle of attack, but the induced velocity is unchanged. The thickness and loading noise components are combined, and the magnitude and phase of the resulting sinusoid are stored for a set of directions around the center of the propeller. While the GL-10 is a tiltwing vehicle, we only focus on forward flight conditions, though a similar method could include the hover mode. The amplitude and phase, along with the set of emission angles, are called source spheres for each operating

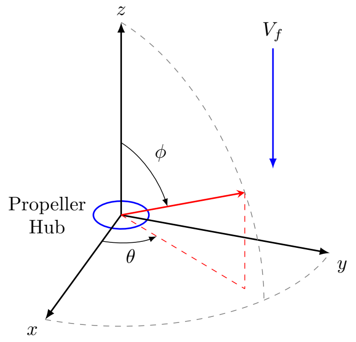


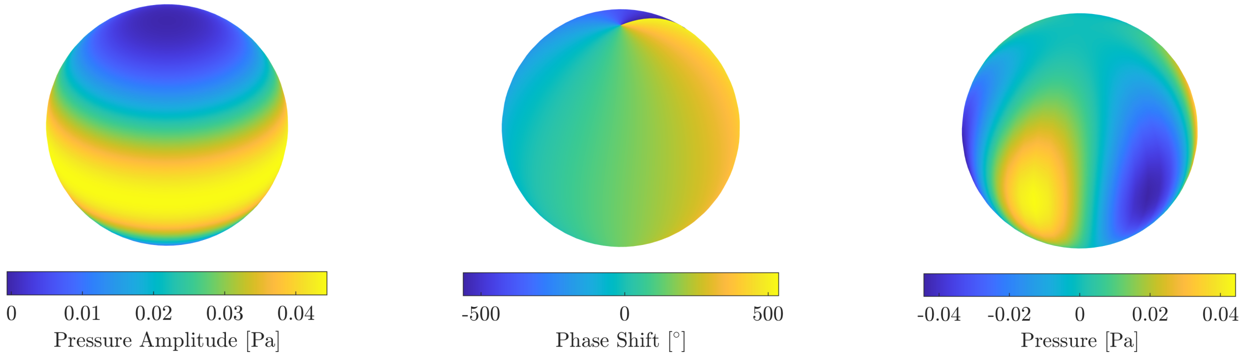
Figure 1 Spherical coordinate system showing propeller hub and freestream alignment.

condition. The radius of this sphere is taken to be twelve times the propeller radius and centered on the rotor hub. In this paper, only the blade passage frequency is considered. However, extensions to additional harmonics can either be achieved by training additional, independent neural networks or adding outputs and training data to the networks used here.

B. Dataset Description

A notional source sphere, from which samples could be drawn, is shown in Figure 2. In Figures 2a and 2b, the source sphere is represented in terms of amplitude and phase, respectively. The amplitude of noise generated is greatest near the equator, which is in the propeller plane. The amplitude is near zero at the poles. Figure 2b shows the phase of the generated noise relative to a fixed reference. The phase shift is dominated by rotation of the propeller blades and, for this three-bladed propeller, a full shaft rotation leads to three blade passages. For this reason, the phase changes 1080° on the surface of the sphere, indicating three full blade passages. Figure 2c shows a time domain representation of a sphere for a single instant in time. Here, we can see the three pressure peaks associated with the three propeller blades. As time progresses, this representation would rotate about the vertical axis. This work focuses on estimating these source spheres. Note that each flight condition would lead to a different distribution of pressures on the sphere.

The dataset is composed of approximately 400 source hemispheres taken at different vehicle conditions. Each hemisphere is sampled fewer than 200 times, with samples spaced at 15° intervals around the sphere. The propeller operating condition varies in terms of the rotation rate, freestream speed, and propeller angle of attack. Rotation rate, ω, is given in units of RPM, the freestream speed, V_f, with units of m/s, and the angle of attack, α, with units of degrees. The shaft rotation rate is varied from 3000 to 9000 RPM in increments of 500 RPM; freestream speed is varied from 15 to 30 m/s in increments of 5 m/s; and the angle of attack is varied from -8° to 8°, in increments of 2°. The distribution of operating conditions is shown in Figure 3. These operating conditions cover the advance ratios that yield positive thrust in the flight direction. Previous work, such as Ref. [9], has interpolated between samples on the surface of spheres but has not interpolated between operating conditions, which will be necessary for simulating propellers attached to a moving vehicle with changing flight conditions.



(a) Source sphere pressure amplitude.

(b) Phase shift of source sphere.

(c) Time domain source sphere pressure at fixed time.

Figure 2 Source sphere representations for ω=5000 RPM, V_f =15 m/s, and $\alpha = 0°$.

Figure 4 shows the variation in the pressure amplitude at a single emission angle. The emission direction is chosen to be $\theta = -\pi/2$ and $\phi = \pi/2$. This equatorial emission angle demonstrates the change in pressure level with respect to the freestream speed and rotation rate. The strongest relationship is between rotation rate and the pressure amplitude and scales exponentially. We can also note that for a given rotation rate, increasing the freestream speed reduces the pressure amplitude slightly due to a reduction in aerodynamic loading on the blade surface.

While Figure 4 shows the change in pressure amplitude for a single observation location, we can also consider the full sphere, with all emission angles, as shown in Figure 5. Each plot in Figure 5 shows a different flight condition. The first is a low rotation rate condition and produces the least noise, as shown by the color scale. Note that in this case, the pressure is uniform across azimuth angles for a fixed polar angle because of the aerodynamic axis of symmetry. However, when the propeller angle of attack changes, this uniformity no longer holds. In the other two plots, the pressure amplitude is increased on one side of the propeller disk and decreases on the other. We can see that this effect is a result of the propeller angle of attack but is also affected by the freestream speed, with higher speeds leading to a greater

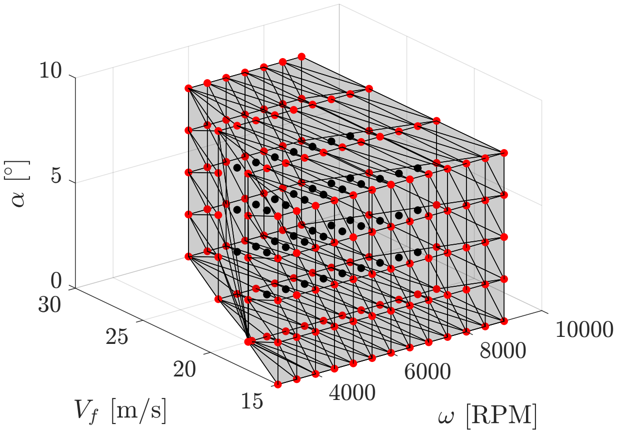


Figure 3 Flight conditions sampling space. Each point is a generated sphere. The lines and grey faces show the boundary where interpolation can occur. Red points are boundary points.

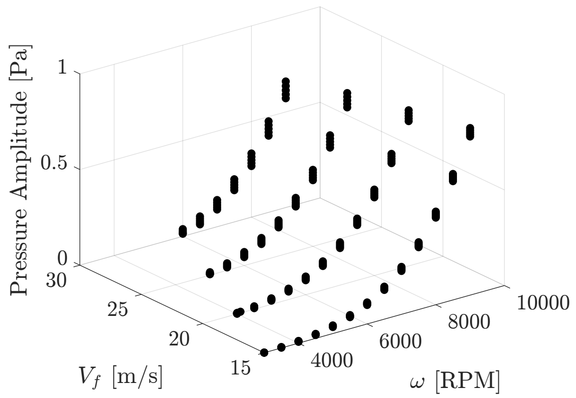


Figure 4 Scatter plot of pressure amplitudes in a single emission direction.

nonuniformity. Finally, we can note that the increase in rotation rate changes the polar angle where the maximum noise is emitted. In the first case, with a rotation rate of 5000 RPM, the maximum occurs just below the plane of the propeller, with a polar angle greater than $\pi/2$. In the 8000 RPM case, the maximum occurs at a greater polar angle. To observe the effect of a phase shift on the radiated sound, we consider the quantity $A \sin \phi$, where A is the sound pressure amplitude, and ϕ is a phase shift. This quantity can be seen in Figure 6, which shows the effect of the phase shift in each emission direction. We can see in this figure that the pressure peaks become more crescent shaped as the rotation rate increases.

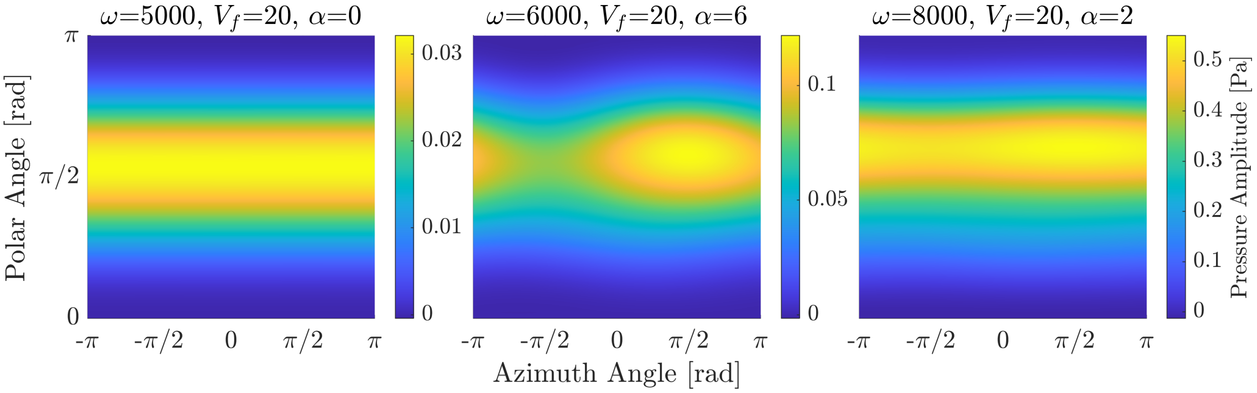


Figure 5 Source pressure amplitude for three different flight conditions. The air flows from the bottom of the plot to the top.

C. Relative Difference

The relative percent difference is used to compare a true value, y, to an estimate, $\hat{y}$, and is used to determine the effectiveness of estimation methods. This difference is given by the equation:

$$E(y, \hat{y}) = \left| \frac{y - \hat{y}}{\max(y)} \right| \cdot 100\%. \tag{1}$$

This form of the relative difference ensures that the errors can be compared between spheres. The maximum of y is used in the denominator rather than $|y|$ to prevent sensitivity to small errors when $y \approx 0$. The maximum value is taken from the set of true values given on the sphere for the given operating point.

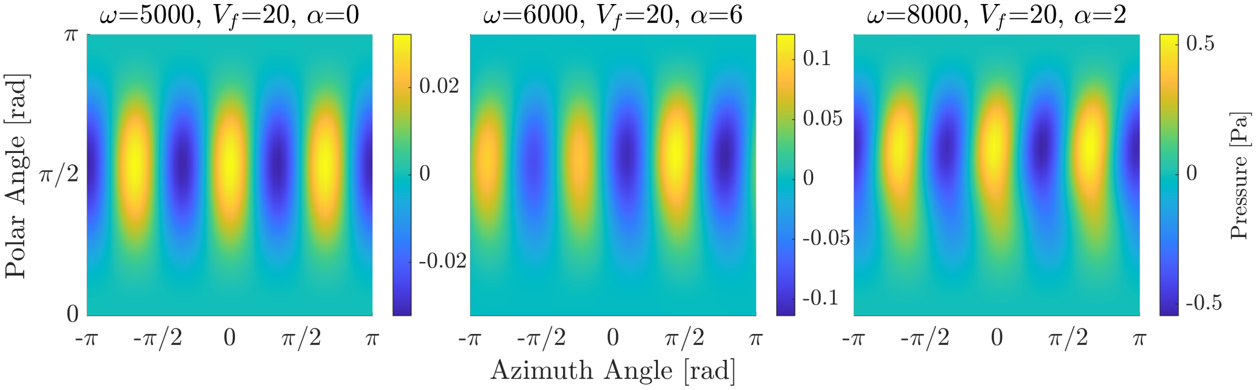


Figure 6 Effect of source phase in three different flight conditions. The air flows from the bottom of the plot to the top.

III. Method

To estimate the pressure amplitude and phase of the source sphere, we take the following steps. First, the hemispheres are combined to form complete spheres. Second, a local approximation is generated for each sphere, which is used to augment the dataset before learning the full model. Finally, the machine learning techniques are described for use on the augmented dataset.

A. Data Preprocessing

The data generated according to Section II is given as hemispheres rather than full source spheres. In Figure 7, we can further note the sparsity of the data in certain locations. We propose a data augmentation procedure to resample individual source spheres. Note that the methods used in Section II.A could be used to generate additional source sphere samples, producing a pressure amplitude and phase at any operating point and any emission angle. These calculations would take additional computational time but could be used to increase the amount of data and reduce error. However, experiments are limited during data acquisition, and the limitations may prevent collecting data at specific operating points or emission angles. For example, in both wind tunnel and flight tests, constraints on the number of experiments, measurement channel count, and measurement locations can limit the dataset. For this reason, we consider a data augmentation procedure that operates directly on a given dataset without requiring additional modeling or measurement.

To generate full spheres from hemispheres, we first note that the given hemispheres are symmetric around the polar axis when the angle of attack is 0°. When the angle of attack is nonzero, the positive angle of attack for the top hemisphere results in a negative angle of attack with respect to the lower hemisphere. By combining the appropriate hemispheres, we can construct full source spheres, as shown in Figure 7. The sphere is parameterized in terms of azimuth and polar angles. The pressure is higher for negative azimuth angles because the air inflow is angled, so one hemisphere of the propeller is experiencing additional loading.

As shown in Figure 2b, the phase source sphere is discontinuous due to the angle wrapping when $\theta = \pi$. For this reason, we reparameterize the source sphere with the following trigonometric relationship: $A\cos(\nu_1 - \nu_2) = A\cos\nu_1\cos\nu_2 + A\sin\nu_1\sin\nu_2$ and let $\nu_2 = \psi$, and $\nu_1 = \Omega t$, in which A is the sphere amplitude, ψ is the sphere phase, and Ωt is the product of the blade passage frequency and the current time. With this parameterization, the goal of sphere estimation is equivalent to estimating the quantities $\gamma = A\cos\psi$ and $\delta = A\sin\psi$. By using the sine and cosine of the phase to parameterize the sphere, the estimation procedure does not need to account for jumps due to the angle wrapping. Furthermore, by multiplying these values by the amplitude, the amplitude at the poles is appropriately attenuated. Note that δ is shown in Figure 6 for several flight conditions. The metric given by Equation (1) is used for both the γ and the δ parameters of the source spheres. The predicted parameter values at each emission angle are compared to the actual values at that emission angle.

While the given data are well distributed on a sphere, the interpolation methods considered here will compute the distances between points in terms of the Euclidean distance. The circular markers in Figure 7 indicate the emission angles where pressure and phase values are provided. The spacing between the points is no longer uniform when they are projected onto the plane. For this reason, we propose a data augmentation procedure to increase the number of points and improve the distribution of the data used for training. The augmentation is applied to both the γ and the δ

source spheres. The procedure has two key features. The first is that it is defined on a sphere. This feature ensures that the model correctly calculates the distances between points, taking into account the wrapping phenomenon that occurs on the boundary. The second feature is that we will fit each sphere individually. This restriction will improve the fit. The proposed method for interpolation on a single sphere is a regression method, with basis functions defined on a sphere.

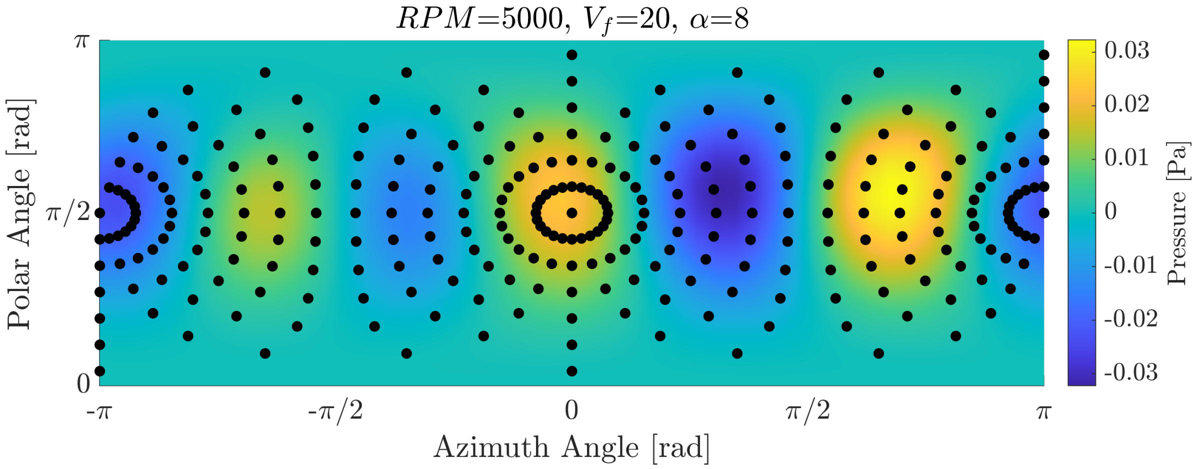


Figure 7 Samples and approximated function. For operating point of ω=5000 RPM, V_f =20 m/s and α = 8°.

To choose the basis for regression, consider points on a sphere $x, z \in \mathbb{S}^2$. The set of points $\mathbf{x}$ are the given data locations, and $\mathbf{z}$ is the set of interpolation points. Let our unknown function be denoted $f : \mathbb{S}^2 \longrightarrow \mathbb{R}$. Then our sampled output can be written $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x}$ is a vector of input points. For any kernel function, K defined on $\mathbb{S}^2 \times \mathbb{S}^2$, we can then write a best linear unbiased predictor for $\mathbf{y}^* = f(\mathbf{z})$ as $\mathbf{y}^* \approx \hat{\mathbf{y}} = K(\mathbf{z},\mathbf{x})K(\mathbf{x},\mathbf{x})^{-1}\mathbf{y}$ [16]. In this work, we choose to use the Abel-Poisson Kernel and its associated Mercer series [17] given by the equation:

$$\begin{aligned} K(x,z) &= \frac{1-\eta^2}{(1+\eta^2-2\eta x^\top z)^{3/2}}, \\ &\approx \sum_{n=0}^{M} 4\pi\eta^n \sum_{l=1}^{2n+1} Y_{n,l}(x)Y_{n,l}(z) \end{aligned} \tag{2}$$

for $\eta \in (-1, 1)$, and $Y_{n,l}$ is a spherical harmonic of degree n and order l. Notice that the matrix $K(\mathbf{x},\mathbf{x})$, required for interpolation, will grow in size with the number of data points given. For large amounts of data, inverting this matrix can lead to slow computation. The series expansion of the function K is approximate, with equality holding in the limit when $M \longrightarrow \infty$. By Theorem 12.2 in Ref [16], the truncated Mercer series provides the best M-term approximation of the kernel K. This spherically defined interpolation procedure is performed using Program 15.6 in Ref. [16], modified to use the series given in Equation (2). The program implements the solution to estimate $\hat{y}$ using a truncated series with a numerically stable Hilbert-Schmidt singular value decomposition. Across all spheres, this interpolation procedure leads to an average relative percent difference of less than 1% for the pressure amplitude, using parameter values $\eta = 0.1$ and $M = 50$. This model is used to augment the dataset, creating a uniform grid of data points for each sphere. The augmented input and output data pairs are denoted $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$. In this work, we increase the number of points per sphere to 625. For this dataset, it takes approximately 25 seconds to train the model on all source spheres. Once the model is trained, sampling the data only requires approximately 0.05 seconds per 1000 data points.

B. Neural Network Direct Approximation

To perform our function approximation task, we use a feedforward network model [18]. In this model, the neurons of the network are organized into layers, with the output of each layer being combined using an affine function and then transformed by a nonlinear function and passed through to the following layer. For an input vector, x with D elements, we have that

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(0)}, \tag{3}$$

in which j is the index of the neuron in the layer and the superscript indicates that this is the first layer of the network. The weights are denoted w. The input to the second layer is then given by $z_j = h(a_j)$, where h is a nonlinear activation function. The activation function used in this work is the squared exponential function, commonly used in function approximation applications. The function is a radial basis function given by the equation $z_j = h(a_j) = \exp\left[-(a_j)^2\right]$. The network is trained by optimizing the weights to minimize the error between the model and the data. The cost function used for training is the mean squared error between the model output and the training data. The Levenberg-Marquardt algorithm is used to train the networks used in this work [19]. By increasing the number of neurons in a layer or the number of layers combined, the number of functions that can be represented increases since the network will be combining more shifted and scaled basis functions. However, this increase leads to a larger number of weights to adjust during the training process.

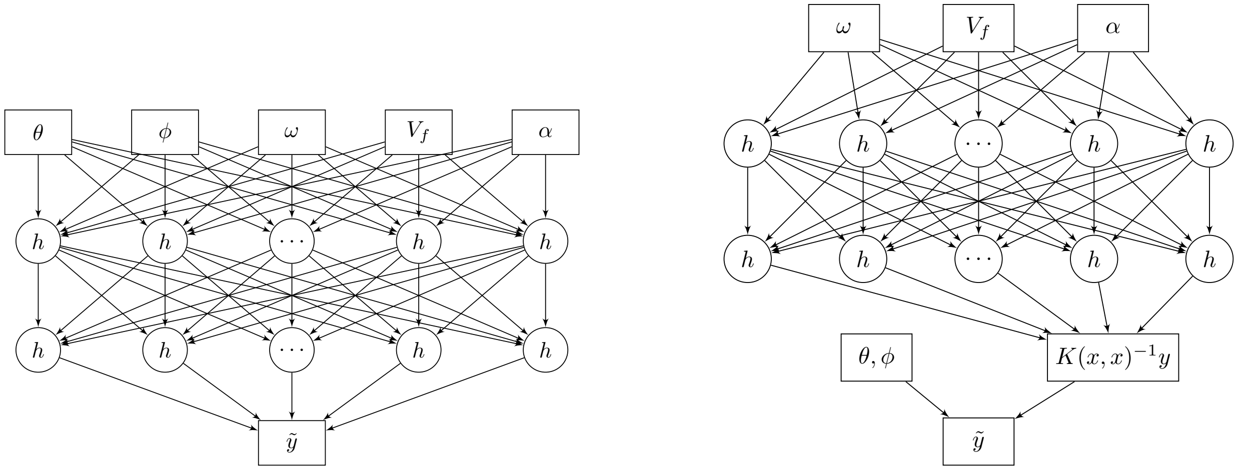
In this work, we consider two neural network models that directly estimate the quantities γ and δ of a source sphere. These networks produce a source sphere estimate for a single location on the sphere, given the operating condition. Furthermore, the outputs used for training are scaled by a factor of $c(\omega) = 1/500 \cdot \exp(\omega/1500)$ to normalize the data for training. The first network is a single layer network with 50 neurons and a linear output layer. The second architecture has two hidden layers with 13 neurons and a linear output layer. The two layer network architecture can be seen in Figure 8a. These two networks are compared to investigate the benefits of additional network layers. As described in Section 6.4 of Ref. [20], a single layer network may be able to approximate a large class of unknown functions, but the number of neurons necessary may be prohibitive. The addition of subsequent layers may reduce the total number of neurons needed in the network.

C. Spherically Defined Source Approximation

The presented neural network architecture is trained on data points given in a Euclidean coordinate system that does not fully reflect the spherical distribution of the data. This choice of coordinates leads to approximation errors near the boundaries of the embedding, where the Euclidean distance between points diverges from the true distance calculated on the sphere. In Section III.A, we considered a kernel regression method that is defined on a sphere. The method appropriately approximates the function output on the entire domain of the function. This method works well for the local fit used in data augmentation; however, scaling the regression across multiple operating conditions leads to two problems. The first is that the operating condition variables are not defined on a sphere and would need additional basis functions to be included. The second is that the weights of the regression are computed using the equation $W = K(x, x)^{-1} y$, where W is the weight vector, x is the training input, and y is the training output. This matrix inversion scales almost cubically in computation time and quadratically in memory use with the number of training samples. Rather than directly computing the weights of this regression, the spherically defined approximation uses a neural network to estimate the weights needed to predict the function output. The function output prediction is then performed using Program 15.6 in Ref. [16], which is modified to use the series given in Equation (2). The parameters of this program are the same as defined in Section III.A. The architecture for the spherically defined approximation is shown in Figure 8b.

D. Training Evaluation

To evaluate the performance of the trained model, we separate the data samples into training and validation datasets. Typically, these datasets are constructed through random assignment. However, in this work, we are concerned with interpolation between source spheres at different operating conditions rather than interpolation on the surface of a particular sphere. For this reason, it is important to not randomly assign data samples but instead assign entire source spheres to the training and validation sets. The validation spheres are randomly selected from operating conditions on the interior of the polytope shown in Figure 3. The model's accuracy is assessed on randomly selected spheres from the interior of the polytope. This process is performed five times for a 5-fold cross-validation demonstration. The cross-validation procedure is performed with respect to both the original dataset and the augmented dataset. Since the original data are sparse near the borders of the planar embedding, neither the training data nor the validation data will indicate performance in this region. To overcome this limitation, we will also perform validation checks using the augmented dataset. While the augmented dataset may have some approximation error, the approximation is performed in $\mathbb{S}^2$ and will provide a reasonable estimate in the entire domain. In addition to the validation demonstration, the model fitting time and model evaluation time are recorded for the models.



(a) Two layer network used to directly estimate the source sphere parameters.

(b) Two layer network for estimating the weights used for spherical regression.

Figure 8 Example architectures for direct output approximation and spherically defined approximation.

IV. Results

Each of the machine learning structures presented in Section III is evaluated to estimate γ and δ of the source sphere. The neural networks are implemented using the MATLAB® Deep Learning Toolbox. The networks are trained in parallel on a four core computer, with each core running at 3.4 GHz.

The performance of each network configuration is given in terms of the maximum percent relative difference across all validation sets, the average percent relative difference across all validation sets, the training time for the network, and the sample time required to evaluate the function after training. The method is described as being either direct, as described in Section III.B, or spherical, as described in Section III.C. The layers are described by their widths, with the set $\{50\}$ indicating a single layer network with 50 neurons and $\{13, 13\}$ indicating a two layer network with 13 neurons in each. The training data are described as either the given data, produced through the numerical method described in Section II.A, or as the augmented data, generated using the method in Section III.A. The percent relative differences are calculated according to Section II.C for all validation data points in the 5-fold cross-validation procedure. The training time is given in seconds, and the sample times are given in units of seconds per 1000 samples.

The performance metrics are shown for the γ estimation procedure in Table 1. The direct method performs well, with less than 3% relative difference on average, when operating only on the given data for both the single and double layer networks. However, as expected, it does not generalize well to estimate points where the given measurements are sparse. This degraded performance can be seen in Lines 2 and 4 of Table 1 where the relative difference increases when the validation data is far from the training data. The increase in error can be mitigated by training on the larger augmented dataset at the expense of computational time, as shown in Line 5, where the original performance is recovered. The spherical method has the best performance for both maximum and average case errors. However, this performance comes at the cost of sampling time, as shown in Lines 6 and 7. Similar performance can be seen in Table 2, where the δ performance is shown. Due to the nature of these quantities, the performance is expected to be similar. Again, the best performance, in terms of both average and maximum percent relative difference, is achieved by the spherical estimation procedure.

In Figures 9 and 10, error surfaces are shown for the δ estimation procedure in the validation training set. Recall that the values of γ and δ will be similar in both distribution and magnitude due to their definitions in Section III.A. In these figures, the spheres are shown in terms of azimuth and polar angles, with color indicating the percent relative difference between the true and estimated value of δ. Since these flight conditions are from the validation dataset, no point on any of these surfaces was used for training the model. In Figure 9, the performance of the two-layer direct estimation method, trained on the augmented dataset, is shown. The goal of data augmentation was to improve fitting along the boundaries of the planar representation of the spheres. We can see that no one region of the spheres appears to have a significant portion of the error. However, there is a correlation between the flight condition and the amount of error observed. The highest error flight conditions are those operating at the lowest rotation rate. This distribution of

Table 1 Table demonstrating the performance of the estimation methods applied to γ on the cross-validation datasets.

	Method	Layer Width	Training Data	Validation Data	Max %RD	Avg. %RD	Training Time [s]	Sample Time [s]
1	Direct	{50}	Given	Given	30	2.7	125	0.01
2	Direct	{50}	Given	Aug.	>100	3.9	125	0.01
3	Direct	{13,13}	Given	Given	24	2.4	40	0.01
4	Direct	{13,13}	Given	Aug.	95	3.4	40	0.01
5	Direct	{13,13}	Aug.	Aug.	23	2.5	60	0.01
6	Spherical	{25,25}	Given	Given	5.5	0.4	183	0.05
7	Spherical	{25,25}	Given	Aug.	5.6	0.2	183	0.05

Table 2 Table demonstrating the performance of the estimation methods applied to δ on the cross-validation datasets.

	Method	Layer Width	Training Data	Validation Data	Max %RD	Avg. %RD	Training Time [s]	Sample Time [s]
1	Direct	{50}	Given	Given	29	2.7	133	0.01
2	Direct	{50}	Given	Aug.	>100	3.9	133	0.01
3	Direct	{13,13}	Given	Given	19	2.5	44	0.01
4	Direct	{13,13}	Given	Aug.	>100	3.4	44	0.01
5	Direct	{13,13}	Aug.	Aug.	26	2.3	51	0.01
6	Spherical	{25,25}	Given	Given	3.5	0.3	163	0.05
7	Spherical	{25,25}	Given	Aug.	3.6	0.2	165	0.05

error occurs for two reasons. The first is due to the scale of the high speed spheres. Recall that the amplitude of the propeller noise scales exponentially with the rotation rate of the propeller. Even using the normalizing factor, $c(\omega)$, the high rotation rate spheres will drive the learning process. The second reason is due to the sensitivity of the error metric. This error metric normalizes the error, making it possible to compare performance across operating points. The percent relative difference is normalized by the maximum observed value at each flight condition. However, when the maximum observed value is small, this division can amplify small errors. In Figure 10, the performance of the spherical estimation method is shown. For this method, the errors are much smaller, less than 3%, and are concentrated around the equator of the sphere. While the direct method has uniformly spread errors, since it is tuning weights for each measurement point on the spheres, the errors in the spherical method are due to the limited number of basis functions used. Additionally, this method does not show a strong correlation between performance and propeller rotation rate, making it more applicable at lower speeds.

V. Conclusion

In this work, we presented and compared three machine learning architectures for estimating propeller source noise given a sparse set of given values. This work compares single layer networks to double layer networks. Both architectures are used to directly estimate the source noise parameters. Additionally, a double layer network is used in a spherically defined architecture to learn the weighting factors of spherical basis functions. The learned weights, when combined with the basis functions, are used to estimate source noise. These architectures are compared in terms of relative difference to a known value, training time, and sample time. The direct methods are slower to train and result in poor performance but can be sampled quickly. To improve performance, at the expense of training time, we present a method for augmenting the dataset. The spherical basis function method trains faster and yields the best performance but

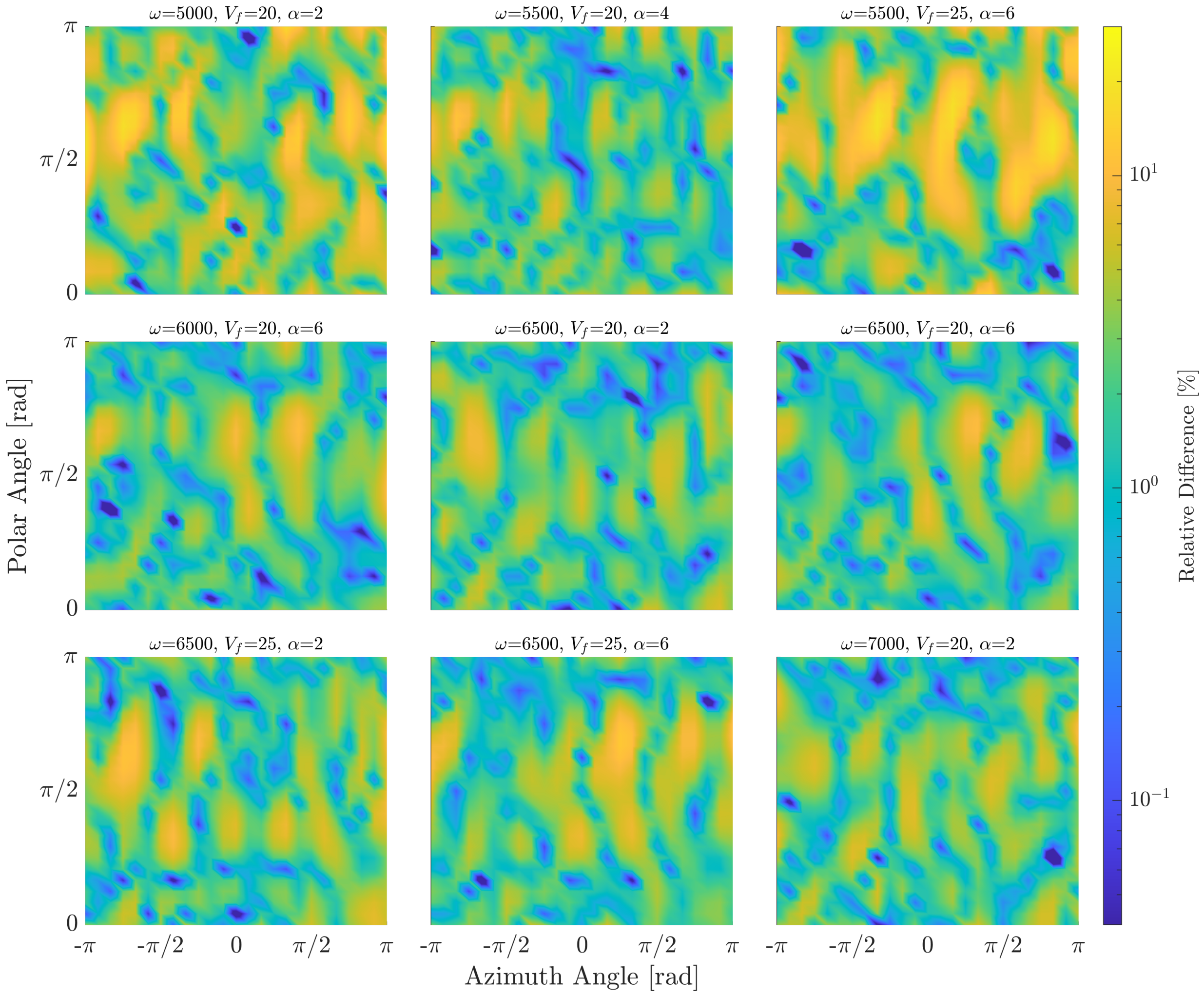


Figure 9 Error surfaces of δ at flight conditions in the validation dataset for direct estimation. Estimates generated using a $\{13, 13\}$ network, trained on the augmented dataset.

requires more time to sample. Furthermore, this function does not need the dataset to be augmented for interpolation on the sphere. Any of these methods may be used to interpolate between source spheres to make the appropriate trade-off between sampling time and accuracy for the given application. Future work will incorporate these noise models and their uncertainty into an acoustically aware UAV framework to demonstrate path planning with acoustic constraints.

Acknowledgments

This work is supported by the NASA Aeronautics Research Mission Directorate, Revolutionary Vertical Lift Technology Project.

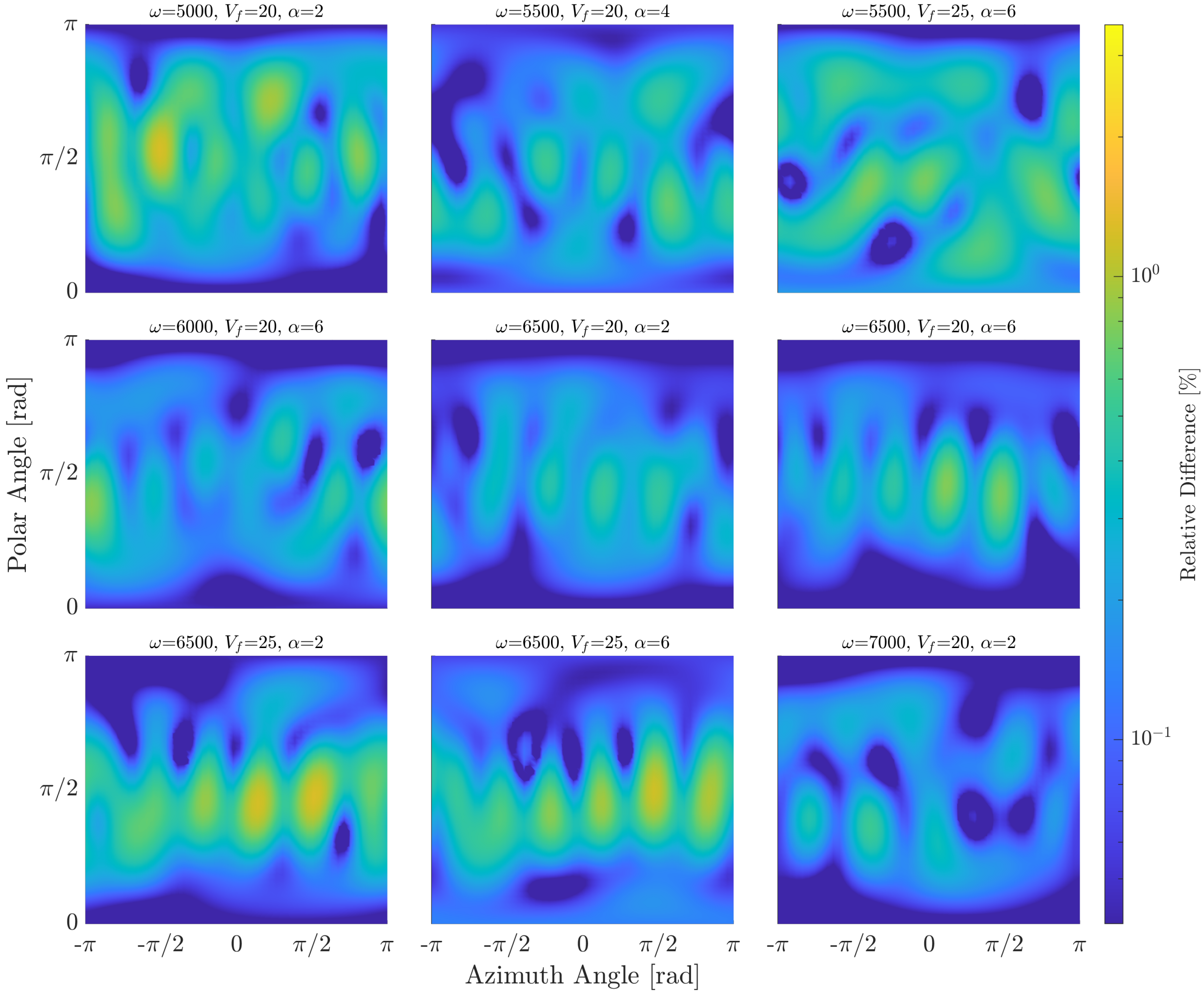


Figure 10 Error surfaces of δ at flight conditions in the validation dataset for spherical estimation, trained on the given dataset.

References

[1] "FAA Aerospace Forecast for Fiscal Years 2020-2040," Tech. rep., Federal Aviation Administration, 2020.

[2] "NASA Urban Air Mobility (UAM) Market Study," Tech. rep., NASA, 2018.

[3] Rizzi, S. A., Huff, D. L., Boyd Jr., D. D., Bent, P., Henderson, B. S., Pascioni, K. A., Sargent, D. C., Josephson, D. L., Marsan, M., He, H., and Snider, R., "Urban Air Mobility Noise: Current Practice, Gaps, and Recommendations," Tech. Rep. NASA/TP-20205007433, NASA, Hampton, Virginia, 2020.

[4] Miljković, D., "Methods for attenuation of unmanned aerial vehicle noise," *Proceedings of 41st IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, 2018, pp. 914–919.

[5] Galles, M., Schiller, N., Ackerman, K., and Newman, B., "Feedback Control of Flight Speed to Reduce Unmanned Aerial System Noise," *AIAA/CEAS Aeroacoustics Conference*, Orlando, FL, USA, 2018, AIAA-2020-0978.

[6] Pascioni, K. A., and Rizzi, S., "Tonal Noise Prediction of a Distributed Propulsion Unmanned Aerial Vehicle," *AIAA Aeroacoustics Conference*, Atlanta, GA, USA, 2019, AIAA-2019-2951.

[7] Patterson, A., Schiller, N. H., Ackerman, K. A., Gahlawat, A., Gregory, I. M., and Hovakimyan, N., "Controller Design for Propeller Phase Synchronization with Aeroacoustic Performance Metrics," *AIAA Aeroacoustics Forum*, Orlando, FL, USA, 2020, AIAA-2020-1494.

[8] Ackerman, K., and Gregory, I., "Trajectory Generation for Noise-Constrained Autonomous Flight Operations," *AIAA SciTech Forum*, Orlando, FL, USA, 2020, AIAA-2020-0978.

[9] Pascioni, K. A., Rizzi, S., and Aumann, A., "Auralization of an Unmanned Aerial Vehicle under Propeller Phase Control," *Inter-Noise*, Chicago, IL, USA, 2018, pp. 421–431.

[10] Greenwood, E., and Rau, R., "A Maneuvering Flight Noise Model for Helicopter Mission Planning," *Journal of the American Helicopter Society*, Vol. 65, 2020, pp. 57–66.

[11] Greenwood, E., and Schmitz, F., "A Parameter Identification Method for Helicopter Noise Source Identification and Physics-Based Semiempirical Modeling," *Journal of the American Helicopter Society*, Vol. 63, 2018, pp. 1–14.

[12] Greenwood, E., "Estimating Helicopter Noise Abatement Information with Machine Learning," *American Helicopter Society Annual Forum*, Phoenix, AZ, USA, 2018.

[13] Nguygen, L. C., and Kelly, J. J., "A Users Guide for the NASA ANOPP Propeller Analysis System," Tech. rep., NASA, Hampton, Virginia, 1997.

[14] Zorumski, W. E., "Aircraft Noise Prediction Program Theoretical Manual, Part 1," Tech. rep., NASA, Hampton, Virginia, 1982.

[15] Farassat, F., "Theory of Noise Generation from Moving Bodies with an Application to Helicopter Rotors," Tech. Rep. TR-R-451, NASA, 1975.

[16] Fasshauer, G., and McCourt, M., *Kernel-based Approximation Methods using Matlab*, 1^{st} ed., World Scientific, Hackensack, NJ, USA, 2016.

[17] Michel, V., *Lectures on Constructive Approximation*, 1^{st} ed., Birkhäuser, Basel, Switzerland, 2013.

[18] Bishop, C., *Pattern Recognition and Machine Learning*, Springer, New York, NY, 2006.

[19] Moré, J. J., *The Levenberg-Marquardt algorithm: Implementation and theory*, Springer, Berlin, Heidelberg, 1978.

[20] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.