Machine Learning Based Approach to Improve Low-Fidelity Predictions for a Compound Helicopter

Gaurav Makkar
PhD Student
Center for Mobility with Vertical Lift (MOVE)
Rensselaer Polytechnic Institute
Troy, NY

Jean-Paul Reddinger
Aerospace Engineer
Vehicle Applied Research Branch
DEVCOM Army Research Laboratory
Aberdeen Proving Ground, MD

Farhan Gandhi
Redfern Professor and Director

Fotis Kopsaftopoulos
Assistant Professor

Center for Mobility with Vertical Lift (MOVE)
Rensselaer Polytechnic Institute
Troy, NY

ABSTRACT

A novel machine learning approach is introduced, with the goal of minimizing the number of points required to correct a low-fidelity model using sparse high-fidelity data. The method is applied to a low-fidelity comprehensive trim analysis of a compound helicopter with three degrees of control redundancy: main rotor speed, auxiliary thrust, and stabilator setting. The final low-fidelity correction model applies small changes to the power requirement and main rotor trim control predictions to more closely match the high-fidelity data. To reduce the computational time, and labor cost of querying the high-fidelity data, the algorithm prioritizes data acquisition by iteratively selecting the data where the error is expected to exceed the model tolerance by the greatest margin. The model is trained until the error model anticipates a scaled accuracy of 5% at the minimum power region at each flight speed, with no more than 20% error across the entire trim envelope. For the low-fidelity correction model, this tolerance is met through training with data from 89 trim states, which is a 79% reduction from the 428 trim states required to obtain comparable accuracy for a purely data-driven machine learning approach. Evaluation on a testing data set yields a rate of 95-98% of testing points actually falling within the error tolerance goals, compared to 80-93% for the purely data-driven model.

NOTATION

$\vec{x}$	State vector
$\vec{u}$	Control input vector
u, v, w	Body velocities, ft/s
p, q, r	Body angular velocities, rad/s
ϕ, θ, ψ	Body roll, pitch and yaw attitude, deg
x, y, z	Inertial positions, ft
β_0	Rotor coning, deg
β_{1s}	Lateral flap, rad
β_{1c}	Longitudinal flap, rad
β_d	Differential flapping, rad
Ω	Main rotor rotational speed, rad/s
T_{coll}	Auxiliary Thrust
δ_{stab}	Stabilator Angle
θ_0	Collective control, deg
θ_{1c}	Lateral Cyclic, deg
θ_{1s}	Longitudinal Cyclic, deg
θ_{ail}	Aileron deflection, deg
θ_{rudder}	Rudder deflection, deg
P	Total vehicle power, hp
$\mathbf{f}$	Low-fidelity model
$\mathbf{g}$	High-fidelity model
$\mathbf{h}$	Error function
$\hat{\mathbf{h}}$	Predicted error function
Y	High-fidelity predictions
$\hat{Y}$	Low-fidelity predictions
ϵ	Error tolerance
d	Normalized distance
η_d	Desired accuracy as a function of d
w	Regression coefficients
ζ_n	Slack parameter for support vector regression
b	Bias term for support vector regression
α	Lagrangian multipliers

Presented at the Vertical Flight Society's 77th Annual Forum & Technology Display, Palm Beach, Florida, USA, May 11–13, 2021.

INTRODUCTION

With the emergence of eVTOL, mid-scale UAS, and compound rotorcraft at the FVL scale, the scientific community is widening its scope beyond the established knowledge base

of large scale conventional helicopters. In the case of eVTOL and UAS, basic research is still being done to identify and understand the dominant physical behaviors such as interactional aerodynamics, handling qualities, and scaling effects (Refs. 1–5). Compound rotorcraft introduce the problem of overactuated control by adding some combination of coaxial rotors, auxiliary propulsion, and control surfaces. This increases the ability for a human or autonomous pilot to achieve targeted behaviors, such as minimum power, reduced vibration or noise, or increased maneuverability (Refs. 6–12). The ability for an aircraft designer to derive an optimal design, or for a control system to minimize a target attribute, is limited by the understanding of the vehicle physics and the capacity for their models to predict the design space and control behavior.

In design optimization, multi-fidelity optimization is used extensively for its ability to scale model fidelity as necessary. It uses a rapidly evaluated low order model to close in on the approximate area of a globally optimum design, and a computationally expensive more accurate model to refine the design parameters (Refs. 13–22). These algorithms work by identifying the area of importance and ignoring the rest of the decision space. Their benefits are limited in cases where the time and monetary cost of obtaining reliable high-fidelity data is excessive, which can be the case for manufacturing and flight testing of a rotorcraft, or when high-fidelity models such as CFD have not been confidently validated.

In the case of control of an overactuated rotorcraft, suboptimal regions of the flight envelope cannot be neglected. Validation and verification of autonomous pilots is a major focus of the AI scientific community, and depends on reliable behaviour across the flight envelope. A good autonomous pilot must maintain a minimum level of confidence across the entire flight envelope, but it benefits most from improved accuracy near minimum power, minimum noise, maximum maneuverability, and similarly desirable flight states.

The goal of this paper is to explore the ability for a low order model to be corrected by a targeted set of high-fidelity data at specific areas of interest. The high-fidelity data used to correct the low fidelity model will be selected based on modeling error, and will be as sparse as possible to minimize the cost of data collection. The final corrected model will have a minimum guaranteed level of accuracy across the entire decision space with targeted improvement in model accuracy at regions deemed important by the designer. This allows the model to be used to make rapid decisions with a known, consistent level of confidence in the results.

MODELING AND ANALYSIS

The aircraft used in the study is a lift and thrust compounded derivative of the UH-60 Black Hawk so as to operate at higher speeds (up to 250 kts). As shown in Figure 1, a fixed wing is added to the baseline helicopter for lift compounding. Propellers placed 10 ft laterally on either wings are used to provide auxiliary thrust in high-speed flight. The forward shaft tilt present in the UH-60 is removed since the auxiliary thrust is provided by the propellers. For better performance at high speeds, the high, nonlinear blade twist of the baseline UH-60A is reduced to an 8° linear twist rate. This reduces the associated high loads in high-speed forward flight due to negative lift and large drag on the advancing blade tips. The gross weight is raised to 20,110 lbs based on the maximum gross weight of the X-49A Speed Hawk, which is a reasonable approximation for the compound aircraft with wings, auxiliary propulsion, and additional structural weight. The forces and moments of the wing are found using Prandtl's lifting line theory. Key properties of the aircraft used in the simulation are provided in Table 1.

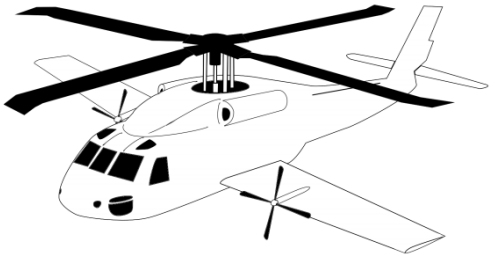

Figure 1: Compound Helicopter Schematic

For trim of a conventional helicopter, an equilibrium for the six rigid body degrees of freedom are uniquely determined by appropriately setting six trim variables: collective pitch (θ_{75}), lateral cyclic pitch (θ_{1c}), longitudinal cyclic pitch (θ_{1s}), tail rotor pitch (θ_{tr}), vehicle pitch (θ), and vehicle roll attitude (ϕ). For compound helicopter designs, a number of additional controls can be expected to be made available. This control redundancy allows an infinite number of potential steady-state trimmed flight conditions over which the controls can be optimized to achieve a target, such as low power. For the present study, three additional controls are considered: main rotor speed (Ω), auxiliary thrust (T_{coll}), and stabilator angle (δ_{stab}) Additionally, differential propeller thrust is used to replace the conventional tail rotor.

Consistent modeling approaches between both models include the following considerations. Interference effects between the wing, rotor, and fuselage are not modeled in this study. The propeller thrust is modeled as a point force (parallel to the waterline of the aircraft) and a yaw moment (parallel to the rotor torque vector), which are applied in line with the vertical coordinate of the center of gravity, and at the quarter-chord of the wings longitudinally. This is done so that the collective propeller thrust acts only in the longitudinal body direction without producing a coupled pitching moment. The magnitude of the thrust and the yaw moment are directly prescribed as controls.

The goal of this study is to evaluate the ability of machine learning techniques to correct error in a low-fidelity model, based on limited data from a high-fidelity model. As such,

Table 1: Compound helicopter configuration details

Parameter	Value
Gross Weight	20,110 lbs
C.G. Location	1.5 ft aft, 5.8 ft below hub
Main Rotor	
Rotor Radius	26.8 ft
Nominal Rotor Speed	258 RPM
Nominal Blade Twist	-8°
Shaft Tilt	0°
Blade Airfoils	SC-1094 R8/SC-1095
Stabilator	
Effective Area	43 ft^2
Airfoil	NACA 0012
C.P. Location	29.9 ft aft, 5.9 ft below hub
Wing	
Effective Area	220 ft^2
Mean Chord	5 ft
Aspect Ratio	9.0
Taper Ratio	0.825
Incidence Angle	3.8°
C.P. Location	0.5 ft aft, 6.5 ft below hub
Auxiliary Propulsor	
Radius	4.5 ft
Speed	1,934 RPM
Solidity	0.12
Number of Blades	4×2
Efficiency (150 - 250 kts)	0.80-0.87
Location	10 ft laterally, on each wing

two different modeling approaches are considered. The low-fidelity model will be represented by a MATLAB model, which is computationally efficient and can be queried in near-real time. The high-fidelity model will be represented by an RCAS model of the same compound helicopter, and is a more costly method of obtaining trim results. In future implementation of this strategy, this model can be replaced by CFD, flight tests, or any prohibitively time- or cost-intensive data acquisition method.

Low-Fidelity Model (Matlab)

The low-fidelity compound helicopter model for the current study is a MATLAB dynamic simulation model developed by Vayalali and Gandhi (Ref. 11), which is a derivative of the UH-60A Black Hawk simulation model developed by Krishnamurthi and Gandhi (Ref. 23) Validation of the UH-60A Black Hawk simulation model was performed in (Ref. 23) against a trim sweep and frequency responses of flight test and GenHel data from (Ref. 24).

The model uses a non-linear, blade element representation for the articulated single main rotor with aerodynamic forces determined by airfoil table. The blades are rigid, with offset flap and lag hinges. Flap and lag hinges are assumed to be coincident, with the flap motion following the lag in sequence. A 3-state Pitt-Peters dynamic inflow model (Ref. 25) is used to represent the induced velocity distribution on the rotor disk. The propeller thrust, torque, and power is modeled using a modified version of Goldstein's vortex theory, combined with blade element theory.

The governing equations of motion are given by

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}) \\ \vec{y} &= g(\vec{x}, \vec{u}) \end{aligned} \tag{1}$$

Where the state vector, $\vec{x}$, is comprised of rigid body and rotor states

$$\vec{x} = [\vec{x}_{fuselage},\ \vec{x}_{rotor}] \tag{2}$$

such that

$$\vec{x}_{fuselage} = [u, v, w, p, q, r, \phi, \theta, \psi, x, y, z] \tag{3}$$

$$\vec{x}_{rotor} = [\beta_0, \beta_{1s}, \beta_{1c}, \beta_d, \dot{\beta}_0, \dot{\beta}_{1s}, \dot{\beta}_{1c}, \dot{\beta}_d, \lambda_0, \lambda_{1s}, \lambda_{1c}] \tag{4}$$

The control input vector for the aircraft model is given by

$$\vec{u} = [\theta_{1c}, \theta_{1s}, \theta_0, \theta_{prop}, \delta_{prop}, \theta_{ail}, \theta_{rudder}, \delta_{stab}]^T \tag{5}$$

High-Fidelity Model (RCAS)

In application, the high-fidelity data is likely to take the form of coupled CFD-CSD or flight test data. For the sake of algorithmic development and evaluation, this study will consider a compound model developed in Rotorcraft Comprehensive Analysis System (RCAS) as a stand-in for higher-fidelity data (Ref. 26). This study does not intend to make any conclusions about the relative merits of the two modeling approaches, rather the goal is to highlight the capability of a machine learning method to correct for model differences in a unified modelling approach.

In the RCAS model, a dual core prescribed wake model is used to model the inflow in forward fight, which captures the effects of producing negative lift on the advancing tip of the blade. The structural model is represented by 13 elastic beam elements, with 36 aerodynamic sections, and an azimuthal resolution of 5° for calculation of airloads and structural response. The flap and lag hinges, and pitch bearings are modeled as torsional spring/damper elements, and pitch control is prescribed through a spring element with a stiffness that is representative of the pitch link and swashplate stiffness of a UH-60A. Propeller power is determined in post-processing using a blade element vortex theory (BEVT) model of two pitch controlled four-bladed propellers, which gives power as a function of advance ratio and propeller speed. The roll moment that would be produced by differential torque is considered negligible.

In summary, there are two primary differences between the MATLAB modeling approach and the RCAS approach. First, instead of rigid blade analysis with a flap degree of freedom, the RCAS model uses elastic blade modeling, with sectional blade stiffness properties that are representative of the UH-60A. Second, the rotor inflow is calculated using a prescribed wake vortex model.

Model Correction Algorithm

This section describes the role of the model correction algorithm, as well as the process for data acquisition used in the generation of results in the following sections. Figure 2 provides an illustrative example of the correction algorithm for a case where x is one dimensional and the output getting corrected is power prediction.

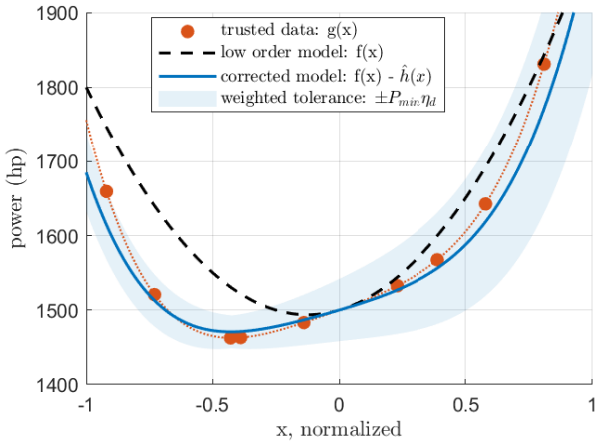

Figure 2: Illustrative example of model correction

The primary assumption of this study is that there is a low-fidelity model ($\mathbf{f}$) that predicts a quantity (Y), and can be queried at effectively no time or cost. The predicted value of $\hat{Y}$ has an unknown associated error, E.

$$\mathbf{f}(x,u) = \hat{Y} \tag{6}$$

$$\hat{Y} = Y + E \tag{7}$$

There is also a high-fidelity model ($\mathbf{g}$) which is assumed to have negligible error, but is very expensive and should be queried as little as possible.

$$\mathbf{g}(x,u) = Y \tag{8}$$

Each model provides a prediction of the power and controls required to trim based on the flight state (x) and the compound control settings (u). If it is assumed that the error (E) is entirely due to modeling error, then there is a function ($\mathbf{h}$) which describes (E) and is dependent only on observable quantities.

$$\mathbf{h}(x,u) = E \tag{9}$$

If known, this function can be used to correct the low-fidelity model and provide accurate predictions of Y without the cost and expense of $\mathbf{g}$.

$$\mathbf{f}(x,u) - \mathbf{h}(x,u) = Y \tag{10}$$

Since $\mathbf{h}$ is unknown, machine learning techniques will be employed to estimate it based on a limited set of data from querying the high fidelity model, $\mathbf{g}$. For this study, three techniques are used to determine the estimate $\hat{\mathbf{h}}$, artificial neural networks (ANN), Gaussian regression models, and support vector machines (SVM).

To start the algorithm, it is assumed that a small initial set of high-fidelity data exists. This data could be random, or could be driven by parametric interrogation of the low-fidelity model. A proposed heuristic is to start with a single data point at the most desirable state (i.e. minimum power), and the remaining points evenly distributed through the control and state spaces. Through one of the three machine learning approaches, the initial learned model, $\hat{\mathbf{h}}_\mathbf{0}$, is trained using this set of data.

In many machine learning applications data is abundant, and a naive approach to data acquisition such as randomly selecting and adding data would eventually provide the required quantity of data such that $\hat{\mathbf{h}} \approx \mathbf{h}$. Some machine learning approaches, such as reinforcement learning, bias the data collection towards combinations of predictors that provide the desired behavior. The goal of the current study is to use as little training data as possible to provide high accuracy near desired behavior, while simultaneously guaranteeing a known level of accuracy over the entire range. This can be expressed as:

$$\mathbf{f}(x,u) - \hat{\mathbf{h}}(x,u) + \epsilon = Y \tag{11}$$

where $\epsilon < \eta_d \cdot R$.

The function η_d governs the relaxation in tolerance at states further from the region of interest and can take any form. An exponential growth function is used here, which outputs allowable error as a fraction of variable R and is bounded by η_{min} and η_{max}. η_{min} is the maximum tolerance allowed in the minimum power region and η_{max} is the maximum allowable error for all possible states, and d is the normalized distance magnitude from the states that produce $\hat{Y}_{min}$.

$$\eta_d = \frac{\eta_{min}}{\phi} \tag{12}$$

$$\phi = e^{-\ln(\eta_{max}/\eta_{min})\cdot|d|^2} \tag{13}$$

$$d = \|u - u_{\hat{Y}_{min}}\| \tag{14}$$

The variable R is the normalizing parameter, which depends on the output being corrected. For a quantity that would have an ideal minimum, such as power, $R = \hat{Y}_{min}^{V_\infty}$. In this case the tolerance is defined as a percentage of the expected optimal power requirement for each velocity based on the low-fidelity model predictions. This value varies with velocity to ensure adequate scaling of tolerance over the entire velocity envelope as quantities like power vary significantly. For quantities that exist over a fixed range, such as pilot controls, the value of R is considered to be the maximum range of the corresponding control.

For the current study, the quantity of interest is power, η_{min} is 0.05 and η_{max} is 0.20. Therefore all error must be less than

5% at the minimum power state, less than 20% over the entire space, and the tolerance equation reduces to the following equation.

$$\eta_d = 0.05 \cdot e^{1.4 \cdot |d|^2} \tag{15}$$

Training Algorithm

When the initial set of data doesn't produce sufficiently accurate corrections, the model can be improved by selecting new data to evaluate and retrain the correction model. If the model uses known error to select subsequent data to add, it can effectively minimize the size of the dataset needed to ensure a threshold accuracy. The following algorithm will be used to iteratively select additional data from $\mathbf{g}$ to add to the training data. Figure 3 provides a flowchart summary of the training and data selection algorithm.

1. For the n^{th} iteration, $\hat{\mathbf{h}}_\mathbf{n}$ is trained from all available training data.
2. The error between the known data and corrected low-fidelity model prediction at that point are weighted.

$$\epsilon_w(x,u) = \phi \cdot \left(\mathbf{f}(x,u) + \hat{\mathbf{h}}_\mathbf{n}(x,u) - \mathbf{g}(x,u)\right) \tag{16}$$

   where ϕ is the radial basis function used in Equation 14.
3. The point in the state and control space with the largest weighted error is found.

$$[x,u]_{n+1} \ni \max(\epsilon_w([x,u]_{0:n})) \tag{17}$$

4. The new data point $[x,u]_{n+1}$ is evaluated using the high-fidelity model, $\mathbf{g}$.
5. If the error between the new data point and the corrected model prediction is not within the threshold defined by Equation 12, the new data point and $\mathbf{g}([x,u]_{n+1})$ evaluation are added to the training data. The preceding steps are then repeated until the points meet the desired tolerance η_d
6. Once the new data point and prediction using the corrected model meet the tolerance criteria, there is another check that is made before the algorithm quits. A point in the space which is farthest from all the points in the training dataset is selected and evaluated using the trained model. If the error is within the tolerance for the selected point, the algorithm quits, else the new data point is added to the training dataset and the process is repeated.

Because each new data point is assumed to be at the point of maximum expected error, this point is likely to represent the greatest weighted error. Therefore if it is within the tolerance, it is likely (but not guaranteed) that the error is within the threshold for all $[x,u]$. If a greater level of confidence is desired, a bias term can be added to Equation 12 for more strict quitting criteria at the cost of more evaluations of $\mathbf{g}$.

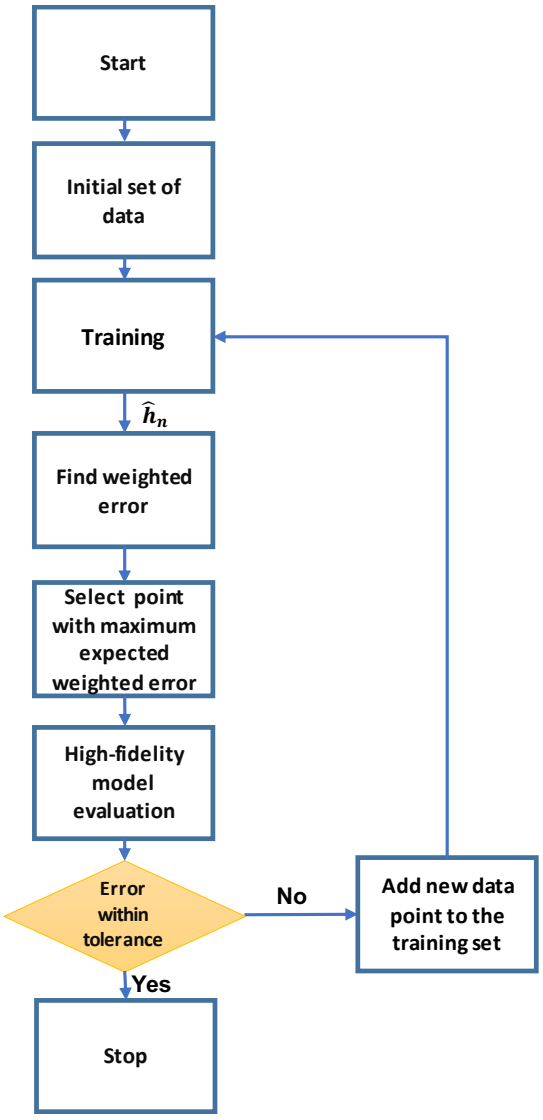

Figure 3: Algorithm Flowchart

Machine Learning Models

This study explores three learning techniques to evaluate their performance when working with sparse datasets. The formulation and hyper-parameters of each are described in the following section.

Artificial Neural Networks: Artificial neural networks are a popular choice for machine learning applications because of their ability to model any arbitrary function, as proven by the universal approximation theorem. In rotorcraft application, regressor neural networks have been used in health and usage monitoring applications, prediction of rotor loads, and autonomous shipboard landings (Refs. 27–29). The nature of current problem involves modeling continuous variables using a regressor neural network.

For the purposes of regression, the artificial neural networks use numerically stable neuron activation functions such as a continuously differentiable sigmoid function to represent the

decision made by each neuron. A hyperbolic-tangent sigmoid activation function is used for the study presented in this paper.

Back-propagation learning algorithm is the most common approach to tune the weight and bias of a neural network. Levenberg-Marquardt back propagation is chosen for back-propagation training and is implemented using MATLAB Statistics and Machine Learning toolbox.

The artificial neural network used in the present study consist of a single hidden layer of neurons, with each neuron connected to every input node, and every output node, as shown in Figure 4. The input node consists of the redundant controls $(\Omega, T_{coll}, \delta_{stab})$. These inputs are scaled to a range of [-1, 1] before training. The output layer gives the estimated correction as a linear combination of the weighted hidden outputs and bias. The hidden layer size is set to 7 neurons, which minimizes RMSE for the results given in the following section at 150 kts. Figure 5 shows the variation in RMSE with hidden layer size using 1-fold cross validation.

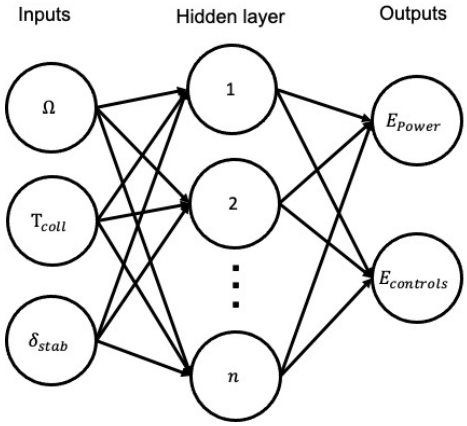

Figure 4: Neural Network Architecture

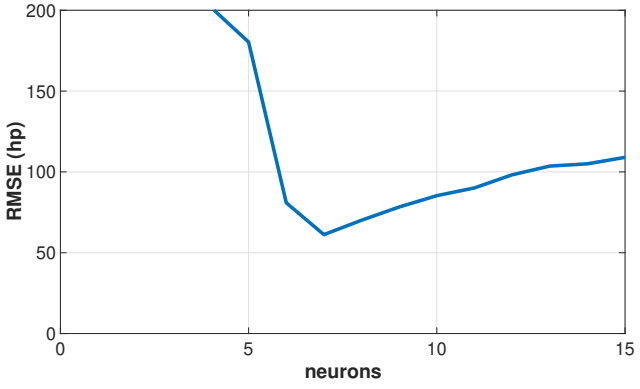

Figure 5: 1-fold cross validation showing variation in neural network power correction modeling error (hp) with increased hidden layer size

Gaussian Regression Models: Gaussian regression models are kernel based linear regression formulations that have the ability to model non-linear relationships between observations and inputs. A Gaussian process is fully defined by its mean function and covariance function (kernel). Given a training data set $\mathcal{D}$ with n input-output pairs $\{ x_i \in \mathbb{R}^{\mathcal{D}}, y_i \in \mathbb{R}, i = 1,2,3...n\}$, a Gaussian regression model can be formulated as follows (Ref. 30)

$$y = p(x) + e \tag{18}$$

The mean function, $m(x)$ describes the mean of any given data point x, and the kernel $k(x,x')$ describes the relationship between any given two data points. As opposed to a traditional regression model, in Gaussian process regression, Gaussian process is placed over $p(x)$, and an independent, identically distributed (iid), zero mean prior with variance σ_n^2 is placed on the noise term e

$$p(x) = GP(m(x), k(x,x')), \qquad e\ iidN(o, \sigma_n^2) \tag{19}$$

For the present study, $m(x)$ is set to zero, and the squared exponential kernel function is used for p

$$k(x,x') = \sigma_0^2 e^{-(\frac{1}{2}(x-x')^T A^{-1}(x-x'))} \tag{20}$$

where σ_o^2 is the outputs variance, and A is a diagonal matrix of the characteristic length scales of each dimension (D) i.e each covariate in the input dataset. Gaussian regression optimizes the hyperparameters $(\theta \equiv \sigma_0^2, A, \sigma_n^2)$ via Type II Maximum Likelihood. For the present method, the marginal likelihood of the training observations is maximized with respect to the hyperparameters, and its negative log is minimized instead as follows:

$$\begin{aligned} \hat{\theta} &= \underset{\theta}{argmin} - log p(y|X,\theta) \\ -log p(y|X,\theta) &= -log N(y|0, K_{xx} + \sigma_n^2) \\ &= \frac{-1}{2} y^T (K_{xx} + \sigma_n^2)^{-1} y - \frac{1}{2} log|k_{xx} + \sigma_n^2| - \\ &\frac{n}{2} log 2\pi \end{aligned} \tag{21}$$

The predictive distribution over the prediction y can be defined using properties of multivariate Gaussian distributions (Ref. 30).

Support Vector Regression: Support vector regression is a supervised learning algorithm that is used to predict discrete values. The basic idea behind the support vector regression is to find the best fit line. In support vector regression, the best fit line is the hyperplane that has the maximum number of points. The problem of support vector regression is to find

a function that approximates mapping from an input domain to real numbers on the basis of a training samples.

Consider the set of training data where x_n is a multivariate set of n observations with observed response values y_n, the SVM algorithm is given by

$$\begin{aligned} &\underset{b,w}{minimize} \quad \frac{1}{2} w^T w \\ &\text{subject to :} \quad y_n(w^T x_n + b) \geq 1 \text{for} \quad n = 1, 2, ..n \end{aligned} \tag{22}$$

where w refers to the regression coefficients.

Since the regression line should not pass through all the data points so as to avoid overfitting and improve generalization characteristics, a slack variable ζ is added to the above equation.

$$\begin{aligned} &\underset{b,w}{minimize} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \zeta_n \\ &\text{subject to :} \quad y_n(w^T x_n + b) \geq 1 - \zeta_n \quad \text{for} \quad n = 1, 2, ..n \\ &\zeta_n \geq 0 \quad \text{for} \quad n = 1, 2, ...n \end{aligned} \tag{23}$$

This trades off the soft in sample error $\sum_{n=1}^{N} \zeta_n$ with weight norm $1/2 w^T w$. C plays the role of a regularization parameter. For the present study y_n is the error in power using the two fidelity models and x_n are the inputs.

The dual form of the above equations can be constructed using a Lagrangian function from the primal function by introducing nonnegative multipliers α_n for each observation x_n.

$$\mathcal{L}(\alpha = \frac{1}{2} w^T w = \sum_{n=1}^{N} \alpha_n (y_n (w^T x_n + b) - 1) \tag{24}$$

The details of the primal and dual formulations are described in (Refs. 31, 32).

Some regression problems cannot adequately be described using a linear model. In such a case the inputs are extended to a non-linear space using kernels. In this study, Gaussian kernel is used to transform the inputs to a non-linear space before finding the regression coefficients. The Gaussian kernel is given by

$$K(x, x') = e^{-\gamma |x - x'|^2} \tag{25}$$

RESULTS AND DISCUSSION

Low-Fidelity Model Correction at 150 kts

A total of 321 feasible trim states were produced using RCAS through parametric variations of control inputs (Ω, T_{coll}, δ_{stab}) at 150 kts. These points are given in Table 2, and represent high-fidelity data that can be used to evaluate the accuracy of, and make corrections to the low-fidelity model. Figure 6 shows a scatter plot of actual power vs predicted power for a testing set of 288 trim states at 150 kts. The x-axis of the plot represents the true power requirement given by the high-fidelity model and the y-axis represents the predicted power requirement obtained using low-fidelity model. Data coincident with the diagonal reference line represents a model which perfectly predicts the vehicle power requirement, while data above and below the line over-predicts and under-predicts the power requirement, respectively. Throughout the space defined by varying rotor speed, auxiliary thrust, and stabilator angle, the low-fidelity model tends to over-predict the actual power requirement. Near the minimum power region, the low-order model over-predicts the high-fidelity data by 9% (126 hp). Figure 7 shows that the root means squared error (RMSE) for the low order model is 301 hp across the trim space, indicating a high expected error of about 21% of the minimum power. This data is also provided in the first row of Table 3.

Table 2: Training/testing data generated using high-fidelity (RCAS) model

V (kts)	No. of trim states	δ_{stab} (deg)	Ω (rad/s)	T_{coll} (lbs)
50	790	-18 - 18	22 - 27	500 - 6000
	in increments of	3	1	500
100	740	-18 - 18	17 -27	500- 6000
	in increments of	3	1	500
150	321	-6 - 12	19 - 27	250 - 3500
	in increments of	3	1	250
200	102	-1.5 - 6	18 -25	2250 - 4500
	in increments of	1.5	1	250

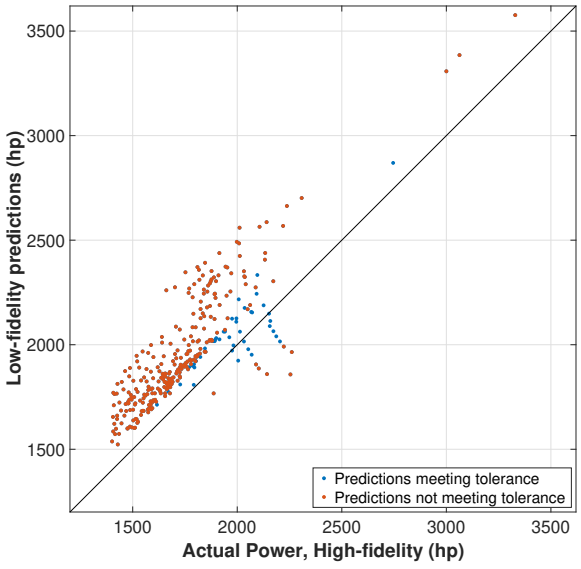

Figure 6: Actual vs predicted power of testing dataset, evaluated using low-fidelity model at 150 kts

Table 3: RMSE of models evaluated on testing dataset at 150 kts

	Power (hp)	Collective (deg)	Lateral cyclic (deg)	Longitudinal cyclic (deg)	Training set (n pts.)	Testing set violations
Low-fidelity model	301 (21%)	4.6 (42%)	0.91 (36%)	4.9 (27%)		
ANN model (initial)	240 (17%)	2.3 (21%)	0.55 (21%)	3.7 (20%)	15	91 (32%)
ANN model (final)	170 (12%)	1.4 (13%)	0.28 (11%)	1.8 (10%)	33	26 (9.0%)
GR model (final)	125 (8.9%)	1.3 (12%)	0.23 (9.2%)	1.6 (9.4%)	34	25 (8.6%)
SVR model (final)	102 (7.2%)	1.1 (9.9%)	0.21 (8.2%)	1.3 (7.2%)	33	25 (8.6%)

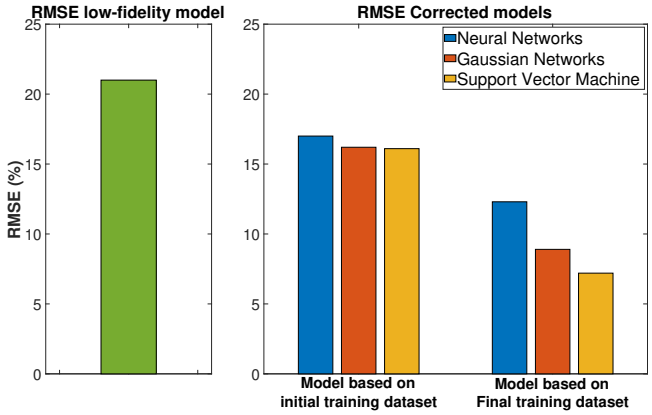

Figure 7: RMSE - Total Power

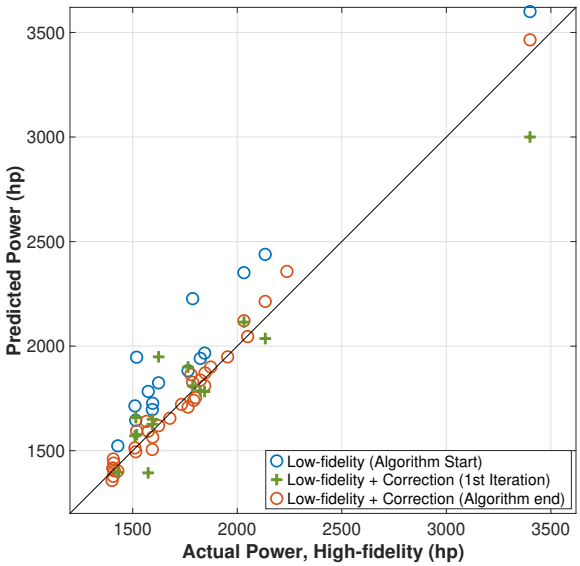

Figure 8: Actual vs predicted power for training dataset at 150 kts

For the iterative learning process, a subset of 15 points are used to initialize the problem, with one point prescribed at the state which minimizes the vehicle power requirement (as predicted by the low-fidelity model), and rest of the points uniformly distributed in the space spanned by the three inputs. Due to the resolution of the parametric variations produced in RCAS, the point of minimum power and all subsequent points requested by the model correction training algorithm are approximated with the nearest available data point that exists within the RCAS dataset. The 15 initial training data points, and their respective predictions using the low-fidelity model are shown on Figure 8.

To model the error between the low-fidelity and high-fidelity data, an initial machine learning model is constructed. Three machine learning approaches are used for comparison, which are described in the previous section; artificial neural network (ANN), Gaussian regression (GR), and support vector regression (SVR). Each method is initialized with the same training dataset, but depending on the algorithm, the methods could result in different training data selected during the learning process.

Consider the artificial neural network-based correction model. Figure 8 shows the improvement in the training dataset when the low-fidelity model is corrected using the neural network, where the general over-prediction of the low-fidelity model is reduced significantly. By using the same model to test on the withheld testing data, Figure 9 provides an accurate evaluation of the predictive capabilities of the initial neural network-based corrected model. By correcting with a learning model, expected error in power prediction on the testing dataset is reduced by 4% of the minimum power, as the RMSE reduces from 301 hp to 240 hp. This trend is also tabulated in Table 3 and shown in Figure 7. Comparing Figures 6 and 9 visually portrays the source of the lower RMSE, as there is reduction in global over-prediction marked by a downward shift along the y-axis.

The training algorithm then proceeds to enrich the training dataset with trim states corresponding to the greatest weighted error. After an additional 18 points have been selected, the new corrected model predictions meets the distance-weighted tolerance in Equation 15. For the neural network model, Figure 8 shows that these additional training data guided by the algorithm are highly clustered around the minimum power region, when compared against the initial dataset. Figure 10 compares the actual total power against the predicted total power for this final neural network-based model. Compared to the RMSE of 240 hp for the model in Figure 9, the model in Figure 10 has an RMSE of 170 hp, with greater error where the actual power exceeds 2000 hp. This improvement is most easily identified in the data at power requirements of less than

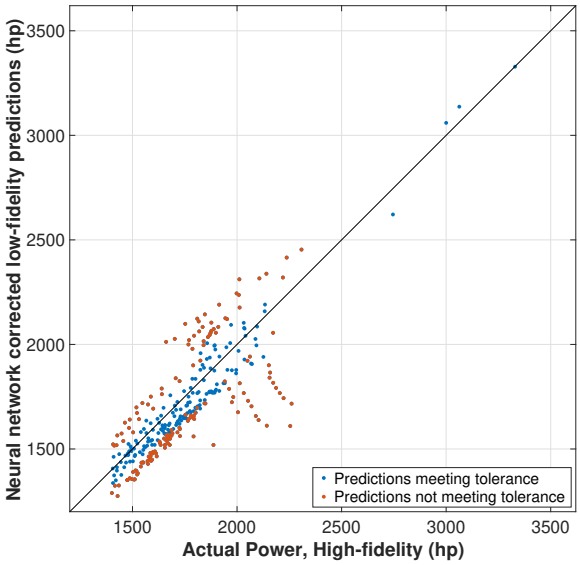

Figure 9: Actual vs predicted power of testing dataset, evaluated using initial ANN-based correction model at 150 kts

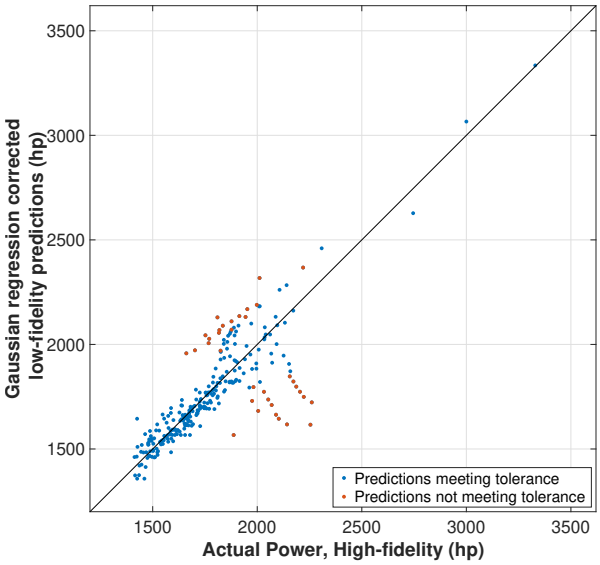

Figure 11: Actual vs predicted power of testing dataset, evaluated using final GR-based correction model at 150 kts

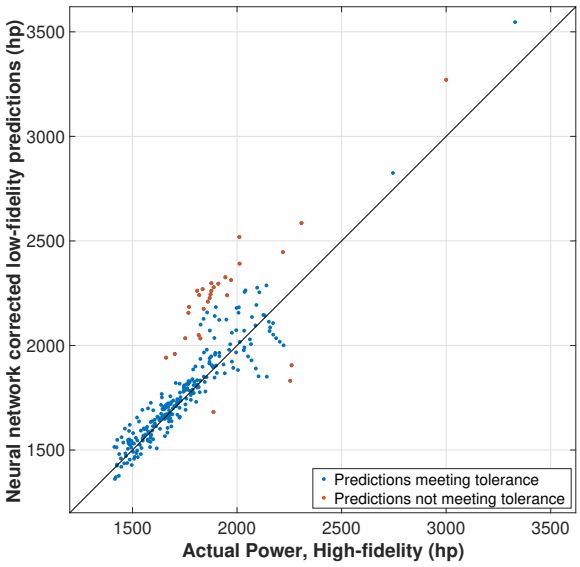

Figure 10: Actual vs predicted power of testing dataset, evaluated using final ANN-based correction model at 150 kts

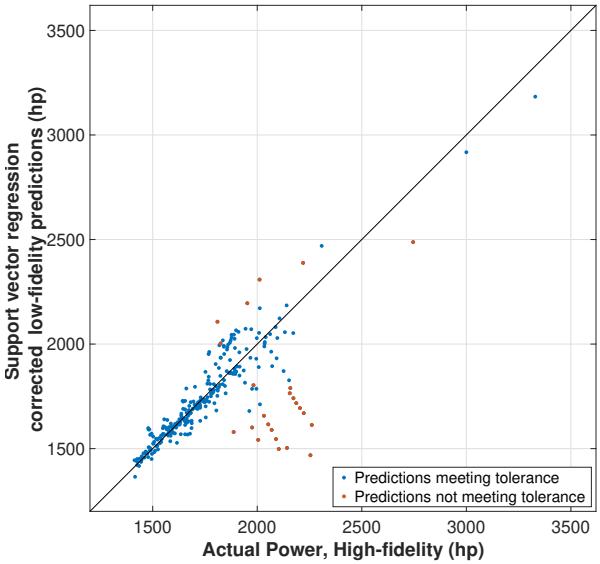

Figure 12: Actual vs predicted power of testing dataset, evaluated using final SVR-based correction model at 150 kts

1500 hp, where compared to the under-predictions in Figure 9, Figure 10 exhibits greater accuracy. Based on these results, the algorithm appears to successfully select training data which minimizes error in the minimum power region of interest.

Figures 9, and 10 can also be used to evaluate the assertion that the quitting criteria for the model is sufficient. Initially, the number of trim states in the testing set which violate the weighted error tolerance function are evenly distributed through the range of power requirements, totaling 91 states or nearly a third of the total testing dataset (Table 3). After the additional training has been performed, the points in violation is reduced to 26, or 9% of the entire testing dataset. The indicated points in Figure 10 shows where this unaccounted error is occurring, mostly an over-prediction of moderate powers. If this level of inaccuracy is insufficient for an application,

more rigorous quitting criteria can be employed at the cost of additional data requirements.

Machine Learning Error Model Comparison at 150 kts

In order to compare the three machine learning modelling approaches, Figures 11 and 12 show the evaluation of the testing datasets for the final Gaussian and support vector regression models, respectively. These Figures are directly comparable with Figure 10. The algorithm selects 33 points before quitting for both the neural network and support vector regression models, while requiring 34 points for the Gaussian regression model. Qualitative visual observation suggests that while the Gaussian regression seems to perform comparably to the neural network model. The support vector regression model maintains a more accurate prediction at low and moderate powers, as evidenced by the tight clustering in line with the reference in Figure 12. As shown in Figure 7, the RMSE of the testing dataset for the fully trained support vector regression model is 102 hp, which is 5% lower than either of the other two models. The superior performance of the support vector regression model can be attributed to the better extrapolation characteristics when dealing with small datasets (Ref. 32). Based on these results, support vector regression is picked as the choice of machine learning method for modeling the error function for the remainder of the study.

Trim Control Prediction at 150 kts

Similar to power requirement, the model correction algorithm can be used to correct for error in trim control setting predictions from the low-fidelity model. Figures 13, 14, and 15 show trim controls for collective control (θ_0), lateral cyclic pitch (θ_{1c}), and longitudinal cyclic pitch (θ_{1s}) respectively. The x-axis on each plot is the normalized distance in the control space from the set of controls that produce minimum power. The data shown is a random subset of the larger testing dataset, for visualization purposes. The markers for the high-fidelity data are accompanied by error bars which denote the tolerable range of weighted error. Initially, the low-fidelity model tends to exceed the upper bound of collective pitch error tolerance throughout the data, while lateral and longitudinal cyclic exceed the error tolerance both above and below the limits. After correction with support vector regression, the high-fidelity data is nearly coincident with the corrected low-fidelity data in most of the testing data, and well within the error tolerance.

The same set of trim states are used for training the model for controls using each of the three machine learning techniques. The three columns to the right of power predictions in Table 3 show the improvement in modeling error for the three main rotor controls. Support vector regression-based correction provides a 1-4% improvement in trim control prediction RMSE compared to the other two learning techniques. Similar to the power predictions, support vector regression significantly improves the low-fidelity model, giving the lowest modeling error.

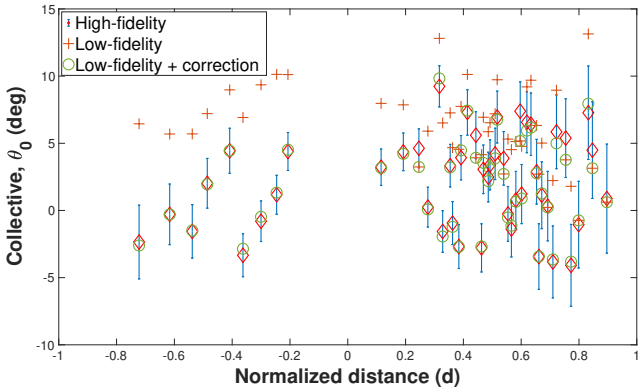

Figure 13: Interval plots for collective control (testing dataset)

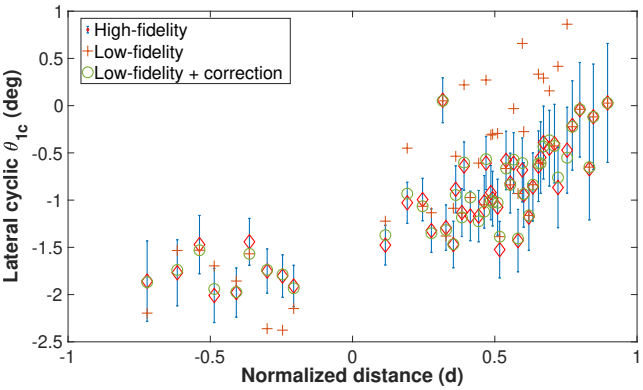

Figure 14: Interval plot for lateral cyclic (testing dataset)

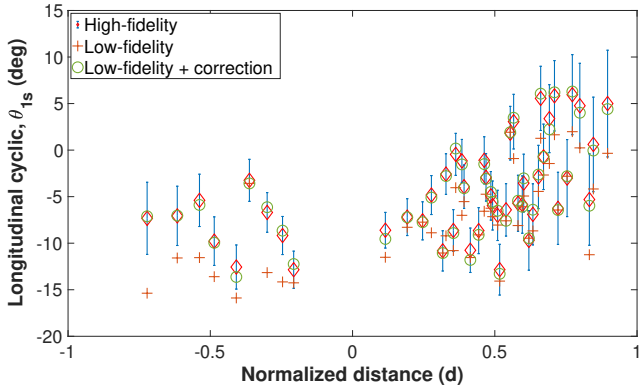

Figure 15: Interval plot for longitudinal cyclic (testing dataset)

50 kts, 100 kts, and 200 kts Results

In this section, the same model correction training algorithm is applied to data at other discrete speeds. Data is generated at 50 kts, 100 kts, and 200 kts through parametric variation of redundant controls, as listed in Table 2. As was the case at 150 kts, one point is prescribed at the state that minimizes total power for the low-fidelity model, and rest of the points are distributed evenly in the space spanned by three inputs. For each

Table 4: RMSE of additional discrete velocity models evaluated on testing datasets

	Power (hp)	Collective (deg)	Lateral cyclic (deg)	Longitudinal cyclic (deg)	Training set (n pts.)	Testing set violations
50 kts						
Low-fidelity	312 (29%)	6.3 (55%)	1.1 (39%)	6.1 (42%)		
Corrected model	194 (18%)	0.8 (7.1%)	0.4 (14%)	0.8(5.6%)	59	29 (3.7%)
100 kts						
Low-fidelity	321 (40%)	6.8 (6.0%)	1.4 (50%)	7.2 (50%)		
Corrected model	132 (16%)	0.2 (1.7%)	0.2 (7%)	0.7 (4.8%)	61	31 (4.2%)
200 kts						
Low-fidelity	514 (17%)	5.1 (45%)	0.9 (32%)	5.9 (41%)		
Corrected model	370 (12%)	1.1 (9.7%)	0.3 (10%)	1.1 (7.6%)	26	6 (5.9%)

Table 5: RMSE of continuous velocity model evaluated on testing dataset

Velocity (kts)	Power (hp)	Collective (deg)	Lateral cyclic (deg)	Longitudinal cyclic (deg)	Training set (n pts.)	Testing set violations
50	202 (18%)	0.8 (6.9%)	0.41 (14%)	0.91 (6.3%)	40	15 (2.0%)
100 (interpolated)	340 (43%)	8.1 (79%)	3.2 (45%)	3.1 (21%)	0	124 (16%)
100 (final)	121 (15%)	1.1(11%)	0.31 (10%)	0.75 (5.2%)	6	11 (0.15%)
150	110 (7.8%)	1.2(10%)	0.31 (12%)	1.1 (6.2%)	25	11 (3.7%)
200	321 (11%)	1.2 (10%)	0.20 (14%)	1.3 (8.9%)	18	3 (4.8%)

of the three velocities, a discrete model is trained by following the correction algorithm training procedure to correct the low-fidelity predictions. Each model is evaulated against the corresponding testing data, and the resulting scatter plots of predicted vs actual power are given in Figures 16, 17, and 18. Table 4 gives the summary of RMSE in power requirement and main rotor trim control settings for each model, as compared to the corresponding low-fidelity model.

For the 50 kts model, the algorithm initializes with 20 points for the first iteration of training. The training algorithm adds 39 points to the initial training dataset, 16 of which are within 300 hp of the minimum power trim state, before it satisfies the quitting criteria for all the training points for total power and controls. The RMSE of power predcitions comes down from 312 hp to 194 hp. For the 100 kts model, the algorithm again initializes with 20 points, but requires an additional 41 points to satisfy the error tolerance. Compared to the low-fidelity model, the corrected model is able to reduce the RMSE of power predictions from 321 hp to 132 hp. For the 200 kts model, the algorithm initializes with 12 points and requires an additional 14 points to satisfy the error tolerance. The RMSE reduces from 514 hp to 370 hp. These results suggest that the model correction algorithm produces consistent results across flight regimes, with slightly improved performance at high velocities.

For brevity, control predictions have been omitted, which follow similar trends that can be seen in Table 4.

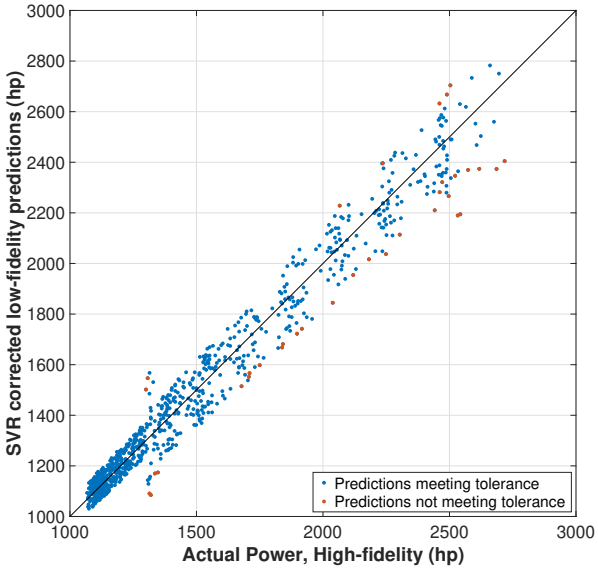

Figure 16: Actual vs predicted power of testing dataset, evaluated using SVR-based correction model at 50 kts

Continuous Velocity Modeling Results

In order to make accurate predictions across flight speeds, and leverage trends that carry throughout the flight envelope, it is necessary to unify the models for discrete flight speeds. If flight speed is included as a predictor of power in the learning model, it is hypothesized that a continuous velocity model will be able to make corrections at any flight speed, including interpolating to velocities at which no training data exists. To test this hypothesis, data at 50 kts, 150 kts, and 200 kts are used to train a continuous velocity correction model.

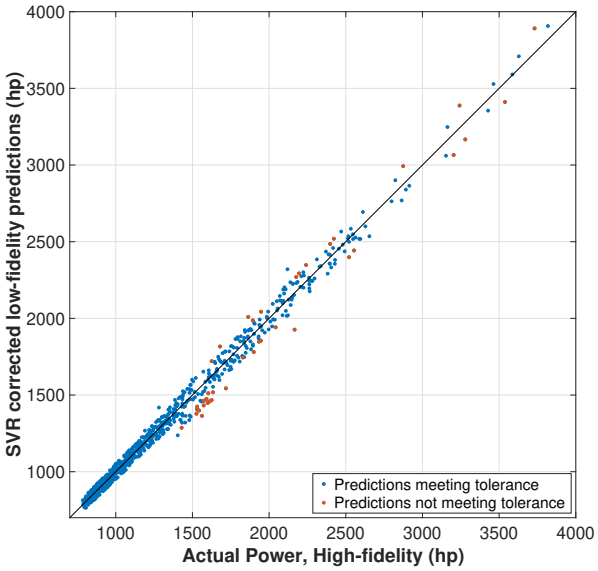

Figure 17: Actual vs predicted power of testing dataset, evaluated using SVR-based correction model at 100 kts

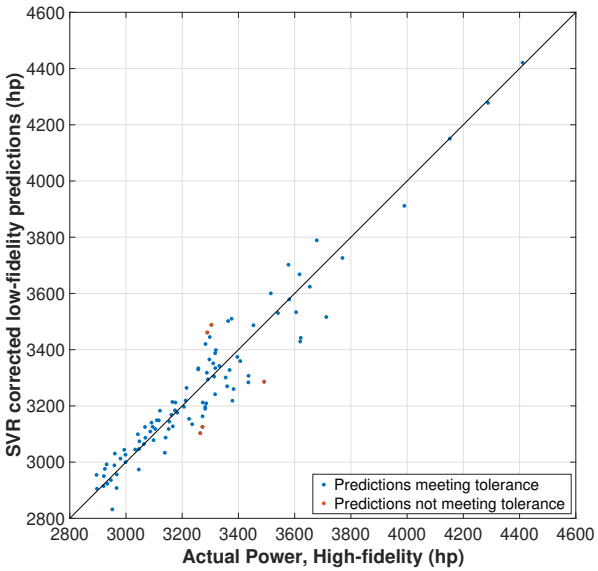

Figure 18: Actual vs predicted power of testing dataset, evaluated using SVR-based correction model at 2000 kts

The input space for this model has four variables: Ω, T_{coll}, δ_{stab}, and V. Calculations for normalized distance (d) are modified to include velocity distance when computing data sparseness. This is subsequently taken into account for tolerance (η_d) calculations in Equation 12. The normalization scheme described in the Model Correction Algorithm section describes how normalization of power will be performed using a velocity dependant minimum, which is done to prevent a model from erroneously biasing data collection towards trim states near the power-optimal velocity of best endurance. As before, a support vector regression-based model of error is used to predict the correction needed to bring the low-fidelity model in line with the high-fidelity data.

The continuous velocity model is initialized with 20 points, 19 of which are distributed over the 4-dimensional input space and one point is at the set of predictor variables which minimize the total power requirement (V_{be}), which is achieved closest to 50 kts (subject to the resolution of the high-fidelity data). After 63 iterations, the model meets the quitting criteria for all the points in the training data set, which brings the total number of training points required to 83.

The total number of feasible trim states for these three velocities is 1213, with only 83 used for training. The remaining points can be used to examine the out-of-sample performance of the trained model. Figure 19 provides a scatter plot of the predicted vs actual power, showing how well the continuous power correction model performs on this testing dataset. Testing dataset performance at each of the three velocities can be directly compared against the corresponding discrete velocity model in Figures 12, 16, and 18. Rows 1, 4, and 5 in Table 5 provide the RMSE of the model evaluated on this testing dataset at each of the three velocities at which training data exists.

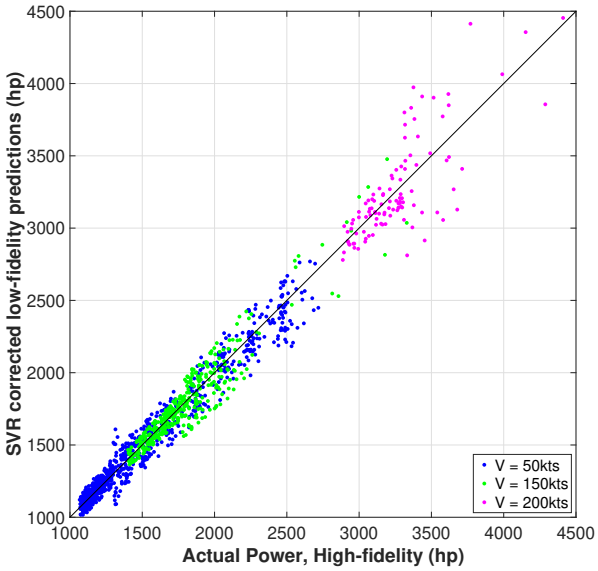

Figure 19: Actual vs predicted power of complete testing dataset using SVR-based correction model trained on 50 kts,150 kts, and 200 kts

Based on this data, and the corresponding plot in Figure 20, the continuous velocity model performs comparably to the discrete model. At 50 and 150 kts the power prediction RMSE is less than 1% larger, while it is 1% smaller at 200 kts. Despite this near equivalence of expected error, the continuous

velocity model reduces the data dependency by 30% compared to the combined 118 trim states required for the three discrete models. Over the velocities spanned by the training data, 29 points violate the intended tolerance, which is a 52% improvement upon the 60 violations observed over the three discrete models. In summary, these results suggest that a continuous velocity correction model seems to be able to leverage trends within the data at disparate flight speeds to perform within 1% of the discrete velocity correction models, but at a 30% reduced data cost, and with 52% fewer violations of the prescribed tolerance within the testing data.

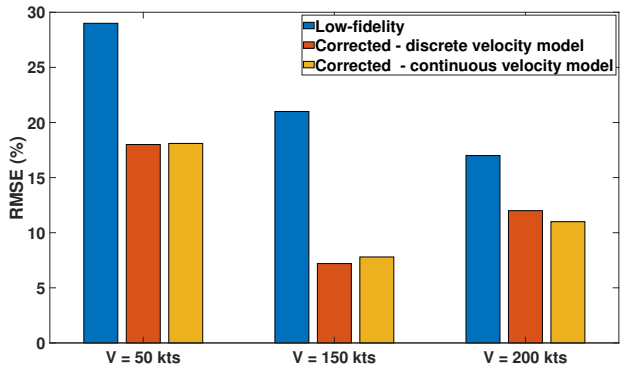

Figure 20: RMSE comparison between the low-fidelity and corrected models for total power at 50, 100, and 200 kts

Interpolation at 100 kts

With flight speed as one of the predictor variables, the trained model is capable of interpolating between flight speeds where no data exists. Figure 21 shows the performance of the model when interpolating to test on the 100 kts data, while the second row of Table 5 provides the corresponding RMSE for power and the main rotor controls.

While the continuous velocity correction model performed well when tested at the same speeds for which training data was used (RMSE between 7-18%), the interpolated values at 100 kts significantly under-perform, with RMSE of 43%. The RMSE for this dataset is 683 hp, with 124 points which don't meet the weighted error tolerance. The loss of predictive capability is even more exaggerated for the models of collective and longitudinal cyclic, for which the RMSE grows from 6-10% to 64-79%. Visual inspection of Figure 21 indicates that this error is evenly distributed across all power regimes, and consists almost exclusively of overpredictions. Since the model was not trained on data at 100 kts the performance wasn't expected to be as accurate as it was for the other speeds. This is because of highly sparse data in the velocity direction and demonstrates the need for more training points at 100 kts. Compared to the discrete model, which uses 61 training points at 100 kts, and reduces low-fidelity RMSE by 24%, the interpolated predictions use zero training points at

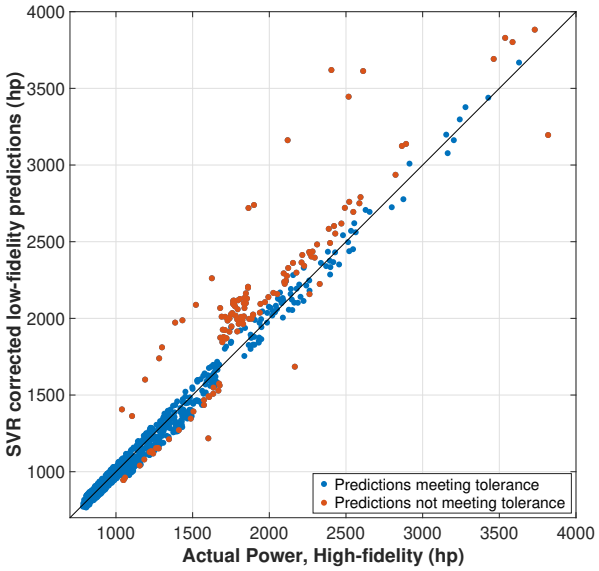

Figure 21: Prediction at 100 kts using continuous velocity model trained at 50 kts, 150 kts, and 200 kts

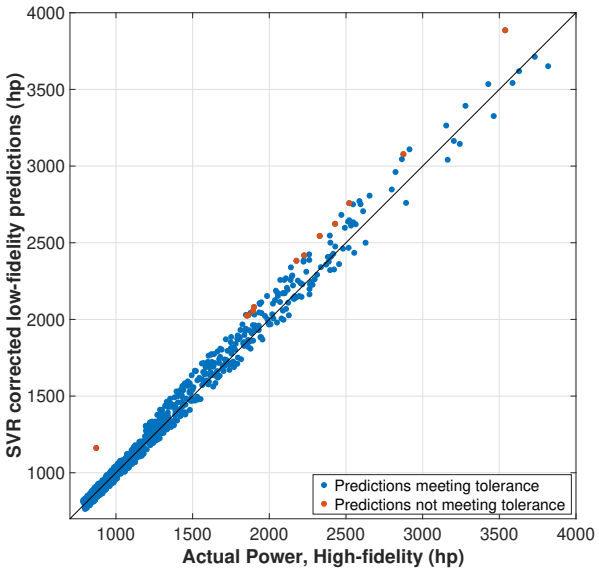

Figure 22: Performance at 100 kts using continuous velocity model trained at 50 kts, 150 kts, 200 kts and 6 points at 100 kts

100 kts, and increase the RMSE of the low fidelity model by 3%.

To improve the predictions, a small number of high-fidelity points at 100 kts are added, guided by the algorithm based on maximum weighted error. The final model adds an additional 6 data points at 100 kts to reduce weighted error to within the tolerance. Performance of this model on the testing dataset

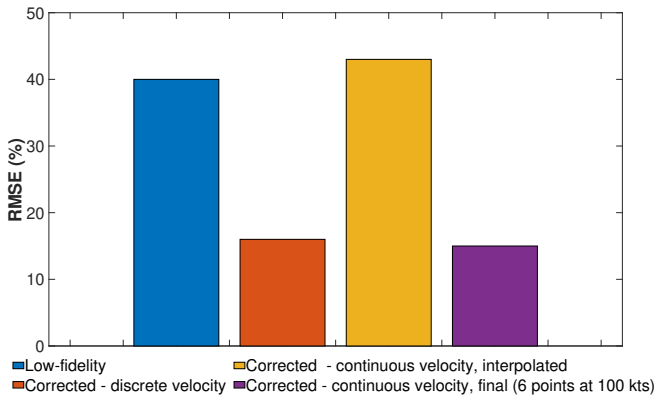

Figure 23: RMSE comparison for data at 100 kts

at 100 kts is shown in Figure 22. Comparing this scatter plot against Figure 21 highlights the marked improvement, as the testing dataset is more closely aligned throughout, and the 124 points in violation of the tolerance are reduced to 11 points. Figure 23 shows that this trend is consistent with error metrics, as RMSE for the testing dataset is reduced from 340 to 171 hp, a 28% reduction in error as normalized by minimum power at 100 kts. With the addition of just 6 points at 100 kts (for a total of 89 across all flight speeds), the continuous velocity model now performs 1% better at 100 kts than the discrete model does with 61 points at 100 kts. The continuous velocity model also performs as well or better at every tested flight speed when compared to the discrete models, which require a total of 179 testing points (twice as many).

Purely Data-Driven Modeling

Algorithm Modifications: To test the value of the low-fidelity model for improving predictive capabilities with sparse data, a purely data-driven model will be used for comparison. The idea behind data-driven modeling is to find relationship between the system variables without any explicit knowledge of the underlying process. This approach has been widely adapted in integrated health management systems (Refs. 33, 34), and guidance systems (Ref. 35).

The difference in the approach for training in this section is the absence of a low-fidelity model to make corrections ($\hat{\mathbf{h}}$). Without a low-fidelity model underlying the correction model, the machine learning model must predict on the raw data (power and controls), rather than the error. Consider the high-fidelity data ($\mathbf{g}$) and let the support vector regression model fitting this data be defined as $\hat{\mathbf{g}}$. If the allowable error (ϵ) is defined the same way as in the algorithm section, then this new formulation will replace the model described in Equation 11 with:

$$\hat{\mathbf{g}}(x,u) + \epsilon = Y \tag{26}$$

Each model is initialized with the same set of data as was used for the evaluation of the low-fidelity correction approach. For this study, the iterative process for selection and adding high-fidelity data to the training dataset is maintained, so that this model must still satisfy the same tolerance and quitting criteria for the error as outlined in the algorithm section of the paper. The error, ϵ, is weighted the same as for the correction approach (using equation 16), and and the training algorithm used to select additional data is identical. Once training is complete the models can be directly compared against the low-fidelity correction models.

Discrete and Continuous Velocity Models: Table 6 reproduces the results from the previous section, in which a separate model was produced to predict power and main rotor controls at 50 kts, 150 kts, and 200 kts. Figure 24 compares RMSE of the discrete velocity models, with the corrected continuous velocity model at 50 kts, 150 kts, and 200 kts. Upon completion of training, the RMSE in prediction was slightly higher than that of the correction model, with 1 to 3% greater error in power (normalized by minimum power at each velocity). Within the testing dataset, an increase of 50-240% of the states were in violation of the tolerance criteria after training was concluded. These results suggest a moderately reduced performance of the final model, but with far less consistency of predictions across the control space.

This observation can be confirmed by Figure 25, in which the actual power requirement vs pure data-driven power requirements are compared. In Figure 25 the RMSE is relatively small for trim states with power requirements less than 2000 hp, but as power increases, the degree of under prediction grows more severe on average, with errors exceeding 30% on average for power requirements between 2700 hp 3400 hp. The major deficiency in this approach, however, is the additional data required. Each entirely data-driven model required 2.1-4.9 times as much data to train to a similar RMSE as its correction-based model counterpart.

Next, the ability for the purely data-driven model to reduce data requirements is considered. Mirroring the process for the error-correction continuous velocity model, a new surrogate model is initialized with a training set using 20 points of data between 50 kts, 150 kts, and 200 kts. The training algorithm converges with a total of 405 trim states across the three velocities. Table 7 contains the metrics on training, as well as the

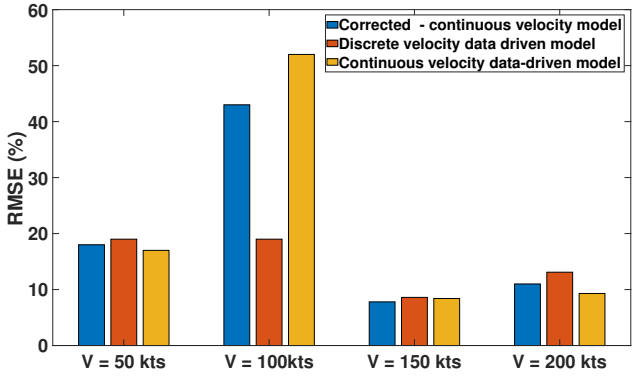

Figure 24: RMSE comparison of corrected continuous velocity model with data-driven discrete and data-driven continuous velocity model

Table 6: RMSE of discrete velocity pure-data model evaluated on testing dataset

Velocity (kts)	Power (hp)	Collective (deg)	Lateral cyclic (deg)	Longitudinal cyclic (deg)	Training set (n pts.)	Testing set violations
50	202(19%)	0.82 (7.2%)	0.20 (7.3%)	1.1 (7.1%)	287	36 (7.1%)
100	147(19%)	0.83 (8.1%)	0.14 (5.2%)	1.2 (8.3%)	281	41 (8.9%)
150	121 (8.6%)	0.74 (6.7%)	0.14 (5.8%)	1.04 (5.8%)	72	32 (13%)
200	380 (13.1%)	0.80 (7.1%)	0.23 (8.3%)	1.2 (7.9%)	58	9 (20%)

Table 7: RMSE of continuous velocity pure-data model evaluated on testing dataset

Velocity (kts)	Power (hp)	Collective (deg)	Lateral cyclic (deg)	Longitudinal cyclic (deg)	Training set (n pts.)	Testing set violations
50	181 (17%)	0.7 (6.1%)	0.32 (11%)	1.0 (7.0%)	268	21 (4.0%)
100 (interpolated)	412 (52%)	5.4 (53%)	0.8 (28%)	3.9 (26%)	0	309 (5.2%)
100 (final)	110 (14%)	0.9 (8.8%)	0.2 (7.1%)	0.89 (6.1%)	23	8 (1.0%)
150	118 (8.4%)	1.4 (12%)	0.48 (18%)	1.2 (6.6%)	99	10 (4.5%)
200	271 (9.3%)	1.3 (11%)	0.51 (18%)	1.2 (8.2%)	38	8 (12%)

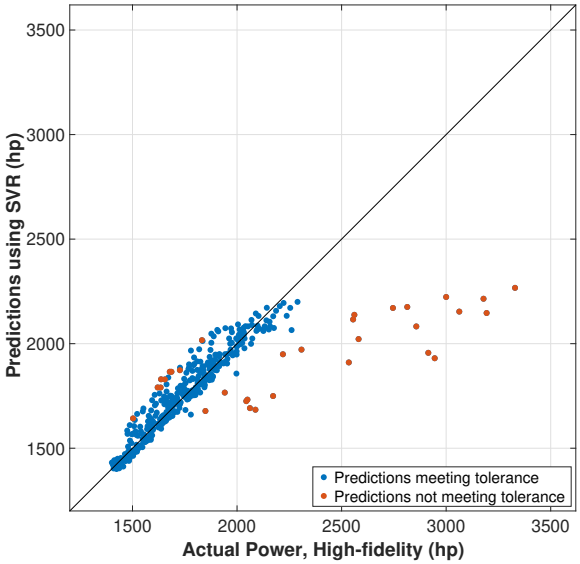

Figure 25: Actual vs predicted power of 150 kts testing dataset, evaluated using final data-driven model

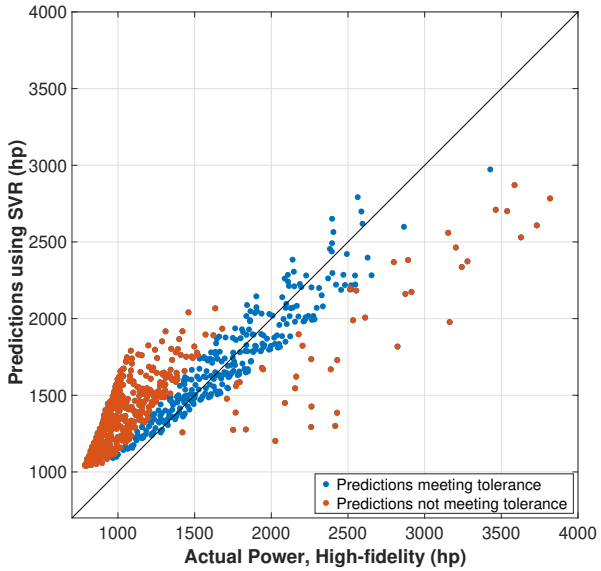

Figure 26: Actual vs predicted power of 100 kts testing dataset, evaluated using initial data-driven model

metrics provided by evaluating the testing set on this model, while Figure 24 includes bars of RMSE for the continuous velocity model evaluated at 50 kts, 150 kts, and 200 kts.

Compared to the discrete data-driven model, which required 417 between the three velocities, the 3% reduction in data is less significant than the 35% reduction in data dependency for the low-fidelity correction model. This result confirms that the presence of the low-fidelity model does have a substantive effect in reducing the data dependency, likely due to the accurate initial guess it can provide at points in the control and velocity spaces at which data does not exist.

Interpolation at 100 kts: Figure 26 shows the predicted vs actual scatterplot of model evaluations on the testing dataset at 100 kts. Comparison of Figures 21 and 26 visually demonstrates some of the difference in modeling capabilities when interpolating. The continuous velocity low-fidelity correction model interpolated on the testing data at 100 kts with an RMSE of 340 hp of the minimum power (Fig. 24), while the RMSE of the continuous velocity purely data-driven model evaluated on the same points was 412 hp, an increase of 9%.

In Figure 21, this error was primarily accounted for by a small set of over-predictions that increased in severity for trim states with higher power requirements. Throughout the domain,

the majority of the model had reasonable agreement with the model, with 124 violations, while all of the minimum power region is accurately captured. In Figure 26, the fundamental trend is incorrectly predicted, as the pure data-driven model largely over-predicts at minimum power, and largely under-predicts at higher power requirements. 309 trim states violate the tolerances, with the majority occurring at the lowest powers, demonstrating a failure to meet the goal of increased accuracy in that region. This analysis demonstrates the extent to which an entirely data-driven approach to modeling is significantly penalized in accuracy when attempting to make generalizations based on sparse data.

While it took 6 additional points at 100 kts for the low-fidelity correction model to reduce the error to a satisfactory threshold, the purely data-driven model requires nearly four times as many to do so. The final model evaluation at 100 kts in Figure 27 shows comparable performance to the final data-driven model in Figure 22, but requires a total of 428 trim states to train, as opposed to 89.

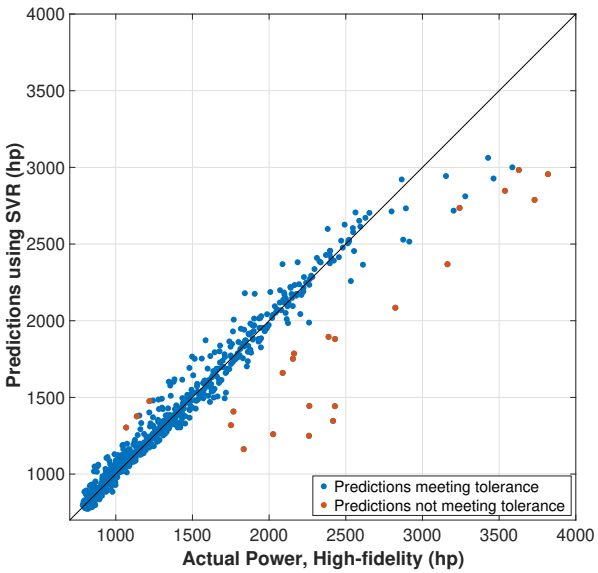

Figure 27: Actual vs predicted power of 100 kts testing dataset, evaluated using final data-driven model

CONCLUSION

This study evaluates the ability of a machine learning model to correct the low-fidelity estimate of vehicle power and main rotor trim controls for a compound helicopter with three degrees of control redundancy: main rotor speed, auxiliary thrust, and stabilator setting. A novel algorithm for iteratively enriching the training dataset based on predicted error weighted by distance in the control space is introduced. This algorithm prioritizes data generation from a computationally expensive or costly high-fidelity model in the more desirable and important low power regions of the trim space. When trained, this algorithm selects new data points to reduce the error near minimum power to within 5%, with less than 20% error globally, using the internal error model to predict which trim state will deviate most. For this purpose, a support vector regression-based error model is shown to slightly out-perform a neural network and a Gaussian regression model.

The analysis confirms that an accurate correction model can be trained with a sparse dataset the data of 26-61 states (depending on flight speed). The tolerance specification is met with a 94-96% success rate, which indicates the percent of testing data which actually fell within the intended weighted error tolerance. The model is capable of accurate predictions on total power requirement, as well as the trim settings for the three main rotor controls.

By integrating the models into a single continuous model with velocity as an additional input, the machine learning model is shown capable of generalizing trends on the error correction between flight speeds and reduce the total data dependency by 30%, with a slight improvement to success rate in the testing data (95-98%). Furthermore, the ability for interpolation to additional flight speeds. A model trained only on 83 trim states between 50, 150, and 200 kts is used to generalize to 100 kts. While the model performs 6% worse than the low-fidelity model without any correction, it only requires 6 additional trim states at 100 kts to outperform the model trained on 61 trim states at just 100 kts.

Finally a comparison is made to a purely data-driven model in order to isolate and evaluate the contribution of the low-fidelity model. Training a model to predict power solely on the high-fidelity data requires 2.1-4.9 times as many trim states, while the fraction of trim states within the testing data that meet the tolerance is reduced to 80-93%. The absence of the low-fidelity model also reduces the ability of the model to interpolate, with the error of the interpolations at 100 kts exceeding the low-fidelity correction based model by 9%.

Author Contact:
Gaurav Makkar, makkag@rpi.edu
Jean-Paul Reddinger, jean-paul.f.reddinger.civ@mail.mil
Farhan Gandhi, gandhf@rpi.edu
Fotis Kopsaftopolous, kopsaf@rpi.edu

REFERENCES

1. Healy, R., Gandhi, F., Mistry, M., and Duffy, M., "A Computational Investigation of Multi-Rotor Interactional Aerodynamics with Hub Lateral and Longitudinal Canting," VFS 76th Annual Forum, Virginia Beach, VA, 2020.

2. Healy, R., Misiorwoski, M., and Gandhi, F., "A Systematic CFD-Based Examination of Rotor-Rotor Separation Effects on Interactional Aerodynamics for Large eVTOL Aircraft," *Leading Edge*, Vol. 2, 2019, pp. 4.

3. Walter, A., McKay, M., Niemiec, R., Gandhi, F., and Ivler, C., "Hover Handling Qualities of Fixed-Pitch,

Variable-RPM Quadcopters with Increasing Rotor Diameter," 76th VFS Annual Forum, Virginia Beach, VA, 2020.

4. Bahr, M., McKay, M., Niemiec, R., and Gandhi, F., "Handling Qualities Assessment of Large Variable-RPM Multi-Rotor Aircraft for Urban Air Mobility," 76th VFS Annual Forum, Virginia Beach, VA, 2020.

5. Ivler, C. M., Truong, K., Kerwin, D., Otomize, J., Parmer, D., Tischler, M. B., and Gowans, N., "Development and Flight Validation of Proposed Unmanned Aerial System Handling Qualities Requirements," Proceedings of the 76th Annual Forum, October 2020.

6. Jacobellis, G., Gandhi, F., and Floros, M., "Using Control Redundancy for Power and Vibration Reduction on a Coaxial Rotor Helicopter at High Speeds," *Journal of the American Helicopter Society*, Vol. 64, (3), 2019, pp. 1–15.

7. Reddinger, J.-P., Gandhi, F., and Kang, H., "Using control redundancy for power and vibration reduction on a compound helicopter at high speeds," *Journal of the American Helicopter Society*, Vol. 63, (3), 2018, pp. 1–13.

8. Reddinger, J.-P., and Gandhi, F., "Physics-based trim optimization of an articulated slowed-rotor compound helicopter in high-speed flight," *Journal of Aircraft*, Vol. 52, (6), 2015, pp. 1756–1766.

9. Reddinger, J.-P., and Gandhi, F., "Response Surface Estimation of Trim Controls for a Compound Helicopter with Control Redundancy," 72th VFS Annual Forum, West Palm Beach, Florida, 2016.

10. Thorsen, A. T., Horn, J. F., and Ozdemir, G. T., "Use of redundant controls to enhance transient response and handling qualities of a Compound Rotorcraft," 70th American Helicopter Society International Annual Forum 2014, 2014.

11. Vayalali, P., McKay, M., and Gandhi, F., "Redistributed Pseudoinverse Control Allocation for Actuator Failure on a Compound Helicopter," 76th VFS Annual Forum, Virginia Beach, VA, 2020.

12. Ozdemir, G. T., Horn, J. F., and Thorsen, A. T., "In-Flight Multi-Variable Optimization of Redundant Controls on a Compound Rotorcraft," AIAA Guidance, Navigation, and Control (GNC) Conference, 2013.

13. Forrester, A. I., Sóbester, A., and Keane, A. J., "Multi-fidelity optimization via surrogate modelling," *Proceedings of the royal society a: mathematical, physical and engineering sciences*, Vol. 463, (2088), 2007, pp. 3251–3269.

14. Viana, F. A., Simpson, T. W., Balabanov, V., and Toropov, V., "Special section on multidisciplinary design optimization: metamodeling in multidisciplinary design optimization: how far have we really come?" *AIAA journal*, Vol. 52, (4), 2014, pp. 670–690.

15. Pinti, O., Niemiec, R., Oberai, A. A., and Gandhi, F., "A Multi-Fidelity Approach to Predicting Rotor Aerodynamic Interactions," AIAA AVIATION 2020 FORUM, 2020.

16. Perry, D. J., Kirby, R. M., Narayan, A., and Whitaker, R. T., "Allocation strategies for high fidelity models in the multifidelity regime," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 7, (1), 2019, pp. 203–231.

17. Narayan, A., Gittelson, C., and Xiu, D., "A stochastic collocation algorithm with multifidelity models," *SIAM Journal on Scientific Computing*, Vol. 36, (2), 2014, pp. A495–A521.

18. Robinson, T., Eldred, M. S., Willcox, K. E., and Haimes, R., "Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping," *AIAA journal*, Vol. 46, (11), 2008, pp. 2814–2822.

19. Morelli, E. A., and Cooper, J., "Frequency-domain method for automated simulation updates based on flight data," *Journal of Aircraft*, Vol. 52, (6), 2015, pp. 1995–2008.

20. Cooper, J., Schlapkohl, T., Morelli, E., Roark, S., Gaublomme, D., Roberts, B., and Xiao, S., "Integrated Methods to Model and Update Rotorcraft Simulations with Experimental Data," 70th Annual Forum of the American Helicopter Society, Montréal, Québec, Canada, 2014.

21. Glaz, B., Goel, T., Liu, L., Friedmann, P. P., and Haftka, R. T., "Multiple-surrogate approach to helicopter rotor blade vibration reduction," *AIAA journal*, Vol. 47, (1), 2009, pp. 271–282.

22. Gogulapati, A., Friedmann, P., and Martins, J., "Optimization of flexible flapping-wing kinematics in hover," *AIAA Journal*, Vol. 52, (10), 2014, pp. 2342–2354.

23. Krishnamurthi, J., and Gandhi, F., "Flight Simulation and Control of a Helicopter Undergoing Rotor Span Morphing," *Journal of the American Helicopter Society*, Vol. 63, (1), 2018, pp. 1–20.

24. Ballin, M. G., "Validation of a Real-Time Engineering Simulation of the UH-60A Helicopter," *NASA TM-88360*, 1987.

25. Peters, D. A., and HaQuang, N., "Dynamic Inflow for Practical Applications," *Journal of the American Helicopter Society*, Vol. 33, October 1988, pp. 64–66.

26. Saberi, H., Khoshlahjeh, M., Ormiston, R. A., and Rutkowski, M. J., "Overview of RCAS and application to advanced rotorcraft problems," American Helicopter Society 4th Decennial Specialists' Conference on Aeromechanics, San Francisco, CA, 2004.

27. Haas, D. J., Milano, J., and Flitter, L., "Prediction of helicopter component loads using neural networks," *Journal of the American helicopter society*, Vol. 40, (1), 1995, pp. 72–82.

28. Zaman, T., and Bayoumi, A. E., "Analysis of Health and Usage Monitoring System (HUMS) Users' Perspective towards Mission Benefits Using Regression Analysis," Proceedings of AHS 70th Annual Forum.

29. Watanabe, T., and Johnson, E. N., "Automatic landing on a moving platform using deep neural network estimation," 7th AHS Technical Meeting on VTOL Unmanned Aircraft Systems and Autonomy, 2017.

30. Rasmussen, C., and Williams, C., *Gaussian processes for machine learning*, MIT Press, Cambridge, MA, 2006.

31. Huang, T.-M., Kecman, V., and Kopriva, I., *Kernel based algorithms for mining huge data sets*, Vol. 1, Springer, 2006.

32. Fan, R.-E., Chen, P.-H., Lin, C.-J., and Joachims, T., "Working set selection using second order information for training support vector machines." *Journal of machine learning research*, Vol. 6, (12), 2005.

33. Schwabacher, M., "A survey of data-driven prognostics," *Infotech@ Aerospace*, 2005, p. 7002.

34. Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H., "Remaining useful life estimation–a review on the statistical data driven approaches," *European journal of operational research*, Vol. 213, (1), 2011, pp. 1–14.

35. Kendoul, F., "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, Vol. 29, (2), 2012, pp. 315–378.