Homogeneous Parametric Modeling of Airloads

Finn Matras
Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
finn.matras@ntnu.no

Dirk Peter Reinhardt
Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
dirk.p.reinhardt@ntnu.no

Kristoffer Gryte
Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
kristoffer.gryte@ntnu.no

Morten Dinhoff Pedersen
Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
morten.d.pedersen@ntnu.no

***Abstract*—This work proposes two parametric modeling strategies for steady aerodynamic forces. We point out that airloads are homogeneous and introduce a parametrization based on spherical harmonics and a neural network. The parametrization using spherical harmonics enables an analogue of frequency-based truncation and a variation on the Singular Value Decomposition (SVD), constituting an orthogonal decomposition of the modeled airloads. Since neural networks are universal function approximators, the model based on this allows for more flexible parametrizations, including actuations and model inversions. Both parametrization strategies are showcased for model identification and reduction purposes, highlighting their strengths and weaknesses.**

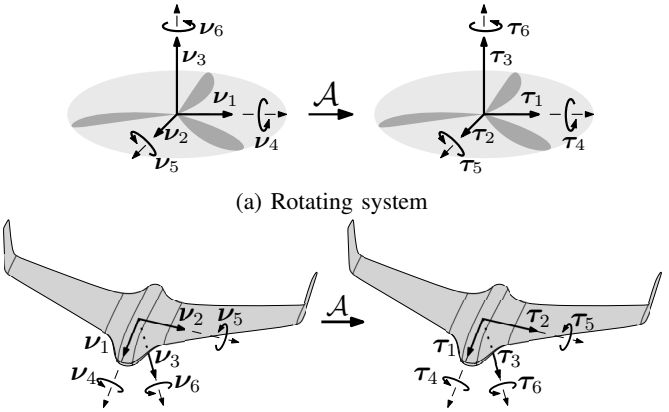
***Index Terms*—Airloads, Homogeneity, Spherical Harmonics, Neural Network, SVD**

I. Introduction

As in our previous work, [1], we seek a general function, $\mathcal{A}$, describing the input-output map from the generalized relative velocities $\boldsymbol{\nu}$ in $\mathbb{R}^6$ to the generalized aerodynamic forces, $\boldsymbol{\tau}$, acting on a rigid body in $\mathbb{R}^6$, from here on called airloads, see Fig. 1. As will become apparent later, this relation also involves a handful of constants such as the speed of sound c, the fluid density ρ, viscosity μ, and characteristic length L. Extending our previous work [1], we propose a parametrization that also allows for actuations $\boldsymbol{u} \in \mathbb{R}^{n_u}$:

$$\boldsymbol{\tau} = \mathcal{A}(\boldsymbol{\nu}, \boldsymbol{u}, c, \rho, \mu, L). \tag{1}$$

The formulation in (1) is quite general and is often solved using high-fidelity methods with significant computational effort, such as Computational Fluid Dynamics (CFD) and Blade Element Theory (BET). These approaches are often used for modeling wind turbines [2], [3], helicopters [4] and small aircraft [5]. The problem with this is that the computational demand either renders the evaluation of the aerodynamic model prohibitive to use in online control-oriented applications or requires expensive computing platforms. This leads to the necessity of finding parametrizations in terms of simpler, well-known functions that capture the dominant effects of the high-fidelity model with sufficient accuracy. Such parametrizations



(a) Rotating system

(b) Flying object

Fig. 1: Example illustration of the desired transformation from generalized relative velocities, $\boldsymbol{\nu}$, to generalized forces, $\boldsymbol{\tau}$, for two systems.

should ideally reflect the properties of the underlying system and allow for model truncation. Under mild assumptions, airloads exhibit homogeneity of degree two in the generalized velocities, i.e., any scaling of $\boldsymbol{\nu}$ results in a scaling of $\boldsymbol{\tau}$ by the same factor squared. We show this by utilizing the Buckingham π theorem in Section II.

The homogeneity gives us the following advantages:

- The space spanned by $\boldsymbol{\nu}$ is reduced from $\mathbb{R}^6$ to $\mathbb{R}^5$ through the removal of a known nonlinear scaling. This restricts the input to $\mathbb{S}^5$.
- The normalized homogeneous inputs, $\hat{\boldsymbol{\nu}}$, are in the range $[-1, 1]$ and can be directly used as inputs to neural networks and other learning methods.
- It makes training more efficient and makes it easier for neural networks to approximate the desired map, [6].
- It allows us to utilize an expansion defined in special orthonormal functions defined on the unit hypersphere.

Some authors, such as [7], use a homogeneous parametriza-

Cite as: F. Matras, D. P. Reinhardt, K. Gryte, and M. Dinhoff Pedersen, "Homogeneous Parametric Modeling of Airloads", *Syst. Theor. Control Comput. J.*, vol. 3, no. 1, pp. 1–11, Jun. 2023.
DOI: 10.52846/stccj.2023.3.1.44

tion, but the fact that the resulting model is homogeneous is not stated explicitly. Other parametrizations such as those presented in [8], [9], [10] and [11] do not enforce homogeneity, even though it is reasonable to assume that the modeled systems exhibit this property.

The key contributions of this work are:

- Dimensional analysis proving that airloads are homogeneous.
- Two parametric modeling strategies, both relying on the inherent property of homogeneity:
  - A modeling strategy using a neural network with weights and biases as parameters. This parameterization allows for the inclusion of actuations.
  - A modeling strategy based on a spherical harmonic expansion, where the expansion coefficients serve as parameters.
- Significant improvements in computational complexity compared to existing high-fidelity solvers such as BET, making the models suitable for simulation and model-based control.

An overview of the parametrization process is shown in Fig. 2. The paper is structured accordingly, and both parametrization strategies are treated equally. Section II states the main assumptions and demonstrates the homogeneity property, Fig. 2(b). The proposed parametrizations using a neural network, Fig. 2(c), and spherical harmonics, Fig. 2(d), are shown in Section III and Section IV, respectively. Two applications, one for model reduction and one for model identification, are showcased in Section V.

We note that the methodology is applicable to other systems obeying homogeneity. Furthermore, the proposed method is generally applicable to any input dimensionality. However, due to the curse of dimensionality, special considerations must be made. We showcase efficient models with inputs in $\mathbb{R}^3$ and $\mathbb{R}^5$.

II. Homogeneity of Airloads

This section demonstrates the homogeneity of airloads and states the necessary assumptions for this property to hold.

A. Assumptions

Certain widely applicable assumptions have to be made in order for the airloads function to be homogeneous, namely:

- Steady flow
- Subsonic motion
- Sufficiently large Reynolds number

In practice, this excludes micro aerial vehicles and aircraft operating close to or above the speed of sound. Large systems such as wind turbines comply with the assumptions. The same holds for small aircraft [5].

B. Dimensional analysis

Let the vector of forces, the vector of torques, and the generalized force-torque vector in $\mathbb{R}^6$ be denoted by

$$\boldsymbol{F} \triangleq \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}, \quad \boldsymbol{M} \triangleq \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}, \quad \boldsymbol{\tau} \triangleq \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \end{bmatrix}, \tag{2}$$

respectively. The rotational and generalized velocities in $\mathbb{R}^6$ that generate the forces and torques are denoted by

$$\boldsymbol{v} \triangleq \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad \boldsymbol{\omega} \triangleq \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad \boldsymbol{\nu} \triangleq \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix}. \tag{3}$$

Working with generalized forces and velocities with consistent units will be advantageous. To this end, we define a characteristic length L and utilize it to form

$$\boldsymbol{K}_L \triangleq \begin{bmatrix} \mathbb{I}_{3x3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3x3} & L\mathbb{I}_{3\times3} \end{bmatrix}, \quad \tilde{\boldsymbol{\tau}} \triangleq \boldsymbol{K}_L\boldsymbol{\tau}, \quad \tilde{\boldsymbol{\nu}} \triangleq \boldsymbol{K}_L^{-1}\boldsymbol{\nu}, \tag{4}$$

where $\mathbb{I}_{3\times3}$ is the 3×3 identity matrix and $\boldsymbol{0}_{3x3}$ is a 3×3 matrix populated with zeros. After the multiplication by $\boldsymbol{K}_L$, the elements of $\tilde{\boldsymbol{\tau}}$ and $\tilde{\boldsymbol{\nu}}$ will have the same units. Note that with $L = 1\,\mathrm{m}$ these variables are numerically equivalent to $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$.

We consider steady state airloads acting on a rigid body where one can assume that $\boldsymbol{\tau}$ at any given time solely depends on $\boldsymbol{\nu}$, and possibly $\boldsymbol{u}$ at the same time, and a handful of relevant constants. We only consider dimensionless actuations, so they can be omitted here. Following the general dimension analysis strategy as presented in [12], there must exist a function such that

$$f(\tilde{\boldsymbol{\tau}}, \tilde{\boldsymbol{\nu}}, \|\tilde{\boldsymbol{\nu}}\|_2, c, \rho, \mu, L) = 0. \tag{5}$$

This function can be transformed into a dimensionless form by utilizing the Buckingham π theorem with cancellation variables $\|\tilde{\boldsymbol{\nu}}\|_2$, ρ and L. Doing so yields

$$f_\Pi\left(\frac{\tilde{\boldsymbol{\tau}}}{\|\tilde{\boldsymbol{\nu}}\|_2^2 L^2 \rho}, \frac{\|\tilde{\boldsymbol{\nu}}\|_2}{c}, \frac{\mu}{\|\tilde{\boldsymbol{\nu}}\|_2 L \rho}, \frac{\tilde{\boldsymbol{\nu}}}{\|\tilde{\boldsymbol{\nu}}\|_2}\right) = 0, \tag{6}$$

where the third and fourth terms are the Mach and reciprocal Reynolds number, respectively. Both dimensionless numbers can be set to zero under the stated assumptions. Discarding these terms, there must be a function $\hat{\mathcal{A}}$ such that

$$\frac{\tilde{\boldsymbol{\tau}}}{\|\tilde{\boldsymbol{\nu}}\|_2^2 L^2 \rho} = \hat{\mathcal{A}}\left(\frac{\tilde{\boldsymbol{\nu}}}{\|\tilde{\boldsymbol{\nu}}\|_2}\right) = \hat{\boldsymbol{\tau}}, \tag{7}$$

where $\hat{\boldsymbol{\tau}}$ is a dimensionless generalized force coefficient. One may therefore represent the generalized force as

$$\tilde{\boldsymbol{\tau}} = \|\tilde{\boldsymbol{\nu}}\|_2^2 L^2 \rho \hat{\mathcal{A}}\left(\frac{\tilde{\boldsymbol{\nu}}}{\|\tilde{\boldsymbol{\nu}}\|_2}\right). \tag{8}$$

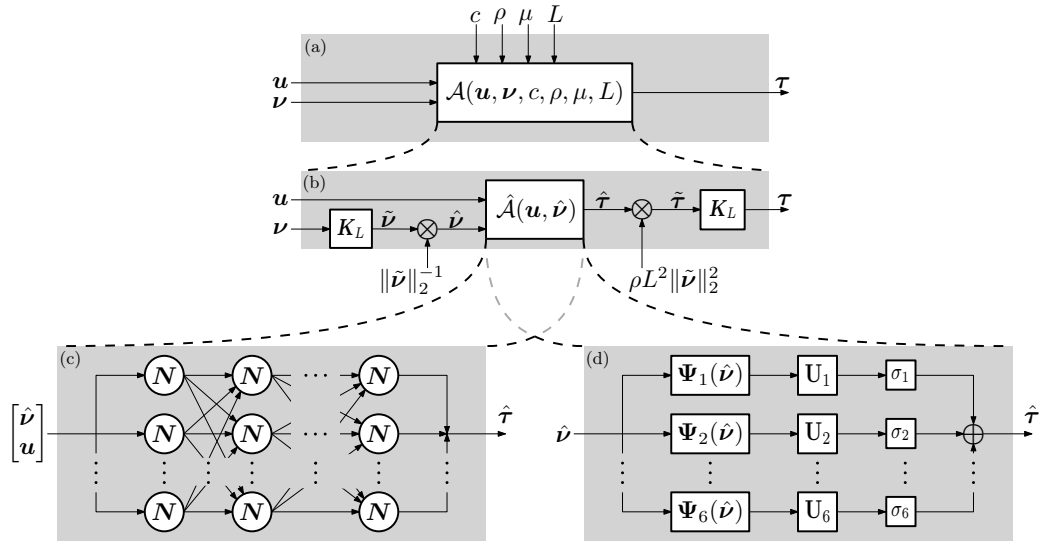


Fig. 2: Illustration of parametrization process. Generic function (a), homogeneity (b), neural network (c), and orthogonality decomposition (d). $\otimes$ indicates multiplication and $\oplus$ represents addition. Note that the neural network (c) allows for actuations.

C. Homogeneity

Let the norm of $\tilde{\boldsymbol{\nu}}$ and its normalized version be given by

$$\lambda = \|\tilde{\boldsymbol{\nu}}\|_2, \quad \hat{\boldsymbol{\nu}} = \frac{\tilde{\boldsymbol{\nu}}}{\lambda}. \tag{9}$$

Inserting these into (8) and setting $L = 1\,\mathrm{m}$ we obtain

$$\tilde{\boldsymbol{\tau}} = \rho\lambda^2 \hat{\mathcal{A}}(\hat{\boldsymbol{\nu}}) = \rho\lambda^2 \hat{\boldsymbol{\tau}}. \tag{10}$$

This equation is part of a family of functions known as homogeneous functions [13], more specifically, positively homogeneous functions with degree of homogeneity two. To our knowledge, this property is neither widely known nor used when parameterizing airloads. The homogeneity property allows us to reduce the input space of $\boldsymbol{\nu}$, $\mathbb{R}^6$, to the unit hypersphere $\mathbb{S}^5$. We then focus on modeling the nonlinear function $\hat{\mathcal{A}}(\hat{\boldsymbol{\nu}})$, with inputs on the unit hypersphere, given that the remaining terms of (10) are trivial. Fig. 2(b) shows a block diagram representation of (10).

The preceding results generalize the equations for sectional lift and drag of an airfoil, where the domain is only $\mathbb{S}^1$. However, $\hat{\mathcal{A}}(\hat{\boldsymbol{\nu}})$ can represent everything from a small section of an airfoil through the spatially integrated forces and torques over an airfoil to the time-averaged forces and torques of a rotating or flapping system.

It is worth mentioning that the dimensional analysis only requires λ to have the unit of velocity, but its magnitude can differ from the norm of the generalized velocities. In fact, any uniform scaling of λ retains the same unit.

III. Parametric Modeling Using a Neural Network

A general approach for modeling unknown functions is to use general function approximators. Neural networks are a well-known type of such approximators. Even though they are generally applicable, several criteria need to be fulfilled for them to perform well. One typical requirement is normalization, i.e., the input and output should be normalized to the range $[-1, 1]$. For the case of airloads, the scaling λ^2 would be challenging for a neural network to learn since it has no direct way of doing this. However, since the generalized velocities are normalized before evaluating $\hat{\mathcal{A}}$, as shown in Fig 2, neural networks are suitable function approximators in this framework.

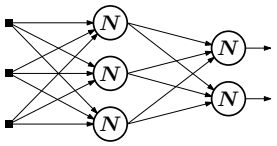


Fig. 3: Illustration of two layer fully connected neural network with three inputs and two outputs.

Since neural networks are general approximators, mixing homogeneous and nonhomogeneous inputs in the same model is possible. This allows us to include actuated aerodynamic surfaces and makes for a compact and robust parametrization.

Fig. 3 illustrates a small neural network with two layers. The three inputs on the left are marked with dots, and each neuron, $\boldsymbol{N}$, contains weights of all inputs, biases, and an activation function. NN abbreviates a parametrization based on a neural network.

A. Model formulation

A neural network, for example, as illustrated in Fig. 3, consists of several layers of neurons connected by weights. Each layer can be mathematically formulated as

$$\boldsymbol{N}_n(\boldsymbol{x}) = f(\boldsymbol{W}_n \boldsymbol{x} + \boldsymbol{b}_n), \tag{11}$$

where $\boldsymbol{W}_n$ is a matrix of weights with the inputs along its columns and output neurons along its rows. $\boldsymbol{x}$ is a vector

of inputs, $\boldsymbol{b}_n$ is a vector of biases and finally $f(\cdot)$ is an activation function. The activation functions are selected based on knowledge of the system, user preference, and desired model behavior. More information about neural networks can be found in the extensive literature covering this topic, such as in [6]. The resulting model then recursively calls these layer functions until the output is generated. The overall function of a NN with N layers is given by

$$\hat{\mathcal{A}}\left(\begin{bmatrix}\hat{\boldsymbol{\nu}}\\ \boldsymbol{u}\end{bmatrix}\right) = \boldsymbol{N}_1(\boldsymbol{N}_2(...(\boldsymbol{N}_N\left(\begin{bmatrix}\hat{\boldsymbol{\nu}}\\ \boldsymbol{u}\end{bmatrix}\right))). \tag{12}$$

B. Model parametrization

Parametrizing the model consists of several steps:

1) Sample point selection: Neural networks, in general, need large amounts of data to train properly. The quantity of data required depends highly on the problem at hand and the design of the neural network. In some instances, Monte Carlo sampling of the input space is a viable option, while quadratures also might be a good idea if they are available. The sample point selections should, if possible, be an iterative process and tuned based on the training behavior of the neural network. In addition, the data should be separated into a training set and a test set so one can evaluate and stop training at the correct time.

The homogeneity allows us to normalize the input and scale the output accordingly. However, the output might still have a large range. This can be combated by manually normalizing the output data and then applying the reciprocal of this normalization to the neural network's output.

2) Airload evaluation: Evaluation of the airloads on the sample points can be done in many ways. Numerical codes such as CFD and BET can be used when a strictly computational methodology is preferred. At the same time, testing in a wind tunnel might be advantageous when such facilities are available. Let the subscript $\cdot_k$ denote $\cdot$ evaluated at sample point k, such that $\boldsymbol{\tau}_k$ is the airload evaluated at $\boldsymbol{\nu}_k$ and possibly actuations $\boldsymbol{u}_k$.

3) Parameter estimation: All the weights $\boldsymbol{W_n}$ and biases $\boldsymbol{b}$ of all the layers must be optimized. Since this quickly becomes a large optimization problem, it is computationally challenging to optimize a cost function. Instead, backpropagation of a loss is used. The loss function is equivalent to the cost function in a typical optimization problem, and we can use it to dictate how we want the model to behave. Using a weighted mean squared error loss function is a good start. By assigning weights $\mathbf{W}_k$ to the samples, we can specify samples that are more important to fit, effectively favoring a certain operating region over the rest. Mathematically, we have

$$\text{loss}(\boldsymbol{\nu}_k, \boldsymbol{u}_k, \boldsymbol{\tau}_k) = \frac{1}{k}\sum_{k=1}^{K} \mathbf{W}_k \|\boldsymbol{\tau}_k - \boldsymbol{N}(\boldsymbol{\nu}_k, \boldsymbol{u}_k)\|_2^2. \tag{13}$$

This loss is then used in a backpropagation algorithm to update the weights and biases.

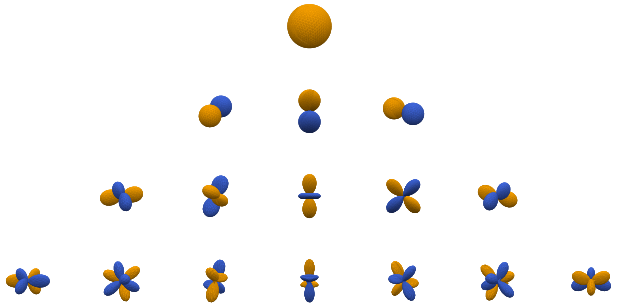


Fig. 4: Illustration of spherical harmonics up to order and degree three. Blue is negative, orange is positive.

C. Computational performance

The computation time of a NN depends on the number of input channels and the activation function. However, for reference, we include that the NN in Fig. 3 with $tanh(\cdot)$ as activation functions has a computation time of $320\,\text{ns}$. The runtime test was performed using Julia [14] v. 1.8.3 running on Windows 11 on an AMD Ryzen 9 4900HS laptop processor.

IV. Parametric Modeling Using Spherical Harmonics

Hyperspherical harmonics are a natural choice for modeling homogeneous functions of degree zero, which constitute a homogeneous, complete, and orthogonal set of basis functions. While the general methodology applies to hyperspherical harmonics for any dimension, and these can be found [15], it is only in the case of up to $\mathbb{S}^2$ that there exist suitable computational algorithms in the literature. In the ongoing, we shall confine the argument of the function to $\mathbb{S}^2$, where the basis functions are known as spherical harmonics [16, §14.30]. Consequently, we only consider three well-chosen input dimensions, which is enough for numerous practical applications such as wind turbines, airplanes, and multicopters.

Fig. 4 illustrates spherical harmonics up to order and degree three. The figure also indicates the presence of sorting by frequency content. Multiple algorithms for computing the spherical harmonics exist, such as [17] and [18]. We made a custom implementation, but this is beyond the scope of this work and might be subject to a different publication. A model based on spherical harmonics is from here on abbreviated by SH.

A. Model formulation

The spherical harmonics are sorted by their order m and degree l. However, for computational purposes, it will be advantageous to sort them according to a one-dimensional index j. Let f be a mapping such that $j = f(l, m)$ and n the total number of indexed harmonics.

By using spherical harmonics as basis functions, it is possible to expand the function $\hat{\mathcal{A}}$ in the following fashion

$$\hat{\mathcal{A}}(\hat{\boldsymbol{\nu}}) = \begin{bmatrix} \sum_{j=1}^{n} \mathbf{B}_{1,j}\psi_j(\hat{\boldsymbol{\nu}}) \\ \sum_{j=1}^{n} \mathbf{B}_{2,j}\psi_j(\hat{\boldsymbol{\nu}}) \\ \vdots \\ \sum_{j=1}^{n} \mathbf{B}_{6,j}\psi_j(\hat{\boldsymbol{\nu}}) \end{bmatrix} = \mathbf{B}\boldsymbol{\psi}(\hat{\boldsymbol{\nu}}), \tag{14}$$

where $\boldsymbol{\psi}(\hat{\boldsymbol{\nu}})$ is a column vector containing the spherical harmonic basis functions, $\psi_j(\hat{\boldsymbol{\nu}})$, and $\mathbf{B}$ is a matrix where each row represents appropriately chosen parameters for the basis functions of the respective airload. Since the expansion is complete, (14) converges as n goes to infinity. However, experience shows that truncation can be applied, which makes n finite. This, in turn, allows us to formulate a finite-dimensional parameter-matrix $\mathbf{B}$, quantifying the airload function and enabling us to perform a Singular Value Decomposition (SVD). By definition we have

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top, \tag{15}$$

which inserted into (14) gives

$$\hat{\mathcal{A}}(\hat{\boldsymbol{\nu}}) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\boldsymbol{\psi}(\hat{\boldsymbol{\nu}}). \tag{16}$$

Taking inspiration from the SVD structure, we define

$$\boldsymbol{\Psi}(\hat{\boldsymbol{\nu}}) = \mathbf{V}^\top\boldsymbol{\psi}(\hat{\boldsymbol{\nu}}), \tag{17}$$

where $\boldsymbol{\Psi}(\hat{\boldsymbol{\nu}})$ represents sorted superpositions of the basis modes, super-modes for short. These super-modes are scaled by $\boldsymbol{\Sigma}$ and weighted by $\mathbf{U}$ to generate the airloads.

The final formulation then becomes

$$\hat{\mathcal{A}}(\hat{\boldsymbol{\nu}}) = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Psi}(\hat{\boldsymbol{\nu}}) \tag{18}$$

and is also shown schematically in Fig. 2(c). This SVD-inspired formulation enables one to perform optimal truncation if necessary.

1) Orthogonality: Spherical harmonics are orthogonal, i.e.

$$\int \boldsymbol{\psi}(\hat{\boldsymbol{\nu}})\boldsymbol{\psi}(\hat{\boldsymbol{\nu}})^\top d\Omega = \mathbb{I}_{n\times n} \tag{19}$$

where Ω is the solid angle. A similar approach for the super-modes reveals that

$$\begin{aligned} &\int \boldsymbol{\Psi}(\hat{\boldsymbol{\nu}})\boldsymbol{\Psi}(\hat{\boldsymbol{\nu}})^\top d\Omega = \int \mathbf{V}^\top\boldsymbol{\psi}(\hat{\boldsymbol{\nu}})\boldsymbol{\psi}(\hat{\boldsymbol{\nu}})^\top\mathbf{V}d\Omega = \\ &\mathbf{V}^\top \int \boldsymbol{\psi}(\hat{\boldsymbol{\nu}})\boldsymbol{\psi}(\hat{\boldsymbol{\nu}})^\top d\Omega\mathbf{V} = \mathbf{V}^\top\mathbb{I}_{n\times n}\mathbf{V} = \mathbb{I}_{n\times n}, \end{aligned} \tag{20}$$

which means that the SVD transformation preserves orthonormality. Thus, $\boldsymbol{\Psi}(\hat{\boldsymbol{\nu}})$ consists of orthonormal basis functions.

B. Model parametrization

The SH was found by obtaining numerical airload data at some points on $\mathbb{S}^2$ and optimizing the parameters. In general, the procedure can be divided into the following steps:

1) Sample Point Selection: An efficient candidate for sample points on $\mathbb{S}^2$ is given by the Lebedev quadrature. The order of the Lebedev quadrature only has to be twice that of the highest spherical harmonic, [19]. The major limitation of this method is that the highest order available is 131, [20]. However, convergence is usually quick, and spherical harmonics of order and degree lower than 30 suffice for the examples presented in this work.

In addition to the sample points given by the Lebedev quadrature, it is desirable to utilize a higher density of sample points in the intended operating region. Let $\boldsymbol{\nu}_k$ denote sample points for $k \in \{1, 2, \ldots, K\}$.

2) Airload Evaluation: Evaluation of the airloads at the sample points can be done the same way as for the NN. Again, let the subscript $\cdot_k = \cdot|_{\boldsymbol{\nu}_k}$ denote $\cdot$ evaluated at sample point $\boldsymbol{\nu}_k$, such that $\boldsymbol{\tau}_k$ is the airload evaluated at $\boldsymbol{\nu}_k$.

3) Parameter Estimation: The parameter matrix $\mathbf{B}$ is found by solving the least squares problem with regularization,

$$\min_{\mathbf{B}} \quad \sum_{k=1}^{K} \mathbf{W}_k\|\boldsymbol{\tau}_k - \tilde{\boldsymbol{\tau}}_k\|_2^2 + \mathbf{R}^\top\|\mathbf{B}_k\|_1, \tag{21}$$

where $\mathbf{W}$ and $\mathbf{R}$ are sample weighting and regularization vectors, respectively.

The sample weights are used to prioritize the samples in a user-specified way. At the same time, the regularization vector, together with the regularization norm, controls the shape and distribution of the model parameters, $\mathbf{B}$. By tuning the regularization, one can also prescribe desired frequency content. $L1$ regularization is used in (21) to enforce model sparsity since the basis functions are not guaranteed to be linearly independent on the evaluated sample points. Alternatively, using $L2$ regularization gives similar performance but does not inherit the computational advantage of a sparse model.

If only the Lebedev quadrature is used as sample points, the regularization weight $\mathbf{R}$ can be set to zero. The Lebedev quadrature also supplies sample weights $\mathbf{W}$. The optimization problem in (21) can be solved using industry-standard and openly available numerical tools.

C. Computational Performance

The average computation time of one basis function is $\sim 4\,\mathrm{ns}$, summing up to about $900\,\mathrm{ns}$ for a model evaluation running on a single CPU core given a model of maximal order and degree 15. The runtime tests were performed using the same hardware and software as in Section III-C.

V. Examples

Two case studies are considered to showcase the methods' versatility. Fig. 1 illustrates the two physical bodies and their coordinate systems.

A. NREL 5MW Wind Turbine Rotor

The NREL 5 MW reference wind turbine described in [21] will be used to illustrate the model reduction approach on a rotating system. The airloads in the absence of induced velocities are found using BET, which computes the full $\boldsymbol{\tau}$

averaged over one rotation. The reader may consult [4] for details about the BET algorithm.

The generalized velocities and forces are shown in Fig. 1a, and the inputs were chosen to be the wind velocity parallel to the axis of rotor rotation, $\boldsymbol{\nu}_3$, the wind velocity perpendicular to the axis of rotation in the upwind direction, $\boldsymbol{\nu}_1$, as well as the rotational velocity about the axis of rotation $\boldsymbol{\nu}_6$.

1) Modeling Using a Neural Network: With this reduced input set, the sample points were selected as evenly spaced points on $\mathbb{S}^2$ using a Lebedev quadrature of order 101, which resulted in 3470 sample points. Additionally, 900 sample points in the operating region were selected and weighted 100 times stronger than the remaining sample points. The airloads were computed for the sample points. Due to the reduced input dimensions, we only have four nonzero outputs, namely F_x, F_z, M_x, and M_z.

Fig. 5 shows the NN. There are three inputs, 40 neurons in the first layer with a $tanh(\cdot)$ activation function, then two layers of 80 neurons and one layer with 20 neurons each with $leakyRELU(\cdot)$ activation functions follow, before the output layer with four neurons and unit activation.

The NN was implemented in Flux [22] [23] using the AdaBelief [24] backpropagation algorithm.

The additional samples in the nominal operating region and higher weights are added to gain a higher fidelity in the area where the model will be primarily used. For a wind turbine, this region is characterized by the torque being in the same direction as the rotational velocity and is highlighted in red in Fig. 6a. Red represents negative torque, the only region in which the torque is negative at negative rotational velocities. This is the only region in which the rotor produces power. The two big spheres on the bottom and the top have torques opposing the direction of rotation, which means that they consume energy. These would be the operating regions for propellers that are used for propulsion.

Even though a neural network theoretically would be able to parametrize a wind turbine rotor with the full $\boldsymbol{\nu}$, this would be a challenging task due to the curse of dimensionality. Not only for training the NN, but also for obtaining the data required for training and testing.

2) Modeling Using Spherical Harmonics: A parametrization up to degree 30 is selected, and subsequently, a Lebedev quadrature with degree 61 is chosen. Additionally, 300 sample points were added in the nominal operating region.

The airloads were evaluated at the selected sample points, and the optimization problem in (21) was solved using JuMP [25] with the IPOPT solver [26].

As can be seen from the parameters in Fig. 7, the SH exhibits a strong low-pass character. The high degree is needed to accurately model the nominal operating region in which the wind turbine generates power. As before, this area is highlighted in red in Fig. 6b.

The super-modes defined in (17) are shown in Fig. 8. Fig. 8g shows the $\mathbf{U}$ matrix and singular values $\boldsymbol{\Sigma}$. It can be seen that the forces and torques in and around the y-axis are zero to

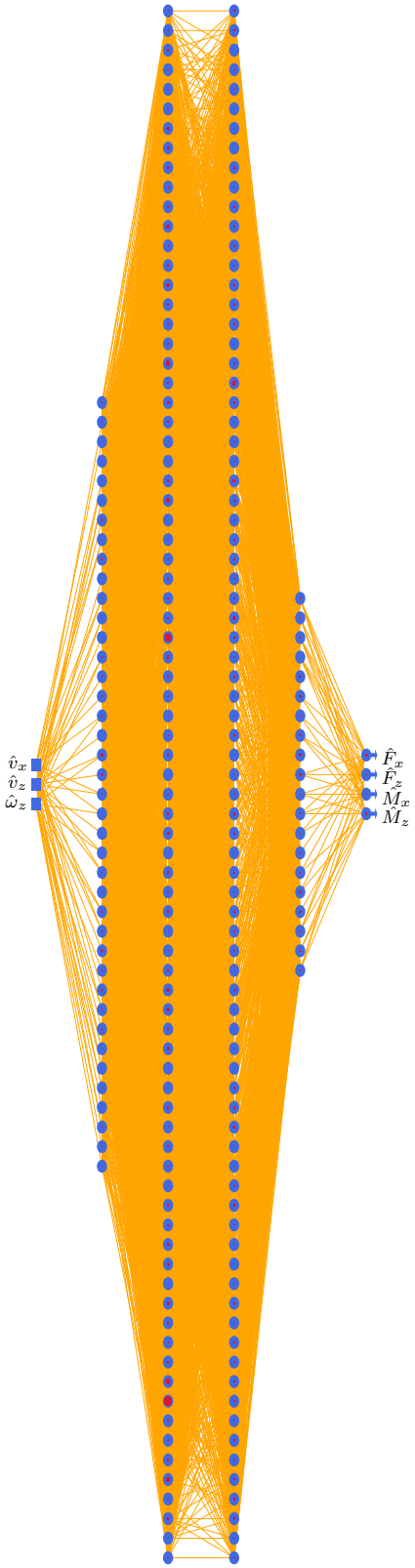
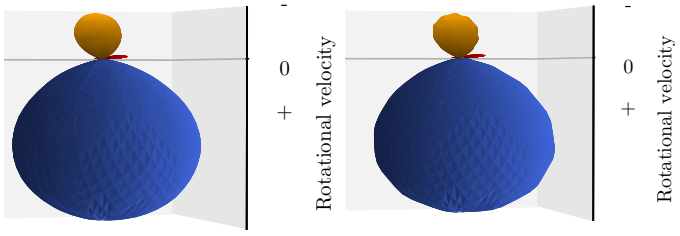


Fig. 5: Illustration of NREL 5MW NN with inputs (squares), nodes (blue circles), outputs (arrows), weights (line width), and biases (red circles).



(a) NN (b) SH

Fig. 6: Average rotor torque during one rotation predicted by NN and SH plotted in spherical coordinates with inputs sampled on the unit sphere. Red is negative.

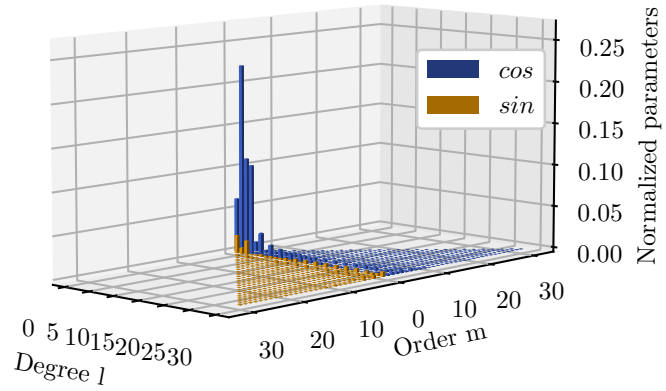
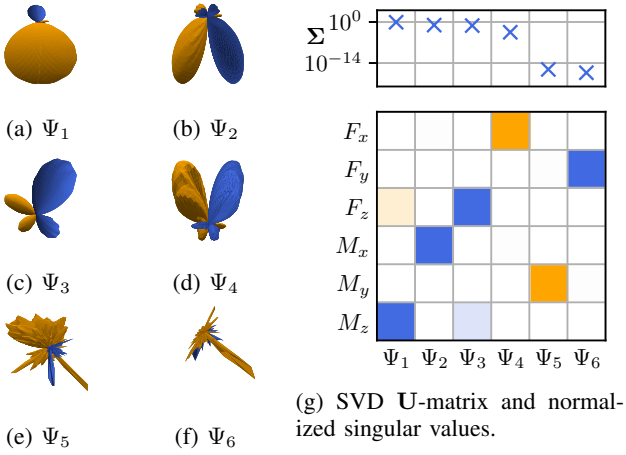


Fig. 7: Illustration of NREL 5MW model parameters $\mathbf{B}$ with respect to model degree l and order m.



(g) SVD $\mathbf{U}$-matrix and normalized singular values.

Fig. 8: Illustration of SVD decomposition of NREL 5MW reference turbine airloads.

floating-point accuracy. This is because the horizontal flow direction is along the x-axis.

3) Model Performance: Wind turbine performance is often evaluated by examining the Tip Speed Ratio (TSR or λ) to power coefficient (C_p) plot, as shown in Fig. 9. The plot considers inflow orthogonal to the rotor disk, a slice of the total parametrized input space $\mathbb{S}^2$ that also contains skewed inflow. The induced velocities were computed as a function

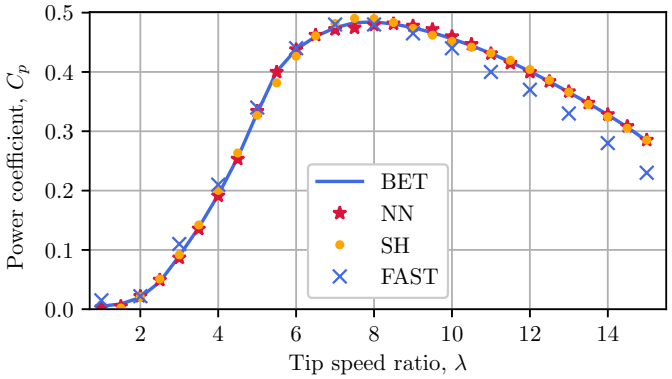


Fig. 9: Illustration of the power coefficient with respect to the tip speed ratio, also known as a $\lambda - C_p$ plot. The FAST data was obtained from [28].

of the thrust and are an approximation to the relation given in [27]. A total power loss of 5% was added to encompass nonmodeled losses. It can be seen that both parametrizations have similar and satisfactory performance.

An evaluation of the full SH takes 8.5 µs, about 300 times faster than BET with similar accuracy. The NN evaluations take 9.0 µs, which for all practical purposes, is the same as for the SH.

B. Skywalker X8 Unmanned Aerial Vehicle

An example of model identification is performed for the Skywalker X8 fixed-wing unmanned aerial vehicle using wind tunnel experimental data gathered by [8].

1) Measurements and Preprocessing: Gryte et al. [8] conducted a series of wind tunnel tests to collect measurements of the generalized aerodynamic forces when the Skywalker X8 airframe is subject to a relative linear velocity $\boldsymbol{v} \in \mathbb{R}^3$ that is within the nominal operating conditions of this type of fixed-wing UAV. The wind tunnel was operated to generate a constant homogeneous airstream with a Reynolds number in the range of the operating conditions of the Skywalker X8 from 2.5×10^5 to 5×10^5. The airframe was rotated to sweep varying angles of attack and sideslip angles at different aileron and elevator actuations. In addition to data in the operational regime, data in the stall regime was obtained.

Even with thorough measurement procedures, a slight misalignment between the body-fixed and measurement frame was found in the data. This was corrected based on arguments to exploit the symmetry concerning the longitudinal plane of the airframe. Details of the calibration procedure will appear in a separate paper [29].

The resulting samples and their angle of attack are show in Fig. 10. Note that the sideslip angles and actuations are not included in the plot for clarity.

2) Modeling Using a Neural Network: The relative linear velocities, $\boldsymbol{v}$, were selected as homogeneous inputs to the model, in addition to the nonhomogeneous elevator and aileron deflections in degrees. The output was $\boldsymbol{\tau}$.

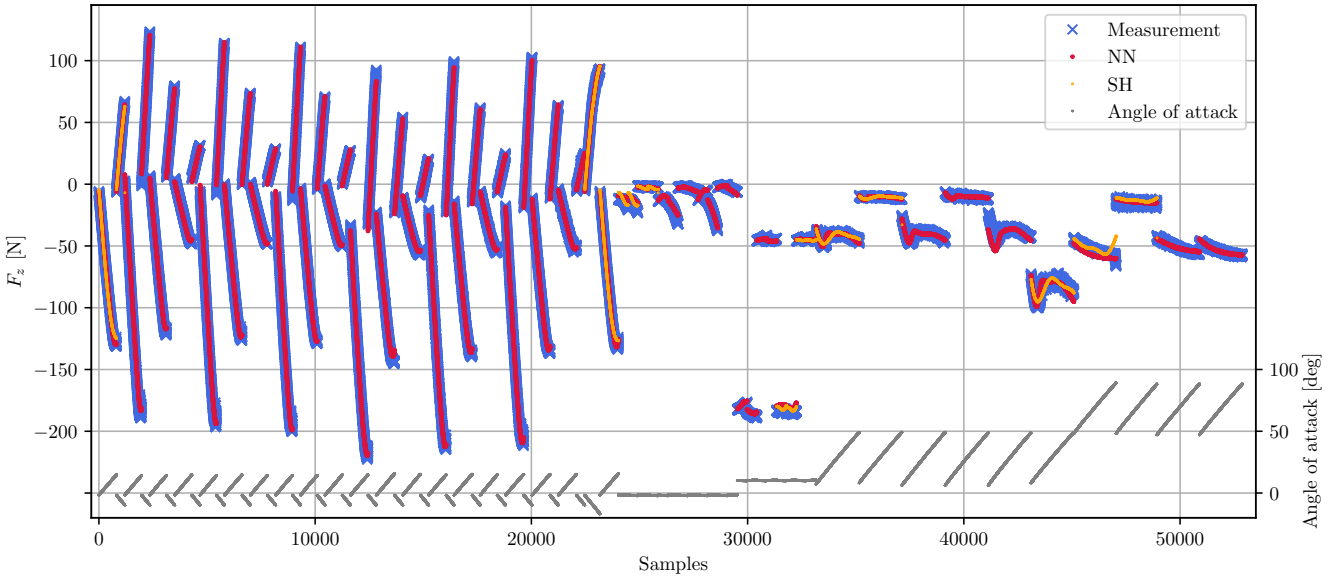


Fig. 10: Illustration of the measurement samples and fitted models.

In addition to preprocessing the data as outlined in the previous section, the data was slightly filtered by smoothing out the measurements. This was done to prevent the neural network from trying to fit the noise and instead concentrate on the underlying model behavior. To reduce the amount of data, only every third sample from the pre-stall region was used, and every fifth sample in the stall region. This effectively also weighted the pre-stall region higher than the post-stall region.

An observed advantage of homogeneity is that a significant reduction in required samples is achieved. The test data from the Skywalker X8 contained samples taken at a wide range of relative airflow velocities. However, when applying the normalization based on homogeneity, we only need samples from one airflow velocity since, after normalization, the airflow is of unit length.

Fig. 11 shows the NN fitted to the measurement data. It consists of five inputs, the first three are homogeneous inputs, and the last two are nonhomogeneous inputs. Then there is one layer with 11 neurons with $tanh(\cdot)$ activation function before the output layer with six neurons and $tanh(\cdot)$ activation function, each representing one element of $\boldsymbol{\tau}$. As before, the width of the lines represents the weights, and the red circles the biases. By analyzing the model, we can, for instance, see that $\hat{M}_x$ and $\hat{M}_z$ have common behavior since they share the top node in the first layer, which only depends on the normalized velocities $\hat{v}_x$ and $\hat{v}_y$. Furthermore, one can also verify that $\hat{F}_z$, $\hat{M}_x$, and $\hat{M}_y$ have a term which depends only on $\hat{v}_x$.

The NN was again implemented in Flux [22] [23] using the AdaBelief backpropagation algorithm [24].

Even though we only have training data for a limited range of the input variables, the resulting parametrization has no such limitations. This means that the parametrization guesses how the model behaves when we sample it outside of the range spanned by the training data. An example of this is shown

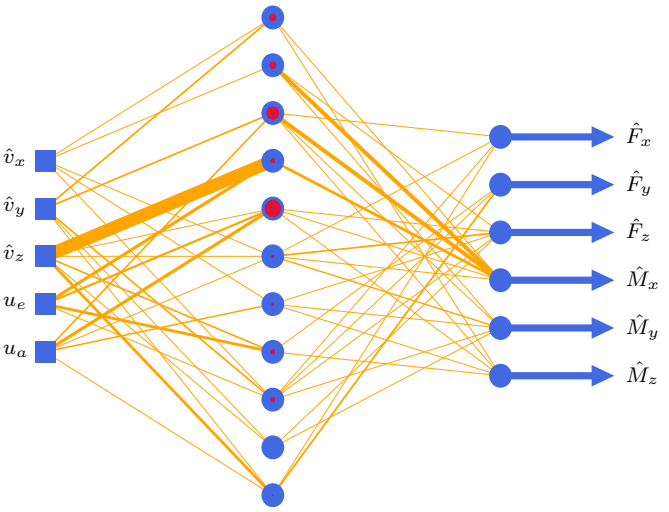
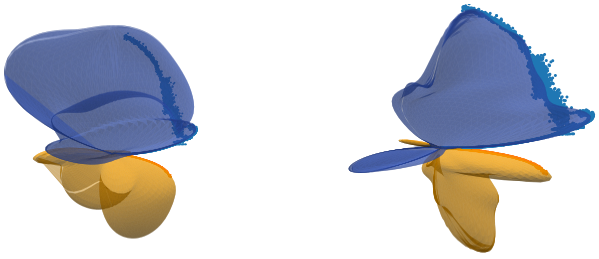


Fig. 11: Illustration of Skywalker X8 NN with inputs (squares), nodes (blue circles), outputs (arrows), weights (line width), and biases (red circles).

in Fig. 12a, where the dots represent the measurements and the transparent shape is the prediction of the NN. It might look like the model is not fitting the blue (negative) samples correctly, but this is just a flaw in the plotting. The sample points are fitted well, as will become apparent in Fig. 10.

Testing a similar neural network without utilizing the homogeneity shows an increase in the mean squared error with a factor of 10000, highlighting the effectiveness of properly normalizing the input data.

3) Modeling Using Spherical Harmonics: The three dimensions of linear velocities, $\boldsymbol{v}$, were selected as inputs to the SH, while the output was set to be $\boldsymbol{\tau}$. The coordinate system



(a) NN (b) SH

Fig. 12: Illustration of Skywalker X8 lift model using SH and NN (spheres) and measurements (dots) from [8].

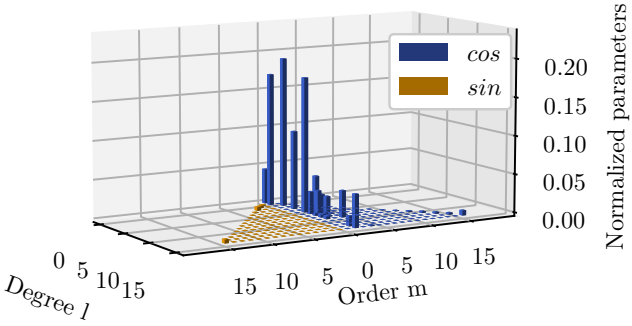
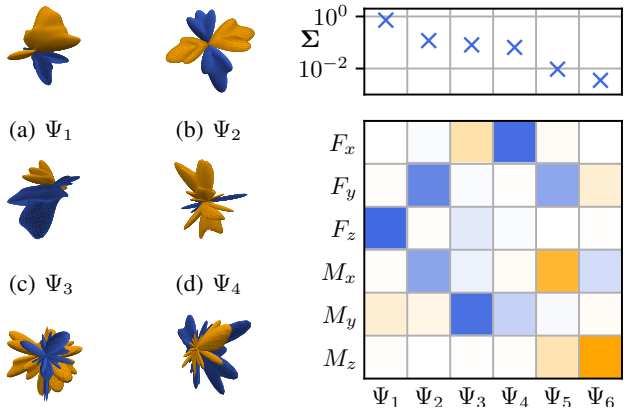


Fig. 14: Illustration of Skywalker X8 model parameters with respect to model degree l and order m.



(a) Ψ_1 (b) Ψ_2 (c) Ψ_3 (d) Ψ_4 (e) Ψ_5 (f) Ψ_6

(g) SVD $\mathbf{U}$-matrix and normalized singular values.

Fig. 13: Illustration of SVD decomposition for Skywalker X8 airloads.

is shown in Fig 1b. The data was limited to constant neutral actuations, and some outliers and areas of high measurement noise were removed. Every third sample of the remaining dataset was used to speed up computations. JuMP [25] with the Ipopt solver [26] was used to solve (21). The maximal degree of the basis functions was set to 15.

Since the optimization problem is formulated based on measurement data that only spans a small part of $\mathbb{S}^2$ it is somewhat uncertain how the model should behave outside this region. The weights for the regularization of the basis functions in (21) are therefore weighted linearly by their degree to enforce a low-frequency model. Additionally, the normalized inputs scale up measurement noise significantly when the input is close to zero. The sample errors were weighted by their normalized input norms squared to combat this.

Fig. 13 shows the resulting body-fixed model. The sparse normalized parameters of the model are shown in Fig. 14. Compared to the parameters in Fig. 7, the model convergence is slower but still satisfactory.

Fig. 12b shows the SH for lift in body coordinates and the corresponding measurements placed onto the sphere as dots. It is seen that the fitted SH acts as a low-pass filter, averaging out the noisy measurements.

Fig. 13g illustrates the U-matrix of the SVD and the corresponding normalized singular values. Most of the forces and torques have one or two dominant components. As expected from relatively smooth airload behavior, one can see that there is an inversely proportional relationship between the singular value and frequency content of the super-modes.

4) Model performance: The performances of the parametrizations are evaluated in Fig. 10 by comparing the estimates and the corresponding measurements. The SH model is only plotted for samples with neutral actuations. This is the same data as in Fig. 12b and Fig. 12a. The NN model is able to include the nonhomogeneous inputs and is therefore valid over the whole operating region, as indicated by its predictions. One can verify that both parametrizations fit the measurements well in both pre-and post-stall regimes since stall was observed to start at angles of attack above 12 deg.

Evaluation of the full SH takes about 3.1 µs, while evaluation of the NN takes 1 µs. In this case, the simple NN is faster than the SH, even though it includes the two actuations and thus constitutes a full parametrization of the measurement data.

VI. Open Questions and Future Work

The presented work has focused on analyzing generalized velocities on the unit sphere $\mathbb{S}^2$. Only the neural network can include input on higher dimensional spheres at this stage. However, it was found that the curse of dimensionality quickly becomes a limitation for practical purposes. Further investigating this issue will be an exciting topic for future work. Ideally, a general formulation for n dimensions would be found, but an extension to $\mathbb{R}^6$ can be considered a complete model with respect to $\boldsymbol{\nu}$.

Another area of future work with a significant impact on system control would be to find a good approach for model inversion. The model based on spherical harmonics is limited to $\mathbb{R}^3$, which makes inversion challenging. Under certain assumptions and limitations, a neural network can be inverted directly, yielding the desired result. Unfortunately, this approach was not found to be effective for general-purpose modeling.

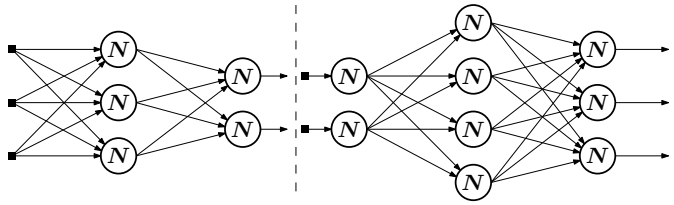


Fig. 15: Illustration of an encoder (left) with three inputs and two outputs and a decoder (right) with two inputs and three outputs.

Fig. 15 illustrates an encoder-decoder approach suggested for future work. It is somewhat similar to an autoencoder, except that the encoded value is specified to the desired airloads. The dashed line in the figure represents the separation between the encoder and decoder. Using the network on the left of this line, one can compute the airloads given the velocities and possibly actuations. In contrast, the network on the right computes the inputs given the velocities and possibly actuations. Both networks take normalized inputs, so while the encoder normalizes with respect to the absolute velocity, the decoder normalizes with respect to the absolute force. As illustrated in the figure, the encoder and decoder are allowed to be of different sizes since their underlying behavior is assumed to be of varying complexity. This approach finds a bijective parametrization to the data, even though the parametrized phenomenon is not bijective. Care must therefore be taken, and if the decoder cannot find a good fit, this might indicate that the data does not stem from a bijective model. One of the major limitations of this approach is that the data are susceptible to measurement noise. Filtering goes a long way, but the homogeneity significantly scales up the noise since it requires the input always to be normalized. So if all inputs are close to zero, any slight noise will be amplified significantly.

VII. Conclusion

This work analyzed and highlighted a little-used feature of airloads; airloads are positively homogeneous with degree of homogeneity two. A direct consequence of this property is that it enables the factorization illustrated in Fig. 2. We are then left with a simplified function with inputs sampled on a unit sphere. Two approaches for parametrizing this function have been presented, a neural network and a parametrization using spherical harmonics. Two examples have demonstrated the efficacy of our proposed strategies.

Using a linear combination of spherical harmonics basis functions, one can parameterize any smooth airload function. The model structure allows for an analog of SVD, which provides additional insight into the airload behavior and facilitates truncation.

The generality of the neural network allows the inclusion of higher dimensions and nonhomogeneous inputs. By analyzing the resulting network, it is possible to obtain further insight into the underlying behavior of the system.

Comparing the performances of the two parametrization strategies, it seems they have similar accuracy and computational complexity when parametrizing complex systems, while the neural network is superior for simpler systems.

Open questions have been stated, and future work has been proposed.

References

[1] F. Matras, D. P. Reinhardt, and M. D. Pedersen, "Parametrization of airloads using a homogeneity-based orthogonal decomposition," in *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, November 2022, pp. 553–558.

[2] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

[3] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind Energy Handbook*. John Wiley & Sons, 2011.

[4] W. Johnson, *Helicopter Theory*, ser. Dover Books on Aeronautical Engineering Series. Dover Publications, 1994.

[5] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. [Online]. Available: https://books.google.no/books?id=YqQtjhPUaNEC

[6] M. Hagan, H. Demuth, M. Beale, and O. De Jesus, *Neural Network Design 2nd Edition*. Martin Hagan, 2014.

[7] Z. Didyk and V. Apostolyuk, "Whole angle approximations of aerodynamic coefficients," in *2012 2nd International Conference "Methods and Systems of Navigation and Motion Control" (MSNMC)*, 2012, pp. 119–121.

[8] K. Gryte, R. Hann, M. Alam, J. Roháč, T. A. Johansen, and T. I. Fossen, "Aerodynamic modeling of the Skywalker X8 fixed-wing unmanned aerial vehicle," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 826–835.

[9] J. A. Grauer and E. A. Morelli, "Generic global aerodynamic model for aircraft," *Journal of Aircraft*, vol. 52, no. 1, pp. 13–20, February 2015.

[10] B. Simmons and P. Murphy, "Wind tunnel-based aerodynamic model identification for a tilt-wing, distributed electric propulsion aircraft," 2021.

[11] J. V. Caetano, C. C. de Visser, G. C. H. E. de Croon, B. Remes, C. de Wagter, J. Verboom, and M. Mulder, "Linear aerodynamic model identification of a flapping wing MAV based on flight test data," *International Journal of Micro Air Vehicles*, vol. 5, no. 4, December 2013.

[12] J. C. Gibbings, *Dimensional analysis*. Springer Science & Business Media, 2011.

[13] Encyclopedia of Mathematics. (2021) Homogeneous function. [Online]. Available: http://encyclopediaofmath.org/index.php?title=Homogeneous_function&oldid=51769

[14] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, February 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671

[15] C. Frye and C. J. Efthimiou, "Spherical harmonics in p dimensions," 2012. [Online]. Available: https://arxiv.org/abs/1205.3548

[16] "*NIST Digital Library of Mathematical Functions*," http://dlmf.nist.gov/, Release 1.1.4 of 2022-01-15, f. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. [Online]. Available: http://dlmf.nist.gov/

[17] T. Fukushima, "Recursive computation of oblate spheroidal harmonics of the second kind and their first-, second-, and third-order derivatives," *Journal of Geodesy*, vol. 87, April 2012.

[18] A. Gil and J. Segura, "A code to evaluate prolate and oblate spheroidal harmonics," *Computer Physics Communications*, vol. 108, no. 2, pp. 267–278, February 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465597001264

[19] X.-G. Wang and T. Carrington, "Using Lebedev grids, sine spherical harmonics, and monomer contracted basis functions to calculate bending energy levels of HF trimer," *Journal of Theoretical and Computational Chemistry*, vol. 2, pp. 599–608, 12 2003.

[20] V. I. Lebedev and D. N. Laikov, "A quadrature formulat for the sphere of the 131st algebraic order of accuracy," *Doklady Mathematics*, vol. 59, no. 3, pp. 477–481, 1999.

[21] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5MW reference wind turbine for offshore system development," National Renewable Energy Laboratory (NREL), Tech. Rep., January 2009.

[22] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah, "Fashionable modelling with Flux," *CoRR*, vol. abs/1811.01457, 2018. [Online]. Available: https://arxiv.org/abs/1811.01457

[23] M. Innes, "Flux: Elegant machine learning with Julia," *Journal of Open Source Software*, May 2018.

[24] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan, "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," December 2020.

[25] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[26] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, April 2006.

[27] M. L. Buhl, Jr, "A new empirical relationship between thrust coefficient and induction factor for the turbulent windmill state," National Renewable Energy Laboratory, Tech. Rep., 8 2005.

[28] F. Golnary, H. Moradi, and K. Tse, "Nonlinear pitch angle control of an onshore wind turbine by considering the aerodynamic nonlinearities and deriving an aeroelastic model," *Energy Systems*, vol. 14, pp. 197 – 227, August 2021.

[29] D. Reinhardt, M. D. Pedersen, K. Gryte, and T. A. Johansen, "A symmetry calibration procedure for sensor-to-airframe misalignments in wind tunnel data," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*, 2022, pp. 1360–1365.