NOISE-CON
2022 June 13-15
Lexington, KY

Developing and Testing a Physics Guided Machine Learning Neural Network to Predict Tonal Noise Emitted by a Propeller

Arthur D. Wiedemann[1]
Virginia Polytechnic Institute and State University
Blacksburg, Virginia, 24061, USA

Kyle A. Pascioni[2]
NASA Langley Research Center
Hampton, VA 23681, USA

Christopher Fuller[3]
Virginia Polytechnic Institute and State University
Blacksburg, Virginia, 24061, USA

ABSTRACT

Artificial neural networks offer a highly nonlinear and adaptive model for predicting complex interactions between input-output parameters. However, these networks require large training datasets, which often exceed practical considerations in modeling experimental results. To alleviate the dataset size requirement, a method known as physics guided machine learning has been applied to construct several neural networks for predicting propeller tonal noise in the time domain over a broad range of flight conditions. Three space-filling designs, namely, Latin-Hypercube, Sphere-Packing, and Grid-Space, were used to distribute points throughout the input parameter space encompassing nondimensional flight conditions and observer geometry. Each neural network's performance was validated by conditions outside of the training set and compared to the Propeller Analysis System tool from the NASA Aircraft Noise Prediction Program. Compared to the Grid-Space input design, the Latin-Hypercube and the Sphere-Packing designs provided better representations of the domain for training. Regarding the network archetype, a fully connected perceptron was found to outperform the partially connected perceptron in its ability to predict tonal noise for small datasets. The black-box nature of these neural networks was also explored to understand how the networks constructed the waveform and understand why some network designs produced better models.

[1] warthur@vt.edu

[2] kyle.a.pascioni@nasa.gov

[3] cfuller@vt.edu

1. INTRODUCTION

With the advent of electric vertical takeoff and landing (eVTOL) technology, distributed propulsion has become a viable option to the Unmanned Aerial System (UAS) and Advanced Air Mobility (AAM) vehicle designer. In many cases, the distributed propulsion system is comprised of several independently controlled lifting rotors or propellers. While the role of each propulsor may be configuration and mission profile dependent, aerodynamically generated rotor or propeller noise is still a common issue amongst eVTOL aircraft. Aside from high-fidelity numerical simulations, aeroacoustic codes such as CHARM [1], CAMRAD2 [2] coupled with WOPWOP [3] or ANOPP/ANOPP2 [4] all have varying abilities to predict distributed propulsion vehicle noise. Generally speaking, computational time increases significantly when an attempt is made to predict interactional noise sources (e.g., rotor-rotor interaction [5], blade-airframe interaction [6], and blade-wake interaction [7]) in addition to airfoil self-noise and steady tonal noise. Interactional sources are expected to be major contributors to the overall noise footprint of these vehicle classes but are physically complex and difficult to efficiently predict.

To this end, this paper investigates an alternative modeling approach employing the use of Artificial Neural Networks (ANNs) to operate as a computationally efficient noise prediction scheme. ANNs have the potential to capture highly nonlinear interactions at low expense; however, ANNs often require large datasets to train. Additionally, it is difficult to predetermine an appropriate number of runs(i.e. test points) to train a network, making experimental test matrices arduous to establish. Methods of integrating physics based knowledge into the ANN and optimizing the distribution of input conditions throughout the design space could alleviate the dataset size requirements to fit the model. With that in mind, the efficacy of training ANNs on three dataset distribution methods and two architectures designs will be addressed. As a first step, the research herein will focus on the applicability of an ANN framework limited to steady noise sources with the goal of expanding to more complex source mechanisms (e.g., interactions) in the future. Previous research has shown ANNs to be an effective method using integrated acoustic metrics [8]; however, a time domain version is preferred here to support auralization or sound quality assessments.

2. BACKGROUND

2.1. Propeller Noise

An isolated propeller produces both tonal and broadband noise. Broadband noise can be categorized into three types: blade self-noise [9], turbulence ingestion noise (TIN), and blade-wake interaction (BWI) noise [10]. Self-noise occurs when the boundary layer turbulence scatters off the airfoil, TIN is produced when a propeller ingests atmospheric turbulence, and BWI noise occurs when the wake from a previous blade impinges on the next blade. Tonal sources can be divided into unsteady noise, such as harmonic excitation by atmospheric turbulence, and steady noise, represented by thickness and periodic loading noise. Thickness noise is a function of the geometry's displacement of the fluid during flight, while the loading noise is a product of the surface pressure distribution across the blade. The broadband and unsteady tonal sources are aperiodic in nature and are excluded from further discussion in this study. For this preliminary study, only the steady thickness and loading noise will be considered.

2.2. Physics Guided Machine Learning

Across many disciplines in the scientific community, ANNs have been successfully applied to model first principle domains [11]. However, ANNs have major drawbacks such as the massive datasets required for training as well as the inability to predetermine a sufficient dataset for training. Another concern is the black-box nature of ANNs, which casts doubt on their reliability. Recently, ANNs have been designed to integrate physics based information to steer the network learning process and minimize nonphysical results. The common term used to define this approach is called Physics Guided Machine Learning (PGML) and is broadly classified into one of five categories: (i) physics guided loss function, (ii) initialization, (iii) physics guided design of architecture, (iv) residual modeling, and (v) hybrid physics ML modeling. The third method, physics guided design of architecture, has been chosen to guide the construction of the ANN in this study.

2.3. Physics Guided Design of Architecture

A classic multilayer perceptron (MLP), shown in Figure 1, is composed of several layers relating inputs to outputs. The expression in layer k relates inputs a_j^k to the outputs a_l^{k+1} with

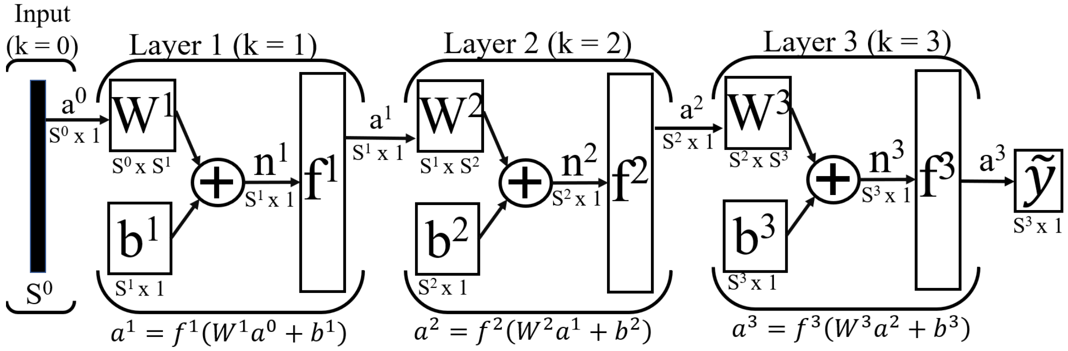

Figure 1: Classic architecture of a multilayer perceptron.

$$n_l^{k+1} = \sum_{j=0}^{S^k} W_{l,j}^{k+1} a_j^k + b_l^{k+1} \tag{1}$$

$$a_l^{k+1} = f^{k+1}(n_l^{k+1}) \tag{2}$$

The input layer, $k = 0$, supplies the initial actuators, a^0, into the first layer, $k = 1$, which is then augmented with a series of weights, W, and summed with a bias, b, to return a neuron value, n. The neurons are then transformed with an activation function, f, to produce the next set of actuator values, a^1. This process is fed forward, layer by layer, until the output, $\tilde{y}$, is returned. For the three layer example given in Figure 1, layers $k = 1$ and $k = 2$ are called hidden layers while the final layer, $k = 3$, is called the output layer. The layout from Figure 1 shows that the same operations are carried out in all the layers; however, the hidden layers use the nonlinear tansig activation function while the output layer uses a linear function. Tansig is a hyperbolic tangent sigmoid function that generates an 's' shape curve with an output range of [-1, 1], as shown in Figure 2. The subscript, l, represents the number of neurons in layer k while subscript, j, represents the connection between neurons in a previous layer to the neurons in the current layer. Since the total number of neurons in one layer may differ from the next, S^k is used to represent the total number of neurons in layer

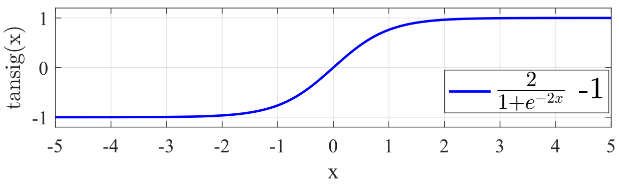

Figure 2: The tansig transfer function response.

k. It should be noted that Figure 1 shows a Fully Connected Artificial Neural Network (FCANN); however, ANNs can be designed to have targeted connections between layers. This is known as a Partially Connected Artificial Neural Network (PCANN). Both types of ANNs will be compared in this work. The training algorithm used in this study was the Levenberg-Marquardt algorithm (LMA). LMA is an optimization algorithm that adjusts the weights and biases during training to minimize the loss function. This algorithm blends the steepest descent method with the Gauss-Newton algorithm, allowing LMA to retain the quick convergence of Gauss-Newton and the stability of the steepest descent algorithm [12]. LMA iteratively updates the network's weights with Equation 3,

$$W(z+1) = W(z) - (J_a^T(z)J_a(z) + \mu I)^{-1} J_a(z)e(z), \tag{3}$$

in which z represents the current iteration, $W(z)$ is the current iteration for all the weights and biases in the network, μ is the combination coefficient, $e(z)$ is the error between the desired, y_g, and actual output, $\tilde{y}_g$, of the ANN, in Equation 4, and $J_a(z)$ represents the Jacobian.

$$e = [e_1, e_2, ..., e_G]^T = [y_1 - \tilde{y}_1, y_2 - \tilde{y}_2, ..., y_G - \tilde{y}_G]^T \tag{4}$$

The J_a is a $\mathbb{R}^{GxQ}$ with G representing the number of empirical input-output pairs and Q representing the total number of weights and biases in the entire network. The loss function used in this study was the mean square error (MSE), as shown in Equation 5.

$$MSE = \frac{1}{G}\sum_{g=1}^{G}(y_g - \tilde{y}_g)^2 \tag{5}$$

3. SIMULATION SETUP AND DATA COLLECTION

The aerodynamic pressure acting on the blade surfaces of the 2-bladed Aeronaut CAM 16x8 propeller was calculated using Blade Element Momentum Theory (BEMT). Surface pressure as well as blade kinematics were then used to calculate the acoustic pressure propagated to far field observer locations using Farassat's Formulation 1A (F1A). This process for aerodynamic simulation and acoustic propagation is included in ANOPP-PAS, which was used throughout this work. The F1A formulation is a time-domain solution to the Ffowcs Williams-Hawkings acoustic analogy and describes the acoustic emission perceived by an observer, at position x, as a surface moves at a subsonic speed. F1A has been used to reliably predict the steady tonal noise components: thickness noise, p'_T, and loading noise, p'_L [13]. ANOPP-PAS was selected as the computational tool for this work based on its previous prediction success for this propeller when compared to experimental results [14]. The propeller has a tip Mach number of $M_{tip} = \omega R_p/c_0$ where ω is the angular speed and R_p is the radius of the propeller, which operates in a freestream with velocity V as shown in Figure 3a. The undisturbed medium has a density of ρ_0 and the speed of sound is c_0. A non-dimensional

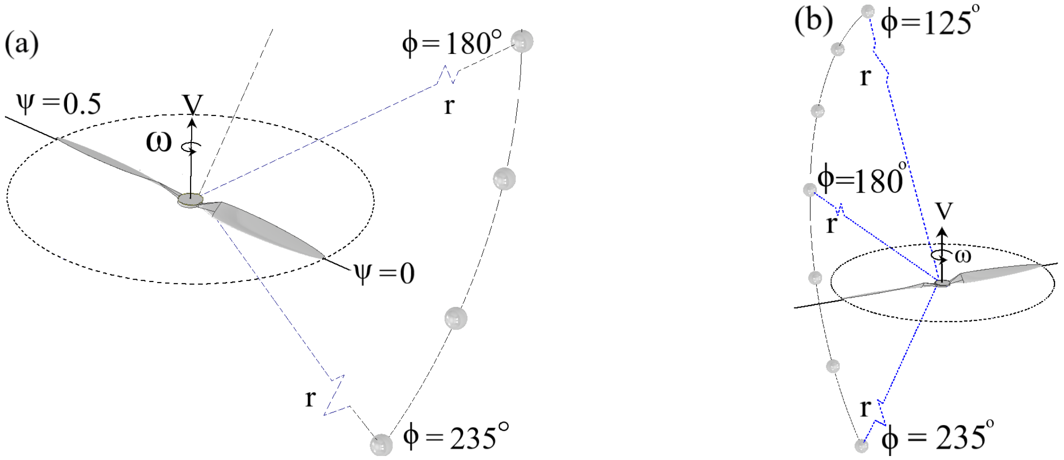

Figure 3: (a) Presents the variables and observers used in ANOPP-PAS and (b) depicts the arc of observers' azimuth observer angles with respect to the propeller.

advance ratio, $J = V/(2NR_p)$, can be calculated using these quantities where N is the revolutions per second. The ANOPP-PAS simulation was conducted over half a revolution with the first blade starting at $\psi = 0$ and ending at $\psi = 0.5$. An ANN that generates a waveform for a full revolution, $\psi = 0$ to 1, would require additional actuators to capture the signal and this would not provide any additional benefit in relating inputs to outputs. Since the propeller is operating at a steady state, the thrust, T, and the thrust coefficient, $C_T = T/(\rho_0 \pi R_p^2 M_{tip}^2 c_0^2)$, are constant during rotation. Figure 3b shows an arc of observer locations positioned at distance, r, away from the propeller's center of rotation. The arc of observers starts below the plane of rotation at $\phi = 235°$ and ends above the plane of rotation at $\phi = 125°$. The in-plane observer is located at $\phi = 180°$. All observers in Figure 3b lie on a fixed vertical plane since the acoustic emission is axisymmetric around the propeller. The distance between the noise source and the observer, r, plays a significant role in the acoustic signature, prompting its use as an additional input parameter; however, once p' has reached the far field, the amplitude of the waveform decays with r for a 1^{st} order estimate of the acoustic signature. In this simulation, all observers were placed 10 meters from the propeller. Table 1 lists the parameters and the input ranges considered in this study. All input ranges were rescaled to [-1,1] for regularization.

Machine learning is an interpolation problem and requires the training data to be representative of the domain of interest. Specifically, space-filling designs should be used to distribute the input data over the parameter space. Space-filling designs are ideal for modeling systems that are deterministic

Table 1: The parameters and design space for the ANN to model

Physical property	Symbol	Input range	Unit
Blade revolution	ψ	0 - 0.5	[-]
Tip Mach number	M_{tip}	0.18 - 0.56	[-]
Advance ratio	J	0.2 - 0.5	[-]
Thrust coefficient	C_T	0.00834 - 0.1179	[-]
Azimuth observer angle	ϕ	125 - 235	Degrees
Blade pitch	θ	[-5 - 5]	Degrees

or near-deterministic with multiple variables and complex interactions between variables [8,15]. Two of these methods, Latin-Hypercube (LH) and Sphere-Packing (SP), were chosen for this study and compared against a Grid-Space distribution. Both LH and SP distribute E number of points throughout the domain of interest with the objective to maximize the minimum distance between points. LH contains a constraint that evenly spaces the levels of each factor where SP has no such constraint. The Grid-Space design evenly distributes points, one factor at a time, over the design space. Training data sets consisting of E = 120 and E = 200 runs were made for comparing the FCANN against the PCANN for all the three distribution methods. For the Grid-Space distributions, the E = 120 dataset parameters M_{tip} and J have a spacing interval of 0.04 and 0.15 while θ is distributed non-uniformly at -5°, -1°, 1°, and 5°. The E = 200 dataset has the parameters M_{tip}, J, and θ are spaced at intervals of 0.04, 0.1, and 1°. The total number of runs, E, is related to the total number of empirical pairs, G, as a function of sampling rate with each wavelength sampled for 128 points per wavelength.

4. MACHINE LEARNING METHODOLOGY

This paper implements concepts from method (iii): physics guided design of architecture. As previously noted, the F1A formulation separates the source terms into thickness and loading noise. These individual source terms can be summed to calculate the total acoustic pressure for an observer positioned in the far field. Using this observation, two separate ANNs were created to model the thickness and loading noise, respectively, based off parameters that influence their particular waveforms. For loading, the parameters C_T, M_{tip}, θ, ϕ, and ψ are known to augment the waveform while the thickness noise waveform is influenced by J, M_{tip}, ϕ, and ψ. Blade pitch, θ, alters the motion of the blades through the medium and could be considered as a parameter in the thickness noise ANN; however, it was determined that the blade pitch contribution across the input range has a negligible influence on the thickness waveform and can be ignored for this analysis. The loading noise and thickness noise ANNs have been built with a PCANN and FCANN architecture as shown in Figure 4. Each architecture estimates the total acoustic pressure, p', for half a revolution, based on the output from the p'_L and p'_T ANNs. The architectures are nearly identical with the exception of ψ, which is introduced into the second hidden layer instead of the first layer for the PCANN. This paper proposes that, by delaying the introduction of ψ into the PCANN, the actuators in the first hidden layer will be constrained to "learn" how to relate the aerodynamic inputs with amplitude and phase

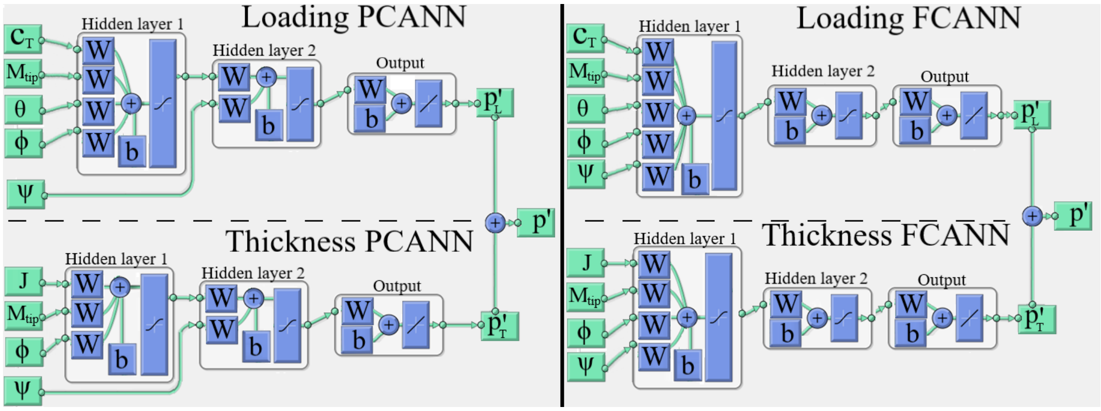

Figure 4: Architecture for the loading and thickness neural network.

of the waveform while ψ constructs the waveform based on the a^1 outputs, which are constant during the blade rotation. The FCANN has no delayed input and ψ is introduced to the first hidden layer. The p'_L ANNs have 32 neurons in each hidden layer with 1 neuron in the output layer whereas the p'_T ANNs have 28 neurons in each hidden layer with 1 neuron in the output layer.

5. RESULTS

The three distribution methods, LH, SP, and Grid-Space, were used to construct the thickness and loading ANNs. Each combination of distribution method and tonal noise component was trained eight different times with a randomized initialization for the weights and biases. After training, the performance of each ANN was evaluated over a hold-out dataset using a performance metric, $\boldsymbol{P}_E$, to determine each ANN's ability to model the acoustic waveform for new conditions. The performance metric, $\boldsymbol{P}_E$, is defined as:

$$\boldsymbol{P}_E = 100\left(1 - \frac{1}{E}\sum_{e=1}^{E}\left|\frac{RMS\left(y_e\right) - RMS\left(\tilde{y}_e\right)}{RMS\left(y_e\right)}\right|\right) \tag{6}$$

with the root-mean-square (RMS) applied to the signal. During training, the $E = 120$ and $E = 200$ datasets were distributed between three subsets: training, validation, and test into a 70%-15%-15% distribution. The training subset adjusts the weights and biases, the validation subset monitors the ANN fit to determine if early stopping is necessary to avoid over-fitting, and the testing subset is used to calculate the performance. Once an ANN was trained, a check was performed with the test data to determine if the $\boldsymbol{P}_E > 90\%$. If successful, the network was stored as one of eight networks for a given distribution/architecture. To ensure consistent comparison across datasets, an additional 20 conditions were generated in ANOPP-PAS. This unseen dataset is used to determine the $\boldsymbol{P}_E$ for Table 2 and 3. Table 2 shows the performance of the p'_T PCANN and FCANN models trained over the $E = 120$ dataset for all three distribution methods. Both architectures have "learned" to predict the acoustic waveform for new conditions when trained on the SP and LH distributions; however, the Grid-Space distribution shows a loss in prediction performance for both architecture designs.

Table 2: The best performing p'_T ANN for a given architecture and space-filling configuration.

	PCANN	FCANN
	$\mathrm{P}_{E=120}\%$	$\mathrm{P}_{E=120}\%$
Sphere-Packing (SP)	96.76	99.367
Latin-Hypercube (LH)	95.24	99.573
Grid-Space	84.02	87.63

Figure 5a demonstrates how the waveform for thickness noise, p'_T, increases with M_{tip} at $J = 0.2$ and $\phi = 180°$. For $M_{tip} < 0.3$, the first blade passing frequency is the dominating contributor to the waveform, resulting in a signal resembling a sine wave. As M_{tip} increases, thickness noise increases in amplitude and the waveform's shape changes to an impulse. Figure 5b shows, that for a given M_{tip} and J, the amplitude of the waveform is highest for the observers in the plane of rotation. Figure 5c compares three J values for $M_{tip} = 0.4$ and $\phi = 145°$. For J, the amplitude is nearly constant but the phase is delayed with larger J inputs for observers below and above the plane of rotation.

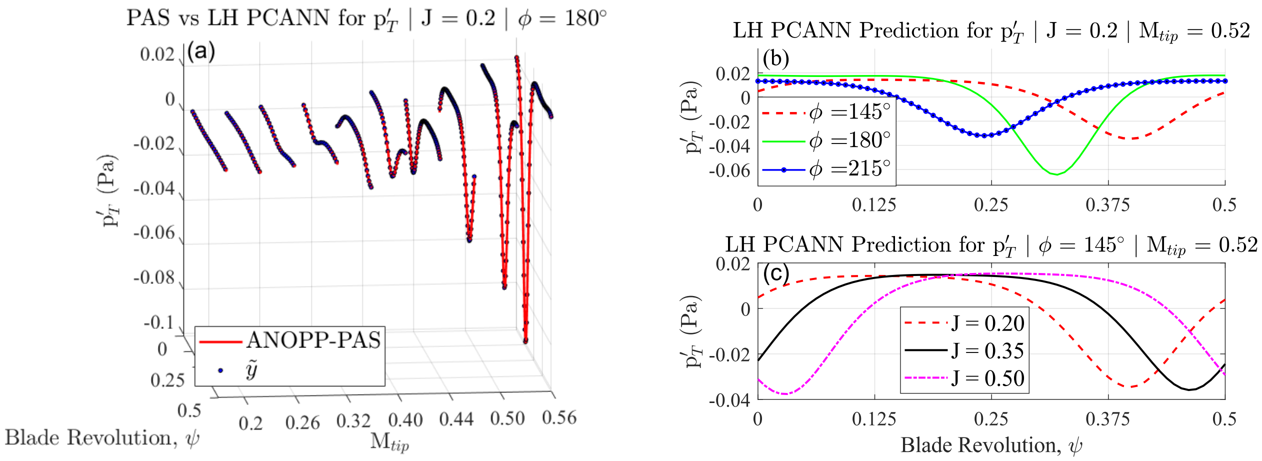

Figure 5: (a) Comparison of the LH PCANN against ANOPP-PAS for new conditions while (b) show the effect of ϕ at M_{tip} = 0.52 and J = 0.2 and (c) shows the effect of J at M_{tip} = 0.52 and ϕ = 145°.

Table 3: The best performing p'_L ANN for a given architecture and space-filling configuration.

	PCANN Loading Performance		FCANN Loading Performance	
	$P\%_{E=120}$	$P\%_{E=200}$	$P\%_{E=120}$	$P\%_{E=200}$
Sphere Packing (SP)	21.35	97.88	98.52	97.17
Latin Hypercube (LH)	15.5	72.53	66.98	95.18
Grid-Space	-2.31	70.88	-6.74	83.49

The performance for the p'_L PCANN and FCANN are shown in Table 3. The PCANNs trained over $E = 120$ conditions failed to accurately predict the p'_L waveform for new conditions. The reason for this failure will be explored in section 6. For the PCANN, only the SP distribution, trained on 200 points, was able to predict the p'_L waveform above 90% accuracy. Table 3 also shows that the ANNs trained using the SP design outperform ANNs trained using LH and Grid-Space designs. For the FCANN, the $E = 120$ training dataset saw drastic improvements in $\boldsymbol{P_E}$ for both the SP and LH design but little variation for Grid-Space. The FCANN $E = 200$ case provided the best $\boldsymbol{P_E}$ across the distribution methods. Overall, the FCANN show a considerable improvement in predicting the p'_L waveform for new conditions in comparison to the PCANN architecture. Figure 6a evaluates the prediction performance of each PCANN as a function of M_{tip} for a constant θ, ϕ, and C_T while Figure 6b evaluates the prediction performance of the FCANN on the same conditions. The ANOPP-PAS trend, plotted with gray squares, shows the predictive RMS trend for previously unseen conditions. The SP PCANN rarely deviates from the ANOPP-PAS trend but the LH and Grid-Space PCANNs fail to globally predict the influence M_{tip} has on p'_L. The FCANNs, trained on the same dataset, successfully predicted the trend across all design spaces as shown in Figure 6b. Most of the PCANNs have converged on a suboptimal set of weights and biases to model M_{tip}. It appears that by delaying the ψ input, the PCANN is constrained by the architecture. Instead of guiding the ANN towards a set of weights and biases that relate aerodynamic inputs to the amplitude and phase of the waveform, the proposed architecture merely constrains the training process. Table 3 reinforces this conclusion as PCANNs require larger datasets to attain similar $\boldsymbol{P_E}$ as the FCANNs. The reason for this will be investigated in section 6. Figure 7 shows the loading noise response for each parameter over

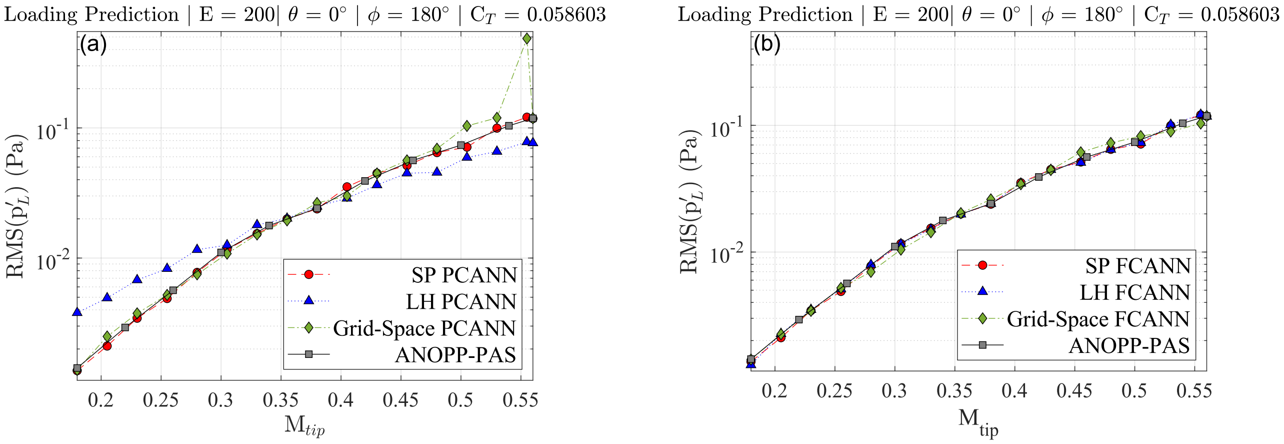

Figure 6: (a) Compares the PCANN M_{tip} trends while (b) compares the FCANN M_{tip} trends.

a portion of the input domain to highlight trends and determine the effect each parameter has on the tonal noise. A reference case, (C_T = 0.06416, M_{tip} = 0.4, θ = 0°, ϕ = 180°), is plotted in each subfigure, in cyan, to highlight changes in amplitude and waveform across plots. Figure 7a adjusts the collective pitch between -2° and 2° and shows that the waveform increases in amplitude with larger θ inputs. Figure 7b compares three observers placed above, in-plane, and below the axis of rotation. Observers positioned below the axis experience a waveform with larger amplitudes while observers above the axis see a decrease in amplitude. The nonuniform emission field is a result of the cambered airfoil. Figure 7c show that the p'_L is highly sensitive to M_{tip} as the amplitude rapidly rises with steady increments in M_{tip}. Figure 7d adjusts the C_T over a range of feasible flight conditions at θ = 0°. The p'_L is marginally influenced by C_T.

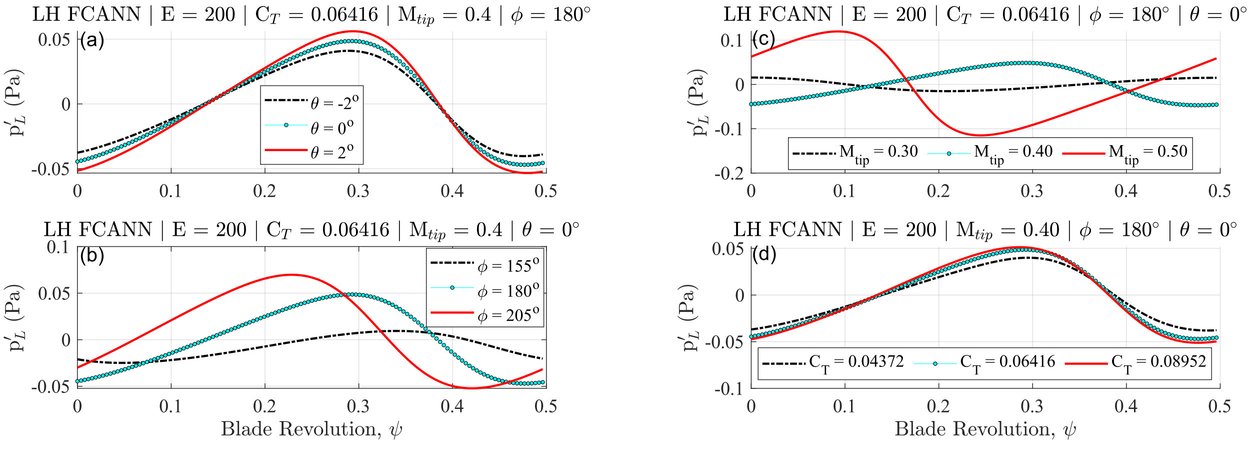

Figure 7: (a) Contrasts the p'_L waveform for three θ conditions and (b) shows the nonuniform emission field for observers above, in-plane, and below the axis of rotation. (c) Shows that p'_L is highly sensitive to changes in M_{tip} while (d) finds small changes in p'_L for C_T.

6. INTERPRETATION OF THE NEURAL NETWORK

This section delves into the architecture of the FCANN and PCANN and provides an ad hoc explanation for the ANN ability to construct the waveform. A sensitivity analysis such as Lek's Profile-Method [16] could be implemented to gain insights into the ANN. This approach adjusts one parameter at a time to contrast the parameter's sensitivity while holding all other parameters constant, usually held at the minimum, 20^{th}, 40^{th}, 60^{th}, 80^{th}, and maximum percentiles. For the waveform, evaluating percentiles of the signal does not provide clarity for the ANN. The signal could be converted to its RMS equivalent, but the parameters influence on the phase would be lost. Instead, by examining the actuators response in each ANN architecture, it is possible to determine how the waveform is constructed by the ANNs and understand why the FCANN outperforms the PCANN. The interpretation starts by examining the output and steps backwards, layer by layer, to determine how the ANN constructs the waveform. Information regarding the output actuator, a^3, is the hidden representation of the acoustic pressure. Equation 7 allows the output to shift between hidden and Pascal unit for the p'_T ANN.

$$p'_T = \frac{a^3 + 1}{2.139} - 0.00180 \tag{7}$$

As previously mentioned in the Physics Guided Design of Architecture section, the hidden layers, for both architectures, use the tansig transfer function. Figure 2 shows the tansig function is highly sensitive to inputs from -3 to 3 while inputs beyond this range are nearly constant. For this interpretation, it is useful to categorize the actuators into two groups, active and passive actuators. An actuator is deemed active if the output of a^2_j changes by 0.01 over the input range of ψ from 0 to 0.5. Actuators that fail to meet this criterion are defined as passive actuators. The advantage of classifying the actuators becomes apparent after running an example. For the LH PCANN inspection, the condition ($J = 0.29$, $M_{tip} = 0.56$, $\phi = 125^o$) was supplied to study the construction of the p'_T waveform. The PCANN prediction, in Pascals, can be seen in Figure 8a and the hidden equivalent, a^3, waveform can be seen in Figure 8b. Figure 8a and 8b have the same waveform but different numerical values because of the linear conversion between the hidden and Pascal units, from Equation 7. During the run, the 28 actuators in the second hidden layer were monitored and 12 actuators meet the active criterion. The active actuators are multiplied by their respective weights and summed to construct the waveform seen in Figure 8c. Figure 8d show that each active actuator is using a portion of their tansig output to shape the waveform while strictly adhering to the tansig output.

A new operating condition will use a different set of active actuators to model the waveform. The PCANN reserves actuators for different input regions, allowing a set of activation functions to alter the waveform at the appropriate time. An ANN trained on a representative dataset will adjust the W and b to "learn" when an actuator should be active or passive. For example, when an active actuator shifts to a passive state, it means the inputs from the previous layer, a^1, have shifted the output, $a^2 = f^2(n^2)$, towards the extremes, preventing ψ from adjusting a^2. Figure 9a shows that the FCANN can predict the p'_L waveform for an unseen condition while Figure 9b demonstrates how the individual active actuators generated the resultant waveform. Note the active actuators, a^2_7 and a^2_8, are not adhering to the tansig slope. Because the ψ input is connected to the first hidden layer in the FCANN, the $a^1(\psi)$ provides an additional set of actuators that can determine which actuators should be active as well as aid in the construction of the waveform from 0 to 0.5. This allows the FCANN to generate actuator outputs that do not strictly adhere to the transformation function. The PCANN underperforms due to the constrained architecture. This suggests that when designing the architecture for an ANN, one should take precaution when constraining the transfer functions.

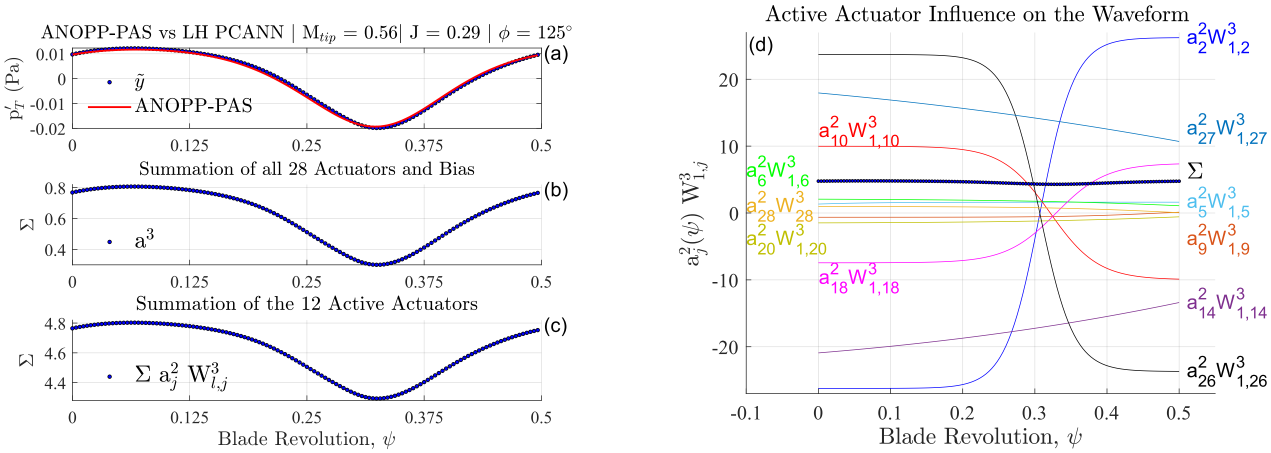

Figure 8: (a) Compares PCANN prediction to ANOPP-PAS and (b) shows the equivalent a^3 output. (c) Shows only 12 of the 28 actuators are required to construct the waveform and (d) plots them individually.

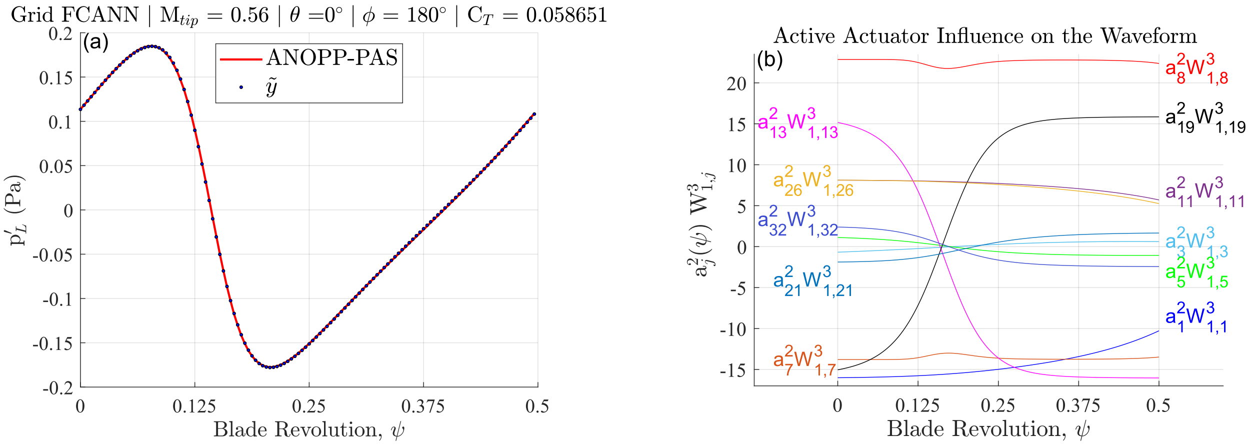

Figure 9: (a) Compares the FCANN against ANOPP-PAS and (b) shows the 11 active actuators used by the ANN to predict an equivalent waveform.

7. CONCLUSION

ANNs can operate as surrogate models, which can capture highly nonlinear interactions with minimal computation resources. However ANNs often require large datasets to alleviate this burden. A concept known as PGML was applied to structure the connections between hidden layers as a means of relating aerodynamic inputs to the acoustic tonal loading and thickness noise. The proposed architecture, a PCANN, was trained on three different distribution methods and compared against a conventional FCANN. The PCANN adequately fits the training data but underpredicts for new data in comparison to FCANN. After exploring the FCANN and PCANN hidden layers, it was determined that the PCANN had constrained the actuators output. For loading noise, the SP FCANN shows the best predictive performance with $\boldsymbol{P}_{E=120} = 98.52\%$. This was the only case where the smaller sample size, $E = 120$, could obtain a highly predictive ANN, $\boldsymbol{P}_E > 90\%$. For thickness noise, the ANN saw little benefit across architecture approach but showed similar trends for the space-filling design. The SP distribution provides an excellent representation of the domain in which to train the ANNs.

8. FUTURE WORK

This work is meant to act as a stepping stone and provide insight for building ANNs when it is difficult to generate large datasets for poorly understood acoustic phenomena. While the proposed architecture has failed to outperform the classic FCANN, this does not diminish the PGML approach. There are additional methods, not explored in this paper, which have alleviated issues dealing with the drawbacks of ANNs. These will be explored in future efforts.

REFERENCES

[1] Wachspress, D. A., Quackenbush, T. R., Boschitsch, A. H. Rotorcraft interactional aerodynamics calculations with fast vortex/fast panel methods. In *American Helicopter Society 56th annual forum*, Virginia Beach, VA, pp. 51-71, 2000.

[2] Perez, G., Brentner, K. S., Bres, G. A., Jones, H. E. A first step toward the prediction of rotorcraft maneuver noise. *Journal of the American Helicopter Society*, *50*(3), pp. 230–237, 2005.

[3] Brès, G. A. Modeling the noise of arbitrary maneuvering rotorcraft: Analysis and implementation of the PSU-WOPWOP noise prediction code, MS thesis, Dept. of Aerospace Engineering, The Pennsylvania State University, University Park, PA, May 2002.

[4] Lopes, L., Burley, C. Design of the next generation aircraft noise prediction program: ANOPP2. In *17th AIAA/CEAS Aeroacoustics Conference*, AIAA 2011-2854, Portland, ORE, June 2011.

[5] Piccinini, R., Tugnoli, M., Zanotti, A. Numerical Investigation of the Rotor-Rotor Aerodynamic Interaction for eVTOL Aircraft Configurations. *Energies*, *13*(22), pp. 5995, 2020.

[6] Zawodny, N. S., Boyd, D. D. Investigation of rotor–airframe interaction noise associated with small-scale rotary-wing unmanned aircraft systems. *Journal of the American Helicopter Society*, *65*(1), pp. 1-17, 2020.

[7] Brooks, T. F., Burley, C. L. Blade wake interaction noise for a main rotor. *Journal of the American Helicopter Society*, *49*(1), pp. 11-27, 2004.

[8] Thurman, C. S., Zawodny N. S. Aeroacoustic Characterization of Optimum Hovering Rotors using Artificial Neural Networks. In *Vertical Flight Society's 77th Annual Forum Technology Display*, 2021.

[9] Brooks, T. F., Pope, D. S., Marcolini, M. A. Airfoil self-noise and prediction. RP-1218, 1989.

[10] Zhao, Y., Shi, Y., Xu, G. Helicopter blade-vortex interaction airload and noise prediction using coupling CFD/VWM method. *Applied Sciences*, *7*(4), pp. 381, 2017.

[11] Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V. Integrating scientific knowledge with machine learning for engineering and environmental systems. *arXiv preprint arXiv:2003.04919*, 2020.

[12] Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, *2*(2), pp. 164-168, 1944.

[13] Farassat, F. Derivation of Formulations 1 and 1A of Farassat. Tech. Report, NASA TM-2007–214853, 2007.

[14] Pascioni, K., Rizzi, S. A. Tonal noise prediction of a distributed propulsion unmanned aerial vehicle. In *2018 AIAA/CEAS Aeroacoustics Conference*, 2018-2951, 2018.

[15] Santner, T. J., Williams, B. J., Notz, W. I. Space-filling designs for computer experiments. *The design and analysis of computer experiments*, pp. 145-200, Springer, New York, NY, 2018.

[16] Lek, S., Belaud, A., Baran, P., Dimopoulos, I., Delacoste, M. Role of some environmental variables in trout abundance models using neural networks. *Aquatic Living Resources*, *9*(1), pp. 23-29, 1996.