Peaking into the Black-box: Prediction Intervals Give Insight into Data-driven Quadrotor Model Reliability

van Beers, J.J.; de Visser, C.C.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Peaking into the Black-box: Prediction Intervals Give Insight into Data-driven Quadrotor Model Reliability

J.J. van Beers [*] and C.C. de Visser [†]

Delft University of Technology, 2629 HS Delft, The Netherlands

Ensuring the reliability and validity of data-driven quadrotor model predictions is essential for their accepted and practical use. This is especially true for grey- and black-box models wherein the mapping of inputs to predictions is not transparent and subsequent reliability notoriously difficult to ascertain. Nonetheless, such techniques are frequently and successfully used to identify quadrotor models. Prediction intervals (PIs) may be employed to provide insight into the consistency and accuracy of model predictions. This paper estimates such PIs for polynomial and Artificial Neural Network (ANN) quadrotor aerodynamic models. Two existing ANN PI estimation techniques - the bootstrap method and the quality driven method – are validated numerically for quadrotor aerodynamic models using an existing high-fidelity quadrotor simulation. Quadrotor aerodynamic models are then identified on real quadrotor flight data to demonstrate their utility and explore their sensitivity to model interpolation and extrapolation. It is found that the ANN-based PIs widen considerably when extrapolating and remain constant, or shrink, when interpolating. While this behaviour also occurs for the polynomial PIs, it is of lower magnitude. The estimated PIs establish probabilistic bounds within which the quadrotor model outputs will likely lie, subject to modelling and measurement uncertainties that are reflected through the PI widths.

I. Introduction

DATA-DRIVEN system identification techniques are often employed to develop models of highly non-linear systems, such as the quadrotor. These data-driven approaches to quadrotor aerodynamic model identification are convenient and attractive due to limited knowledge on analytical descriptions of the quadrotor [1, 2]. Subsequently, many quadrotor models have been successfully identified in literature using a plethora of grey and black-box system identification techniques [1, 3, 4].

Of particular note are the use of polynomial step-wise regression and Artificial Neural Networks (ANNs). Sun et al. [1] construct polynomial models of the quadrotor for the high-speed ($14\ ms^{-1}$) flight regime through step-wise regression. By exploiting knowledge on influential states of the system, informed choices on the candidate regressors are made. Subsequently, the stepwise algorithm sequentially selects the best model structure, hence the term 'grey'-box. Even though the selected regressors are visible in the final model, their selection is inherently phenomenological. Hence, these models occasionally produce unexpected predictions, particularly when noise is present in the system. ANNs are an increasingly popular modelling choice by virtue of their generalizability and aptitude for capturing unknown non-linearities [3]. Consequently, they are employed to approximate dynamics in unexplored and extreme regions of the flight envelope. Take, for instance, the use of ANNs to facilitate quadrotor model identification on high-speed flight data (up to $18\ ms^{-1}$) by Bauersfeld et al. [4]. However, as a black-box modelling technique, it is ambiguous as to how exactly the ANN inputs (e.g. system states) interact to produce the model outputs. Evaluating the reliability and validity of the resultant ANN models is challenging and proves to be a potential barrier to their widespread use in the aerospace industry [5]. Indeed, uncertainty in the inputs may culminate in some unexpected model outputs for similar (e.g. noisy) inputs [6]. Ensuring safety and reliability are paramount to the accepted use of autonomous quadrotors and, thus, this lack of model reliability measures for such grey and black-box models needs to be addressed.

One way to describe the reliability of a model's prediction is through its confidence in that prediction. To this end, the associated prediction intervals (PIs) - the interval wherein a future observation will lie with a given probability (i.e. confidence) - may be used as a proxy for reliability. PIs account for both variations due to model uncertainty and variations due to measurement uncertainty (e.g. noise), up to a specifiable confidence level (typically 95%). Therefore, the often-used confidence interval (CI), which describes only model uncertainty, is a component of the PI.

---

*j.j.vanbeers@tudelft.nl, PhD candidate, Faculty of Aerospace Engineering: Control and Simulation
†Associate Professor, Faculty of Aerospace Engineering: Control and Simulation

Fortunately, for polynomial models, there are valid analytical formulations for obtaining these PIs. Although, these have yet to be applied to the aforementioned quadrotor grey-box models. In contrast, due to the inherent obscurity behind black-box models, analytical formulations that maintain the validity of any estimated PIs are challenging, if not impossible, to derive. Consequently, some ANN literature [6–9] propose various methods to estimate these PIs. One of the more straightforward and convenient approaches is known as the bootstrap method wherein an additional ANN is employed to estimate the PIs [6]. This approach is particularly attractive as it may be applied to an already trained ensemble of ANNs without needing to retrain the underlying ensemble. In their review of popular ANN PI estimation techniques, Khosravi et al. [6] found that the bootstrap approach did well to reflect uncertainty in the inputs. However, the bootstrap method often produces excessively wide PIs, which is undesirable. Instead, it is perhaps more lucrative to augment the training procedure in some way to ensure high-quality (i.e. valid, yet narrow) PIs. Accordingly, Pearce et al. [8] propose a quality-driven direct estimation of the PIs which incorporates common PI quality metrics in the cost function. This approach builds upon the lower upper bound estimation (LUBE) [7] method, but is argued to produce higher quality PIs while maintaining compatibility with common ANN libraries [8]. These ANN PI estimation methods, or indeed any others, provide a clear utility in establishing model reliability but have yet to be applied to ANN quadrotor aerodynamic models.

The contributions of this paper are two-fold; first, the aforementioned PI estimation methods are numerically validated for quadrotor aerodynamic models for the first time through an existing high-fidelity quadrotor simulation developed by Sun et al. [1, 10]. Subsequently, to further demonstrate their utility, models and validated PIs are applied to real high-speed (up to 14 ms^{-1}) quadrotor flight data. To the best of our knowledge, this is the first time such PIs are identified for quadrotor models. How the PIs respond when tasked with model interpolation and extrapolation is also explored. Through these analyses, insights into the suitability and limitations of the all PI estimation methods are uncovered.

The remainder of this paper is structured as follows: section II provides some context on the quadrotor and defines important states for model identification. Further background on prediction intervals and the considered estimation techniques are presented in section III. Thereafter, the numerical validation procedure and results are given in section IV. Finally, section V applies the PI estimation techniques to real quadrotor flight data and evaluates the sensitivity of the estimated PIs to model interpolation and extrapolation.

II. Derivation of Quadrotor Model Inputs

As this paper is concerned with the validity of PIs for quadrotor models, it is important to first provide context on the system itself. This then motivates the choices for the necessary inputs variables of the aerodynamic models.

Following Newton's laws of motion, eq. (1) and eq. (2) can be derived to describe the kinematics of the quadrotor in the body reference frame, $\{B\}$ (see fig. 1). Here, m denotes the mass of the quadrotor and I_v its moment of inertia. $\mathbf{V}_B = [u, v, w]^T$ represents the body velocity and $\mathbf{\Omega}_B = [p, q, r]^T$ the body rotational rates. $\mathbf{F}_B$ and $\mathbf{M}_B$ describe the (aerodynamic) forces and moments acting on the quadrotor body. The rotational matrix from the North-East-Down
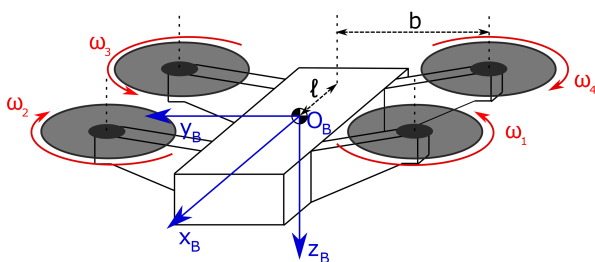

Fig. 1   Quadrotor body reference frame.

Fig. 2   Quadrotor test platform, the *HDBeetle*.

(NED) earth reference frame, $\{E\}$, to $\{B\}$ is denoted by R_{BE}. $\mathbf{g}$ is the gravitational vector in $\{E\}$.

$$m \left(\dot{\mathbf{V}}_B + \mathbf{\Omega}_B \times \mathbf{V}_B \right) = R_{BE} m \mathbf{g} + \mathbf{F}_B \tag{1}$$

$$\mathbf{I_v} \dot{\mathbf{\Omega}}_B + \mathbf{\Omega}_B \times \mathbf{I_v} \mathbf{\Omega}_B = \mathbf{M}_B \tag{2}$$

Subsequently, aerodynamic models of the quadrotor should incorporate the body velocity, $\mathbf{V}_B$, and the body rotational rates, $\mathbf{\Omega}_B$.

$$T = \kappa_0 \sum \omega_i^2 \tag{3}$$

Analytical models of the quadrotor provide further insight into useful input variables. For example, eq. (3) describes a simple model of the total thrust generated by the rotors [2, 11]. ω_i denotes the angular velocity of the ith rotor[†] and κ_0 is a constant for a given quadrotor which may be identified from hovering data. Note that, per the definition of $\{B\}$ in fig. 1, the thrust vector points in the negative z_B direction.

$$\omega_{avg} = \sqrt{\frac{\sum_{i=1}^{4} \omega_i^2}{4}} \tag{4}$$

Consequently, the rotor speeds, ω, are informative for aerodynamic quadrotor models. However, to avoid over-fitting, their aggregated influence (through eq. (4) [1]) is used instead. Likewise, the differential thrust between the rotors can account for differences between the rotors. Following the quadrotor layout in fig. 1, the so-called control rolling, pitching, and yawing moments are defined in eq. (5), eq. (6), and eq. (7) respectively. Here, $S_R \in \{-1, 1\}$ describes the rotation direction of the first rotor (such that clockwise = 1).

$$U_p = (\omega_1 + \omega_4) - (\omega_2 + \omega_3) \tag{5}$$

$$U_q = (\omega_1 + \omega_2) - (\omega_3 + \omega_4) \tag{6}$$

$$U_r = S_R \left[(\omega_1 + \omega_3) - (\omega_2 + \omega_4) \right] \tag{7}$$

Furthermore, literature on analytical quadrotor models [2, 11] also emphasize the importance of the angle of attack of the rotor plane, the influence of the induced velocity, and the effects of blade flapping on the aerodynamics of the quadrotor. While the angle of attack itself is difficult to obtain accurately for the quadrotor, the attitude angles (i.e. $[\phi, \theta, \psi]^T$) may instead be used as a proxy of this, assuming that $|\theta| < \frac{\pi}{2}$ and $|\phi| < \frac{\pi}{2}$. The quadrotor attitudes are represented by their trigonometric identities (i.e. $\sin(\cdot)$ and $\cos(\cdot)$) to accommodate their periodic nature. This also bounds, and effectively normalizes, the attitude. The induced velocity describes the added velocity of the airflow as it passes through the rotors and may be computed through eq. (8) where v_h denotes the induced velocity during hover [2].

$$v_{in} = \frac{v_h^2}{\sqrt{(V \cos \alpha_r)^2 + (v_{in} - V \sin \alpha_r)^2}} \tag{8}$$

The effects of blade flapping are related to the advance ratios [2] - a variable which contextualizes the body velocities with respect to the rotor tip velocities. Equation (9) describes how these advance ratios may be computed.

$$\mu_x = \frac{u}{\omega_{avg} R} \quad , \quad \mu_y = \frac{v}{\omega_{avg} R} \quad , \quad \mu_z = \frac{w}{\omega_{avg} R} \quad , \quad \mu_{v_{in}} = \frac{w}{\omega_{avg} R} \tag{9}$$

Therefore, the basis of input states for the quadrotor aerodynamic models identified in this paper are summarized in eq. (10).

$$X = \left[u, v, w, v_{in}, p, q, r, \omega_{avg}, \sin \phi, \cos \phi, \sin \theta, \cos \theta, U_p, U_q, U_r, \mu_x, \mu_y, \mu_z, \mu_{v_{in}} \right] \tag{10}$$

[†]It is assumed that the rotors are aligned with the x_B-y_B plane.

III. Prediction Interval Estimation Methods

Although regression aims to approximate a dependent variable given a set of predictors, measurements of these are often contaminated with bias and noise. Further uncertainties and errors arise from modelling misspecification. Both sources of uncertainty induce a variation in model predictions, which may be modelled through eq. (11) [6]. Here, $\sigma_{\hat{\epsilon}}^2$ encompasses the uncertainty associated with the measurements (i.e. dependent variable) and $\sigma_{\hat{y}}^2$ denotes the variance due to modelling misspecification and uncertainty. The total variance, σ^2, may then be used to construct the prediction intervals (PIs) for a specifiable confidence level, $1 - \alpha$. Typically, $\alpha = 0.05$ is used, leading to 95% intervals. Note that, since the PIs consider both $\sigma_{\hat{\epsilon}}^2$ and $\sigma_{\hat{y}}^2$, they encompass the associated confidence intervals (CIs).

$$\sigma^2 = \sigma_{\hat{y}}^2 + \sigma_{\hat{\epsilon}}^2 \tag{11}$$

The corresponding PI bounds, for a given measurement value, y_i, and accompanying total variance, σ_i^2, can be obtained through eq. (12). In eq. (12), $t_{N-2}^{1-\alpha/2}$ gives two sided test statistic for the chosen confidence level, $1 - \alpha$, and n gives the degrees of freedom (i.e. number of observations of the sample, i).

$$L_i = \hat{y}_i - t_{N-2}^{1-\alpha/2} \frac{\sigma_i}{\sqrt{n}}, \qquad U_i = \hat{y}_i + t_{N-2}^{1-\alpha/2} \frac{\sigma_i}{\sqrt{n}} \tag{12}$$

The PI coverage probability (PICP), defined in eq. (13), accesses the validity of a PI by verifying if the desired confidence level is reached [6, 8]. N denotes the number of test data points and c_i describes if the measurement is contained within the bounds of the PI (where L_i = lower bound, U_i = upper bound). Thus, PIs are considered valid if $PICP \geq 1 - \alpha$. Note that the PICP metric alone imposes no requirement on the width of the PIs, which may be impractically wide.

$$PICP = \frac{1}{N} \sum_{i=1}^{N} c_i \quad , \quad \text{where:} \qquad c_i = \begin{cases} 1, & y_i \in [L_i, U_i] \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Therefore, complimentary to the PICP, the mean PI width (MPIW) is often used to parameterize the width of the PIs and is typically normalized with respect to the range of the measurement data. The MPIW is given by eq. (14). Both the PICP and the MPIW should be used in tandem to describe the overall PI quality, seeing as the deficiencies of one metric are perfectly accommodated by the other [7]. A high quality PI is thus valid, yet narrow.

$$MIPW = \frac{1}{N} \sum_{i=1}^{N} (U_i - L_i) \tag{14}$$

A. Polynomial

Conveniently, for polynomial models, the total variance associated with a prediction may calculated through eq. (15) under the assumptions that the measurement error term, ϵ, is zero-mean, independent, and identically distributed. In eq. (15), $\mathbf{x_0}$ denotes the input data regressor matrix for which predictions should be made (i.e. to predict y_0), $\mathbf{X}$ represents the regressor covariance matrix used for training the model, $\mathbf{I}$ gives the identity matrix and σ_e^2 may be approximated through eq. (16).

$$\hat{\sigma}_0^2 = \sigma_e^2 \left(\mathbf{I} + \mathbf{x}_0 \left(\mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{x}_0^T \right) \tag{15}$$

$$\hat{\sigma}_e^2 = \frac{1}{N_t - 2} \sum_{i=1}^{N_t} \hat{e}_i^2 \tag{16}$$

In eq. (16), N_t represents the number of samples used for training and $\hat{e}_i = y_i - \hat{y}_i$ denotes the residual error for training sample, i. The computed $\hat{\sigma}_0$ may then be inputted into eq. (12) to construct the polynomial PI.

4

B. ANN: Bootstrap

The bootstrap method exploits the inherent stochasticity of the ANN training process and can be applied to an already trained ensemble of ANNs. The spread of predictions across an ensemble of ANN models, initialized with distinct weights and trained on different subsets of the training data set, gives an approximation of the model misspecification uncertainty [6, 8]. The associated variance may be constructed through eq. (17) [6], where B gives the number of ANN models, $\hat{y}_i$ denotes the ensemble averaged prediction of the measurement. For this reason, the bootstrap method can only be applied to an ensemble of ANNs. Although the appropriate number of ensembles depends on the system, increasing the number of ensembles does not always lead to improved performance [6].

$$\hat{\sigma}_{y_i}^2 = \frac{1}{B-1} \sum_{j=1}^{B} \left(\hat{y}_i^j - \hat{y}_i \right)^2 \tag{17}$$

The variance due to the measurement errors, $\sigma_{\epsilon_i}^2$, is typically unknown and cannot easily be obtained explicitly. Instead, an additional ANN may be used to (implicitly) estimate this variance. Through the residuals between the model predictions and the measurement data, a new data set, $D_r = \{x_i, r_i^2\}_{i=1}^{m}$, may be constructed linking the inputs, x_i, to the variance residuals, r_i^2, defined in eq. (18) [6]. Note, that the data used to derive D_r should be distinct to that used for training the initial ensemble of ANNs.

$$r_i^2 = \max \left(\left((y_{m,i} - \hat{y}_i)^2 - \hat{\sigma}_{y_i}^2 \right), 0 \right) \tag{18}$$

The $\sigma_{\epsilon_i}^2$-predictor ANN may then be used to approximate $\sigma_{\epsilon_i}^2$ by minimizing the cost function defined in eq. (19) [6]. Note that this cost function is derived assuming that $\sigma_{\epsilon_i}^2$ is Gaussian. Moreover, due to the approximation of $\sigma_{\epsilon_i}^2$ through the residual, r_i^2, the Bootstrap method tends to estimate wider PIs [6].

$$J = \frac{1}{2} \sum_{i=1}^{m} \left[\ln \left(\sigma_{\epsilon_i}^2 \right) + \frac{r_i^2}{\sigma_{\epsilon_i}^2} \right] \tag{19}$$

C. ANN: Quality Driven

Instead of estimating the PIs after-the-fact, another approach is to estimate them directly [7]. This is advantageous in that the PI bounds are obtained directly and no assumption needs to be made on their underlying distribution (e.g. eq. (12) is not used). Furthermore, the PI quality metrics (i.e. $PICP$ and $MPIW$) may be considered during training. Such an approach is outlined by Pearce et al. [8] wherein a 'quality-driven' (QD) loss function incorporating proxies of the PI quality metrics are used to train an ensemble of feed-forward neural networks (FNNs). The output of each network are the upper and lower bounds of the PI associated with a prediction. The prediction itself may be reconstructed from these bounds, for example, by assuming their midpoint. The QD loss function is described in eq. (20).

$$QD = MPIW_C + N\lambda \max \left(0, (1 - \alpha) - PICP_s \right)^2 \tag{20}$$

In eq. (20), $MPIW_C \subseteq MPIW$ describes the $MPIW$ only of samples *which encompass their measurement value* (i.e. MPIW for samples where the measurement lies outside the PI bounds are ignored). N is the batch size used for training and λ is a tunable parameter that scales the importance of the $PICP$ condition. $PICP_s$ represents the 'soft' $PICP$, defined in eq. (21), where $S(\cdot)$ denotes the sigmoid function and s is some softening factor. 'softening' of the $PICP$ is encouraged to mitigate the non-linearity of the second term in eq. (20) for compatibility with the optimizers of existing neural network library implementations (e.g. `tensorflow`). Pearce et al. [8] suggest a value of $s = 160$ for consistent performance in regression tasks.

$$PICP_s = \frac{1}{N} \sum_{i=1}^{N} S(s(y_i - L_i)) \cdot S(s(U_i - y_i)) \tag{21}$$

5

# IV. Numerical Validation of Prediction Intervals

In order to determine if the ANN PI estimation techniques do indeed produce valid PIs for the quadrotor, a numerical validation scheme is constructed. Equivalent polynomial models with associated PIs are also evaluated for reference.

A. Simulation Model Identification

For all modelling techniques, individual aerodynamic models are identified for the forces and moments of each of the axes (i.e. 6 models total, per identification technique). Moreover, a prediction interval criterion of 95% is specified for each PI estimation technique (i.e. each of the estimated PIs should contain at least 95% of the variation due to uncertainties in the data). The model identification scripts are programmed in Python (version 3.8.13).

Polynomial aerodynamic models are identified through stepwise regression [12] following the same procedure as outlined by Sun et al. [1] for their grey-box models. The basic principle of stepwise regression is to identify the best model structure given a pool of candidate regressors. This is accomplished by sequentially adding the best-fitting regressor, orthogonalized with respect to the regressors currently in the model structure, and evaluating the current model structure for redundancy after regressor additions. In consultation with the authors of [1], the rotor speeds are also incorporated in the candidate regressors of F_z for improved performance. The candidate regressors, along with the final model structures, can be found in appendix A.A.1 and appendix A.A.3 respectively.

The architectures of the ANN models for both PI estimation techniques are identical with only method-specific differences. Inspired by the ANNs of [3], all ANN models used here are composed of an ensemble of 10 dense FNNs, each with an input layer, a single hidden layer with 20 neurons, and an output layer. ReLU and a linear function constitute the hidden layer and output layer activation functions respectively. ADAM is used as the optimizer. The mean-squared-error and QD cost functions are employed for the bootstrap and quality-driven methods respectively. All ANNs are trained for 50 epochs. For the bootstrap method, an additional FNN with identical structure to the base ANNs is used, with the exception of 25 hidden neurons and a modified cost function given by eq. (19). `tensorflow` (version 2.2.0) is used to construct and train the ANNs. The input states for each ANN model can be found in appendix A.A.2.
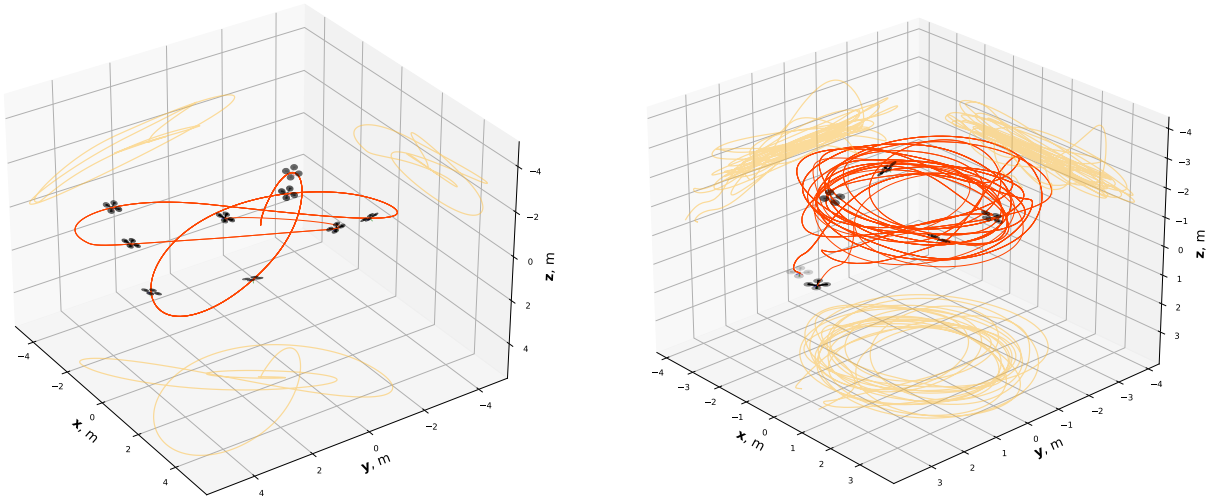
(a) Sinusoidal trajectory flown in the simulation of [10] to gather mock-flight data.

(b) Circular flight trajectory flown in the CyberZoo to achieve velocities of up to 5 ms^{-1}.

Fig. 3 Example trajectories (orange) of flights employed for model identification. Also shown are the planar trajectories (yellow) of these manoeuvres, along with some snapshots of the quadrotor's orientation at various points in the trajectory.

B. Data Generation

An existing high-fidelity quadrotor simulation, developed by Sun et al. [10] in MATLAB®(version R2022a), is used to generate mock-flight data for quadrotor model identification and PI numerical validation under consistent conditions. The simulation model itself is a rotor-local polynomial model (i.e. aerodynamic forces and moments are modelled per rotor), whereas the models identified in this paper consider only their cumulative effect on the quadrotor frame (i.e point-mass forces and moments). This discrepancy introduces some model uncertainty, which should be accountable by the estimated PIs.

In the simulation, users can specify positional (x_E, y_E, z_E) and yaw angle (ψ) commands. These are used to generate a collection of simulated flight states and aerodynamic forces & moments for model identification (training set) and numerical validation (validation set). All reference signals are either step-like pulses or sinusoidal in nature, with varying amplitude and frequency. The training set consists of both decoupled and coupled force and moment manoeuvres (e.g. decoupled: sinusoid along x-axis only; coupled: sinusoid along multiple axes). An example coupled trajectory used for training is illustrated in fig. 3a. The validation set only consists of coupled manoeuvres, distinct from those seen during training (i.e. different couplings, amplitudes, and sinusoid frequencies).

As the simulation relies on a non-linear controller [10], the flight data is initially generated noise-free to avoid the non-Gaussian transformation of input uncertainties. Noise, sampled from a Gaussian distribution, is instead injected into the relevant model inputs (i.e. eq. (10)) and measurements before identification and numerical validation to simulate uncertainty. The employed noise statistics and illustrative examples of these noisy realizations are depicted in appendix A.B. The noise variances are chosen to be similar in magnitude to, if not worse than, widely used Inertial Measurement Unit (IMU) sensors such as the MPU6000 [13].

For numerical validation, the identified models are tasked with making predictions on 1000, independent noise contaminated, realizations of the validation set of flight trajectories. The subsequent spread in model predictions and measurements is observed, and the ability of the PIs to contain these evaluated for each run.

C. Numerical Validation Results

Table 1a and table 1b summarize the results of the numerical validation of the estimated PIs subject to Gaussain noise contamination. By definition, PIs can only be considered valid when both the variation in model predictions due to uncertain inputs and uncertain measurements satisfy the PICP condition (here: $\geq 95.00\%$). This is distinguished in the tables through the *Model Variation* and *Measurement Variation* columns respectively. For reference, the PICP and (normalized) MPIW with respect to the total variation in the training data are also shown through the *Training* column.

Indeed, table 1a and table 1b show that all estimation techniques comfortably encompass at least 95% of the variations due to both uncertain inputs and uncertain measurements. Therefore, all considered estimation methods produce valid PIs. While this is perhaps unsurprising for the polynomial models, this is a significant result for the ANN PI estimation techniques as it demonstrates their capacity to estimate valid PIs for quadrotor aerodynamic models

Table 1   Results of the numerical validation of prediction intervals (PIs), defined for a confidence level of 95%**, subject to Gaussian noise injected into the model inputs and measurements. The containment of the variation in model predictions due to uncertain inputs is shown in the *Model Variation* columns. The *Measurement Variation* columns summarize the containment of measurement uncertainty. For reference, the containment of the models with respect to the combined variation for the *Training* data is also shown.**

(a) Force models

		PICP, %			MPIW, %		
		Training	Model Variation	Measurement Variation	*Training*	Model Variation	Measurement Variation
Fx	ANN (*Bootstrap*)	96.36	100.00	97.24	3.26	6.97	6.09
	ANN (*Quality driven*)	97.18	100.00	97.84	3.46	7.00	6.28
	Polynomial	97.59	100.00	96.89	3.95	6.30	5.63
Fy	ANN (*Bootstrap*)	95.78	100.00	96.76	3.15	6.16	5.95
	ANN (*Quality driven*)	97.54	100.00	98.33	3.60	7.28	7.08
	Polynomial	96.74	100.00	96.51	3.82	5.67	5.61
Fz	ANN (*Bootstrap*)	97.52	99.99	98.90	5.37	7.62	6.12
	ANN (*Quality driven*)	97.59	100.00	99.40	3.02	9.25	6.92
	Polynomial	96.68	100.00	99.94	8.13	27.32	21.78

(b) Moment models

|		PICP, %			MPIW, %		
		Training	Model Variation	Measurement Variation	*Training*	Model Variation	Measurement Variation
Mx	ANN (*Bootstrap*)	96.05	98.63	97.67	2.40	3.83	3.84
	ANN (*Quality driven*)	98.51	99.55	98.40	2.96	3.87	3.81
	Polynomial	96.22	98.75	98.29	3.14	5.58	5.80
My	ANN (*Bootstrap*)	96.84	98.26	97.89	2.76	5.76	6.39
	ANN (*Quality driven*)	99.13	99.65	98.84	3.75	6.11	7.20
	Polynomial	96.04	98.54	98.00	2.95	7.19	8.18
Mz	ANN (*Bootstrap*)	98.05	100.00	98.85	6.36	8.77	8.41
	ANN (*Quality driven*)	97.86	100.00	97.89	5.28	7.08	6.57
	Polynomial	96.75	100.00	96.92	4.64	6.00	5.77

7
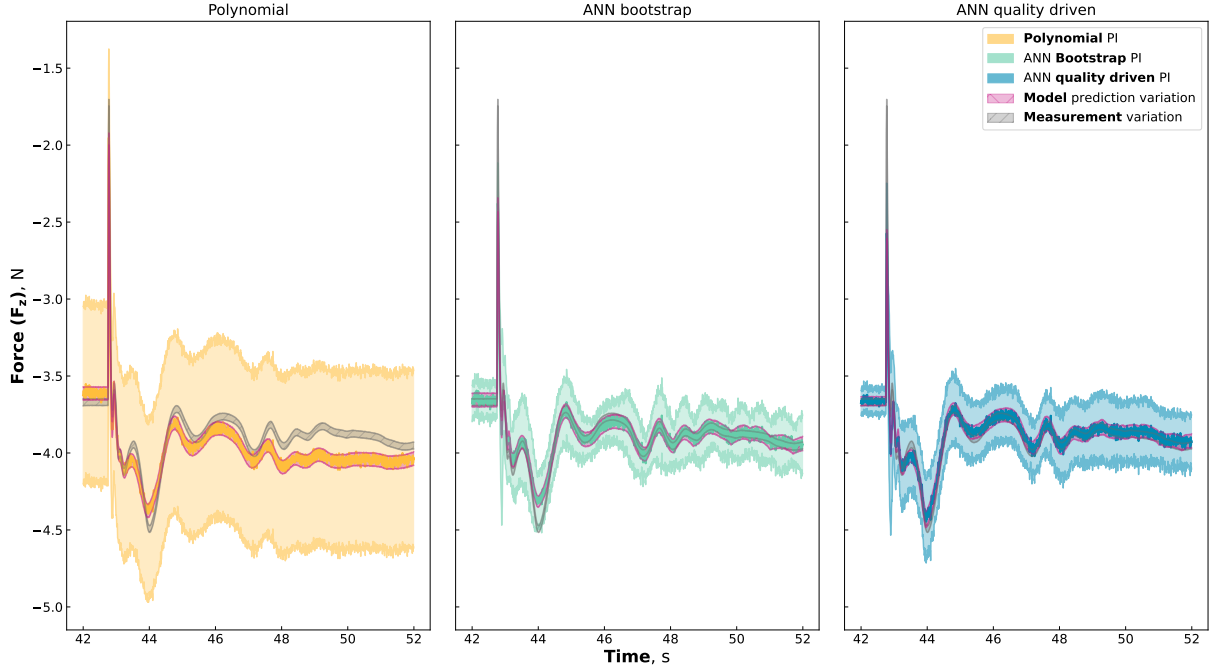
Fig. 4   Illustration of the 95% prediction interval (PI) performances of each of the PI estimation techniques - Polynomial (orange), ANN bootstrap (Cyan), ANN quality driven (Blue) - with respect to Gaussian uncertainty in the F_z measurements (grey) and model inputs, which culminate in model prediction variation (pink).

subject to Gaussian noise contamination.

The ANN quality driven method produces the widest PIs on average (compare the $MPIW$ of the methods in table 1a and table 1b). This result may stem from the fact that the ANN quality driven method requires more training epochs than the bootstrap method to converge [7, 8] and, here, they are trained for the same amount of epochs. Hence, the PI performance of the quality driven method may improve if afforded more training epochs, at the risk of over-fitting. The ANN bootstrap and polynomial methods produce PIs of similar widths, with the exception of F_z wherein the polynomial model estimates excessively wide PIs. This poor performance is a symptom of the lack of approximation power of the polynomial F_z model which, at times, struggles to fit the measurement data. This is evident in the under-approximation* of this model in fig. 4 and both the number of selected regressors and relatively poor coefficient of determination (R2) of this model in table 5. The poor F_z performance likely occurs because the simulation model of Sun et al. [10] is rotor-local (i.e. aerodynamic models for each rotor) whereas the polynomial represents their aggregated effect. To account for this discrepancy and maintain valid PIs, the error residuals in eq. (16) must grow.

As illustrative examples of when all model identification techniques enjoy sufficient approximation power, fig. 5 and fig. 6 compare the estimated PIs for F_x and M_y respectively. Interestingly, although the PI performances are comparable between all techniques (see table 1a and table 1b), their behaviour is visually different. The polynomial PIs remain relatively consistent in width across manoeuvres and trimmed flight whereas the ANN-based PIs grow and contract with the magnitude of the forces and moments, reflecting uncertainty in the model. This an important attribute of the these PIs that will be revisited as their model extrapolation sensitivity is explored with real quadrotor flight data in section V.

Overall, these numerical validation results suggest that the bootstrap method is the superior ANN PI estimation technique. Moreover, the fact that this method is compatible with already trained ensembles of ANNs only further promotes its utility. Note, however, that validity of the PIs is only demonstrated for Gaussian noise and uncertainties. Furthermore, when afforded sufficient approximation power, the polynomial estimation produces the highest quality PIs.

---

*Naturally, this under-approximation calls into question the validity of the polynomial F_z model with respect to the Ordinary Least Squares (OLS) assumption of zero-mean error residuals. However, this assumption is satisfied for this model with $-3.17 \cdot 10^{-14} \approx 0$.
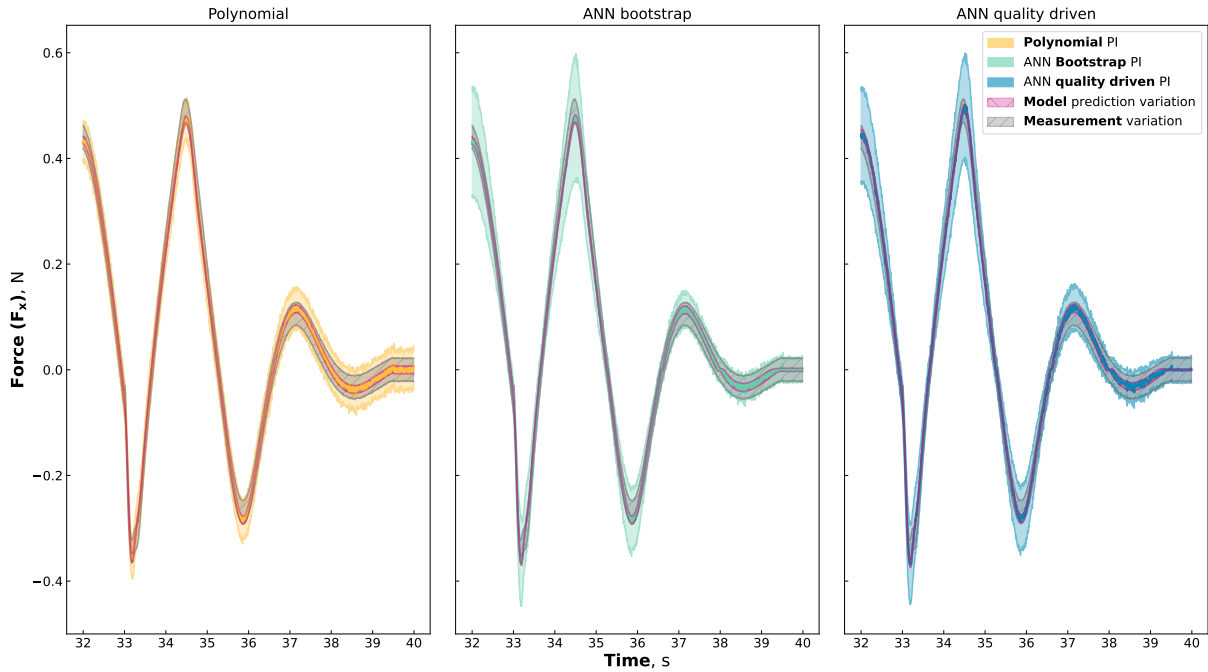
Fig. 5   Illustration of the 95% prediction interval (PI) performances of each of the PI estimation techniques - Polynomial (orange), ANN bootstrap (Cyan), ANN quality driven (Blue) - with respect to Gaussian uncertainty in the F_x measurements (grey) and model inputs, which culminate in model prediction variation (pink).
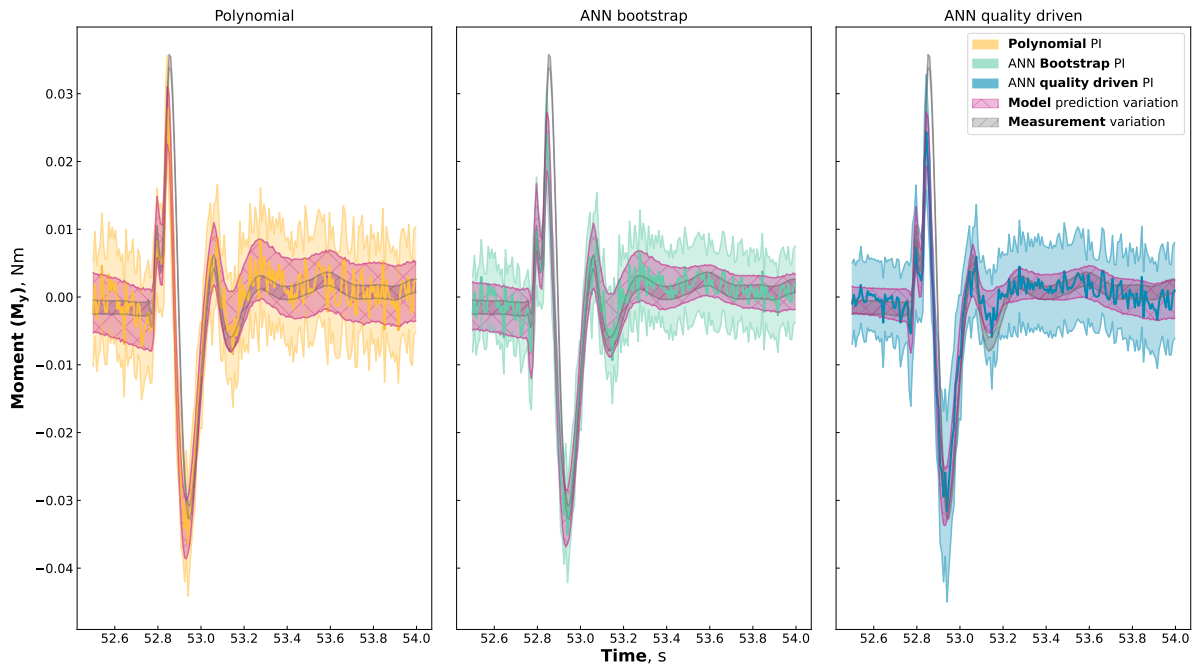

Fig. 6   Illustration of the 95% prediction interval (PI) performances of each of the PI estimation techniques - Polynomial (orange), ANN bootstrap (Cyan), ANN quality driven (Blue) - with respect to Gaussian uncertainty in the M_y measurements (grey) and model inputs, which culminate in model prediction variation (pink).

V. Application to Real Quadrotor Flight Data

While the numerical validation results show that the discussed PI estimation methods produce valid PIs, it is as of yet unclear how practical such PIs are when applied to real quadrotor flight data. Moreover, it would be beneficial if the estimated PIs widen when tasked with making predictions beyond the identification flight envelope but remain the same, or even decrease, when interpolating. To evaluate these, models and associated PIs are estimated on real quadrotor flight data obtained in a wind tunnel at different flight speeds.

A. Quadrotor platform

The *HDBeetle* - a custom 3-inch quadrotor constructed from commercial components - is used as the test platform and is depicted in fig. 2. The physical properties of the *HDBeetle* are summarized in table 2. This quadrotor is equipped with the Holybro Kakute F7 (STM34F745) flight controller. BetaFlight is elected for the flight control software since it is open-source and supports Bi-directional DSHOT. The latter of which facilitates the high-frequency and sufficiently synchronized reading of the motor (e)RPMs[†]. Such information is necessary to identify accurate models of the quadrotor. A disadvantage of BetaFlight is that it has limited autonomous support, hence all flights are piloted. Subsequently, the *HDBeetle* is outfitted with the DJI Digital FPV system to facilitate piloted first-person-view (FPV) flight.

The body accelerations and rotational rates are measured by the on-board IMU (MPU6000). Moreover, BetaFlight also estimates the quadrotor's attitude through sensor fusion of the IMU data. While these states are logged at 500 Hz by BetaFlight, they are pre-filtered (through a series of notch and low-pass filters) before logging. Fortunately, the motor rotational speeds are available, also at 500 Hz, without any pre-processing.

Table 2   Physical properties of the *HDBeetle*. Note that mass of the quadrotor is an average, and may change slightly across flights due to battery exchanges. Moreover, the moment of inertia is obtained experimentally through a trifilar pendulum with the propellers removed.

Mass, kg	Moment of Inertia, $kg \cdot m^2$			Characteristic length b, m	Rotor radius R, m
	I_{xx}	I_{yy}	I_{zz}		
0.433	8.65E-04	1.07E-03	1.71E-03	0.077	0.0381

B. Data Acquisition

The quadrotor is flown in both the CyberZoo and Open Jet Facility (fig. 7) of the Faculty of Aerospace Engineering at the Delft University of Technology. Both facilities are equipped with the OptiTrack motion capturing system, which completes the state vector by providing accurate attitude and velocity information at 120 Hz[‡].

The CyberZoo is an open flight arena of approximately 7x10x10 m in height, width, and length respectively. This confined space only allows for hovering and low-speed flights (up to 5 ms^{-1}). The aim of the low-speed flights is to fly as fast as possible by orbiting around an object stationed at the center of the CyberZoo, as depicted in fig. 3b. High-speed flights (up to 14 ms^{-1}) are flown in the Open Jet Facility (OJF) wherein both 'steady state' and 'active' flights are flown. The steady state flights involve keeping the quadrotor in approximately[§] the same position while the active flights aim to excite the longitudinal forces and moments. These therefore involve flying into and away from the incident wind to excite F_x & F_z (refer to fig. 8) and rapidly pulsing the pitch to excite M_y. As only these forces and moments can be excited safely for piloted FPV flight in the wind-tunnel, the demonstration of PI practicality for real-world models is limited to the longitudinal forces and moments. Nonetheless, the results found should extend to the lateral models by virtue of the symmetry of the quadrotor. Both the steady state and active flights were performed at wind speeds of 5, 8, 10, and 14 ms^{-1}. These wind speeds are recorded through LabView at 500 Hz.

[†]Note that the rotor RPMs are not measured directly, but rather, a proxy of this through their electronic RPMs (i.e. eRPMs). The true RPMs may be reconstructed by scaling the eRPM value by some constant, typically, the number of motor poles (= 12 for the *HDBeetle* motors).

[‡]To facilitate smooth tracking, a trade-off has to be made between the OptiTrack exposure and sampling rate settings. For our applications, a sampling rate of 120 Hz is the highest found that affords suitable tracking performance.

[§]Note that due to the piloted flights, the quadrotor is not held exactly in the same spot. Thus, the objective is to keep it within the clean airflow and as stationary as possible.

Fig. 7   The Open Jet Facility (OJF) of the Aerospace Engineering faculty at the Delft University of Technology. The OJF has a test section (i.e. nozzle) of 2.85 by 2.85 m and is capable of achieving wind-speeds of up to 35 ms^{-1}.
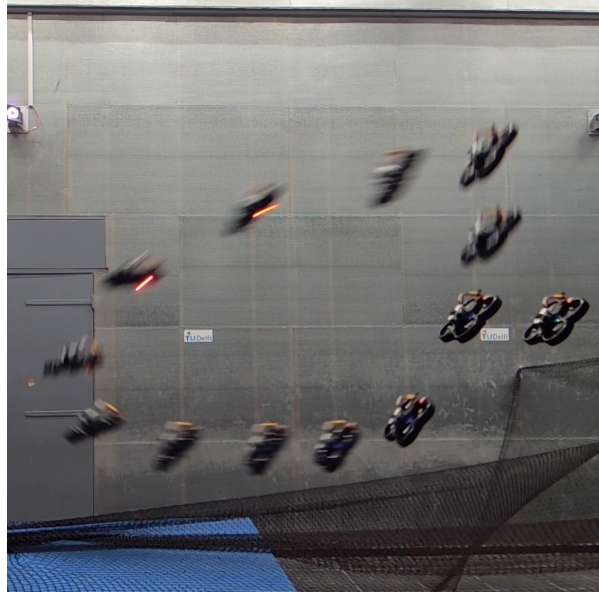

Fig. 8   Example trajectory of a forward-backward manoeuvre to excite the F_x and F_z forces during a 10 ms^{-1} flight in the Open Jet Facility. The nozzle of the wind tunnel is to the left of the picture, thus the incident wind moves from left to right.

C. Data Processing

Due to a lack of reliable timestamp information in the equipped version of BetaFlight, the onboard data is aligned with the OptiTrack data (re-sampled to 500 Hz) through a cross-correlation of the attitude angles from each data source for each flight. For the OJF data, the wind tunnel measurements are synchronized and added to the (re-sampled) OptiTrack velocities using timestamp information prior to the onboard alignment.

The synchronized data is then passed through an extended Kalman Filter (EKF), based on [14], to fuse the onboard sensor information with the (wind-combined) OptiTrack data and to estimate IMU biases. While OptiTrack is often considered ground-truth, occasional dropouts in tracking occur, for example, during high-speed flight where the quadrotor is pitched significantly and the tracking markers are thus obscured (see, for example, attitudes reached during some manoeuvres: fig. 8). This results in erroneous oscillations in the OptiTrack data, which is mitigated by the IMU data in the EKF. Subsequently, the data is further filtered using a low-pass 4th order butter-worth filter at 100 Hz [1].

D. Model Identification

Quadrotor longitudinal aerodynamic models are then identified using the same procedure as for the simulation models in section IV.A but with candidate polynomials and ANN input vectors as shown in appendix A.C.1 and appendix A.C.2 respectively. For identification, the body rotational rates and rotor speeds are taken from the onboard data while the velocities and attitudes stem from OptiTrack. Table 9 summarizes the identified polynomial model structures. Figure 9, fig. 10, and fig. 11 depict the predictive capacities, and associated PIs, of the identified models of the *HDBeetle* constrained to regions of relevant excitation manoeuvres. All models appear to fit the measurement data well, especially for low frequencies.

To investigate the behaviour of the estimated PIs when subjected to interpolation and extrapolation, the quadrotor models are identified on data limited to 10 ms^{-1} but excluding 8 ms^{-1} wind tunnel flights. These models are then tasked with making predictions on the 0, 5, and 10 ms^{-1} training (identification) data, on wind tunnel 8 ms^{-1} data (model interpolation), and on wind tunnel 14 ms^{-1} data (model extrapolation). It is expected that the PIs stay approximately the same, or shrink, when interpolating but widen (through a larger MPIW) once the models begin extrapolating.
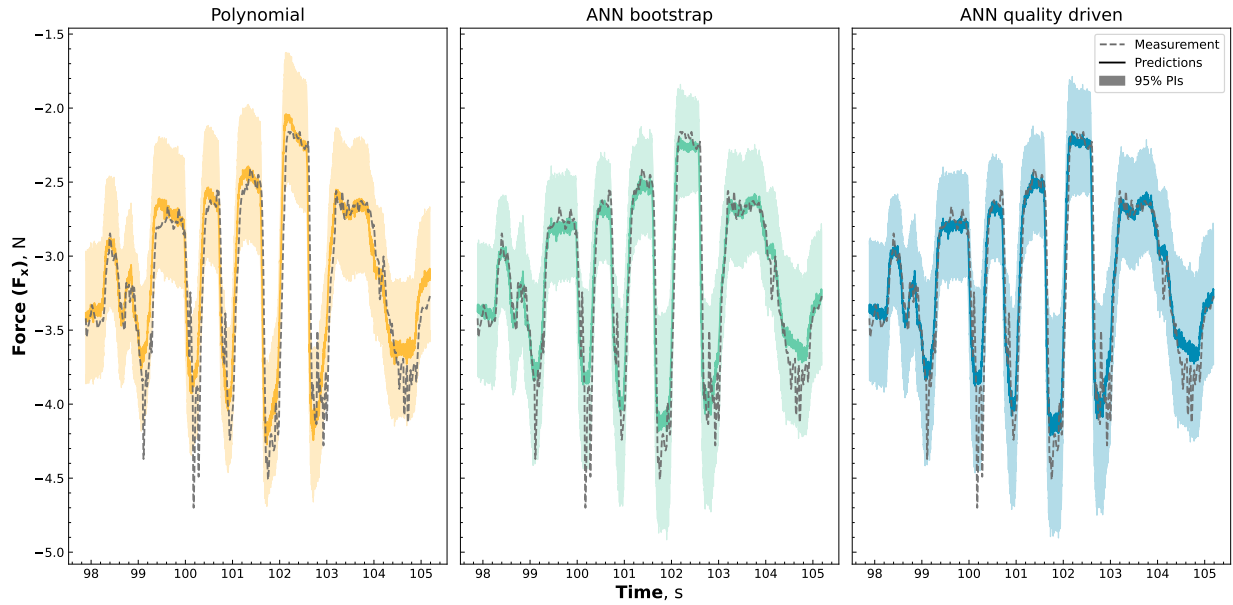
Fig. 9   Model predictive capacities, and associated 95% prediction intervals (PIs), of the identified F_x models of the *HDBeetle* during throttle pulse manoeuvres in the Open Jet Facility (wind speed = $10\ ms^{-1}$). The dotted grey line denotes the measured F_x and the solid lines describe the model-specific predictions. 95% PIs are represented by the corresponding shaded region.
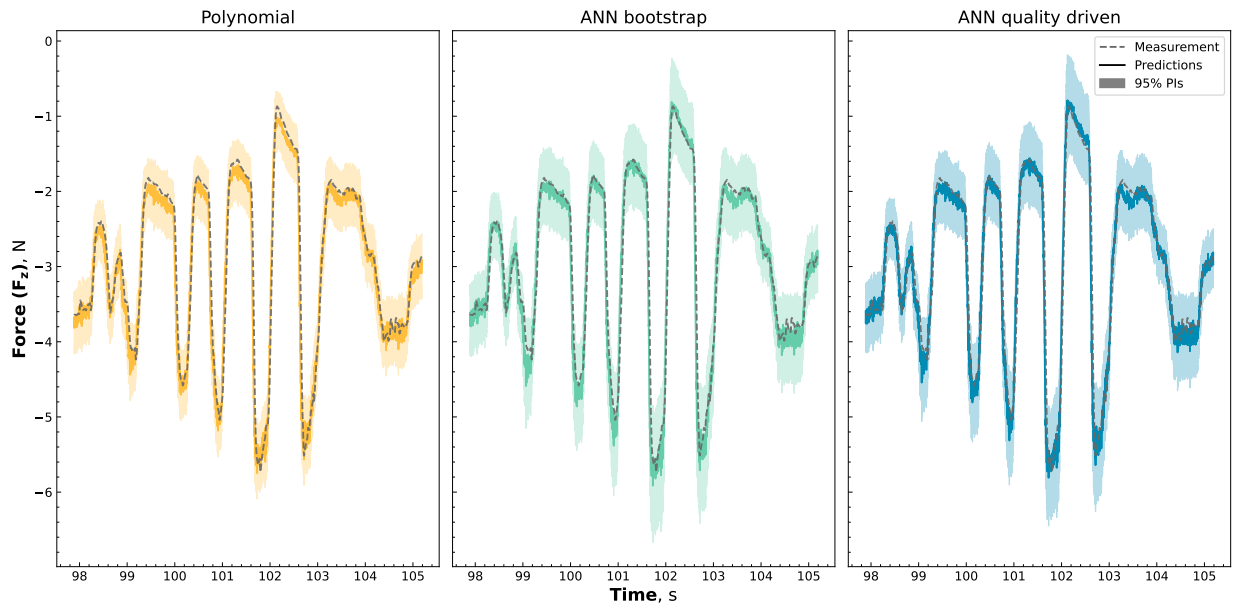

Fig. 10   Model predictive capacities, and associated 95% prediction intervals, of the identified F_z models of the *HDBeetle* during throttle pulse manoeuvres in the Open Jet Facility (wind speed = $10\ ms^{-1}$). The dotted grey line denotes the measured F_z and the solid lines describe the model-specific predictions. 95% PIs are represented by the corresponding shaded region.
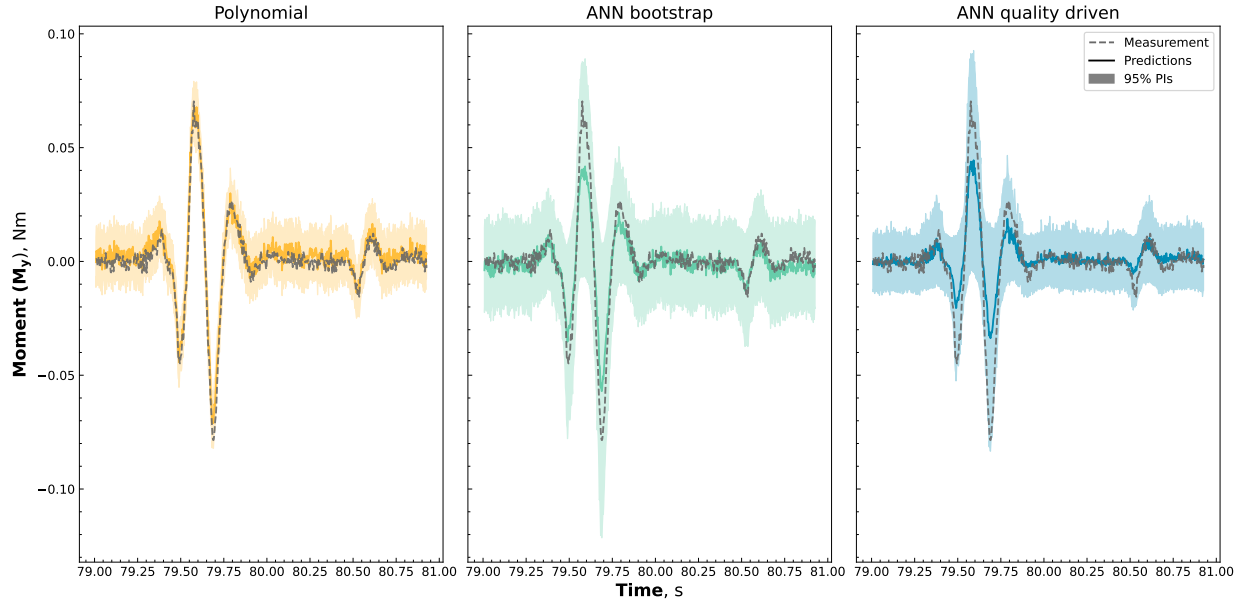
Fig. 11   Model predictive capacities, and associated 95% prediction intervals, of the identified M_y models of the *HDBeetle* during pitch angle pulse manoeuvres in the Open Jet Facility (wind speed = $10\ ms^{-1}$). The dotted grey line denotes the measured M_y and the solid lines describe the model-specific predictions. 95% PIs are represented by the corresponding shaded region.

E. Real-world PI Performance

Table 3 presents the PI performances of the longitudinal *HDBeetle* aerodynamic models subject to the identification (training) data, model interpolation ($8\ ms^{-1}$ data), and model extrapolation ($14\ ms^{-1}$ data). To facilitate a comparison between the difference flight speeds, the absolute MPIW values are shown in table 3.

Both the bootstrap and quality driven ANN PI estimation techniques experience a decrease in MPIW when interpolating and an increase when extrapolating. This suggests that the ANN PIs adequately reflect uncertainty in the model due to extrapolation. As the estimated PIs are found to be valid in section IV, they can be said to provide

Table 3   Comparison of the 95% prediction interval (PI) performances for the identified longitudinal models of the *HDBeetle* when applied to prediction on the identification (incident wind of $10\ ms^{-1}$), interpolation (incident wind of $8\ ms^{-1}$), and extrapolation (incident wind of $14\ ms^{-1}$) data sets.

|		PICP, %			MPIW, N		
		Identification	Interpolation	Extrapolation	*Identification*	Interpolation	Extrapolation
Fx	ANN (Bootstrap)	*98.26*	97.35	97.15	*0.89*	0.77	1.26
	ANN (Quality driven)	*98.25*	97.30	96.97	*0.89*	0.77	1.20
	Polynomial	*95.29*	96.06	89.19	*0.83*	0.83	0.83
Fz	ANN (Bootstrap)	*98.04*	97.79	39.21	*0.73*	0.69	1.06
	ANN (Quality driven)	*97.98*	97.78	33.48	*0.71*	0.69	1.03
	Polynomial	*95.70*	97.54	75.85	*0.69*	0.69	0.69
My	ANN (Bootstrap)	*99.95*	99.97	100.00	*0.06*	0.04	0.08
	ANN (Quality driven)	*99.79*	99.84	99.85	*0.04*	0.03	0.05
	Polynomial	*97.15*	97.31	98.92	*0.02*	0.02	0.02

13
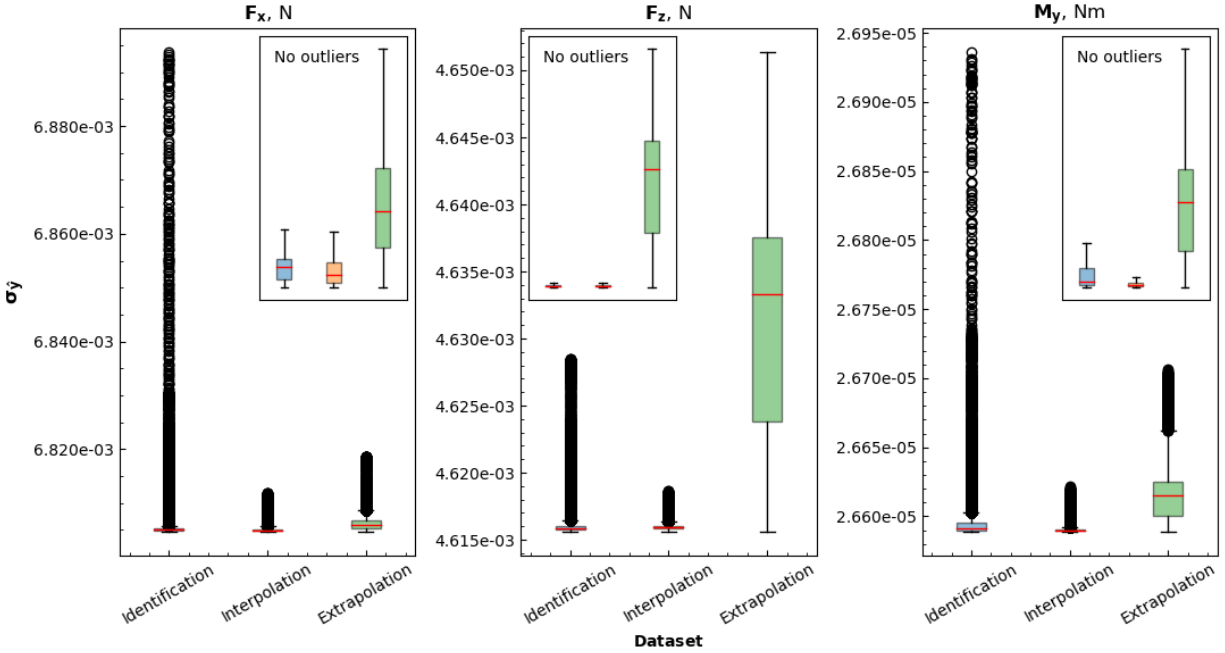
Fig. 12 Comparison of the variance due to modelling uncertainty of the polynomial estimated PIs (second term in eq. (15)) for the longitudinal models of the *HDBeetle* for the identification, interpolation, and extrapolation data sets. The inset plots depict these comparisons without the outliers.

probabilistic bounds - which grow with uncertainty - within which the model predictions will likely lie despite their black-box nature. This is significant as it establishes an expected range of predictions for black-box models, subject to system uncertainties. Such information may be exploited to avoid surprises and inform (model-predictive or robust) controllers or to identify and evaluate extreme scenarios by considering the interval limits. Likewise, the PI growth can be monitored as a proxy for model extrapolation and confidence in ANN predictions.

Contrarily, from table 3, it would appear that the polynomial PI widths remain constant regardless of interpolation or extrapolation. This is inconsistent with the expected behaviour. Revisiting the polynomial PI equation, eq. (15), reveals that the PIs can only grow from differences between the observed model regressor variances and the new regressor values subject to the new input data. In fact, the polynomial PIs do indeed grow due to this model uncertainty term, albeit very slightly, as illustrated in fig. 12. It is clear that the extrapolation data set stimulates a significant widening of the polynomial PIs while the interpolation set does not. Note that the outliers represent 0.7% of the data and may arise from artefacts in the data. For instance, in the wind tunnel, the local air speeds of the turbulent air near the edges of the nozzle are unknown. As the quadrotor is flown manually, excursions to these regions occur on occasion, especially for active flights. Likewise, un-modelled effects, such as reverse flow, which occur in the CyberZoo flights may also contribute to these outliers. While the growth of the polynomial PIs appears to be visually significant in fig. 12, the associated changes in magnitudes are minuscule in comparison to the PI width (for F_x: $\sigma_{\hat{y}} = 0.002\%$; F_z: $\sigma_{\hat{y}} = 0.048\%$; and M_y: $\sigma_{\hat{y}} = 0.070\%$ of MPIW). For the quadrotor models identified here (i.e. including the simulation models) it appears that the error residual variance, $\hat{\sigma}_e^2$ (first term in eq. (15)), is the dominant term governing the width of the polynomial PI. Indeed, eq. (15) is only valid for interpolation (i.e. within the scope of the training data) and not extrapolation as nothing can reliably be said about the new variation of the measurement data. This highlights some of the limitations of the Polynomial PI estimation technique, as there is - visually - minimal indication of extrapolation, unlike for the ANN counterparts. Nonetheless, this extrapolation is still apparent in the growth of the $\sigma_{\hat{y}}$ model uncertainty terms.

Furthermore, all PI estimation techniques show a drop in PICP when extrapolating, reflecting a degradation in model fit. This is most pronounced for models of F_z, and especially so for the ANN models. Overall, this deterioration in performance can be explained by differences between the identification, interpolation, and extrapolation datasets for key states of the longitudinal models, as summarized in Figure 13. Indeed, the greatest differences between the extrapolation and identification sets occur for states related to F_z. Physically, the rotors need to work harder to counteract higher
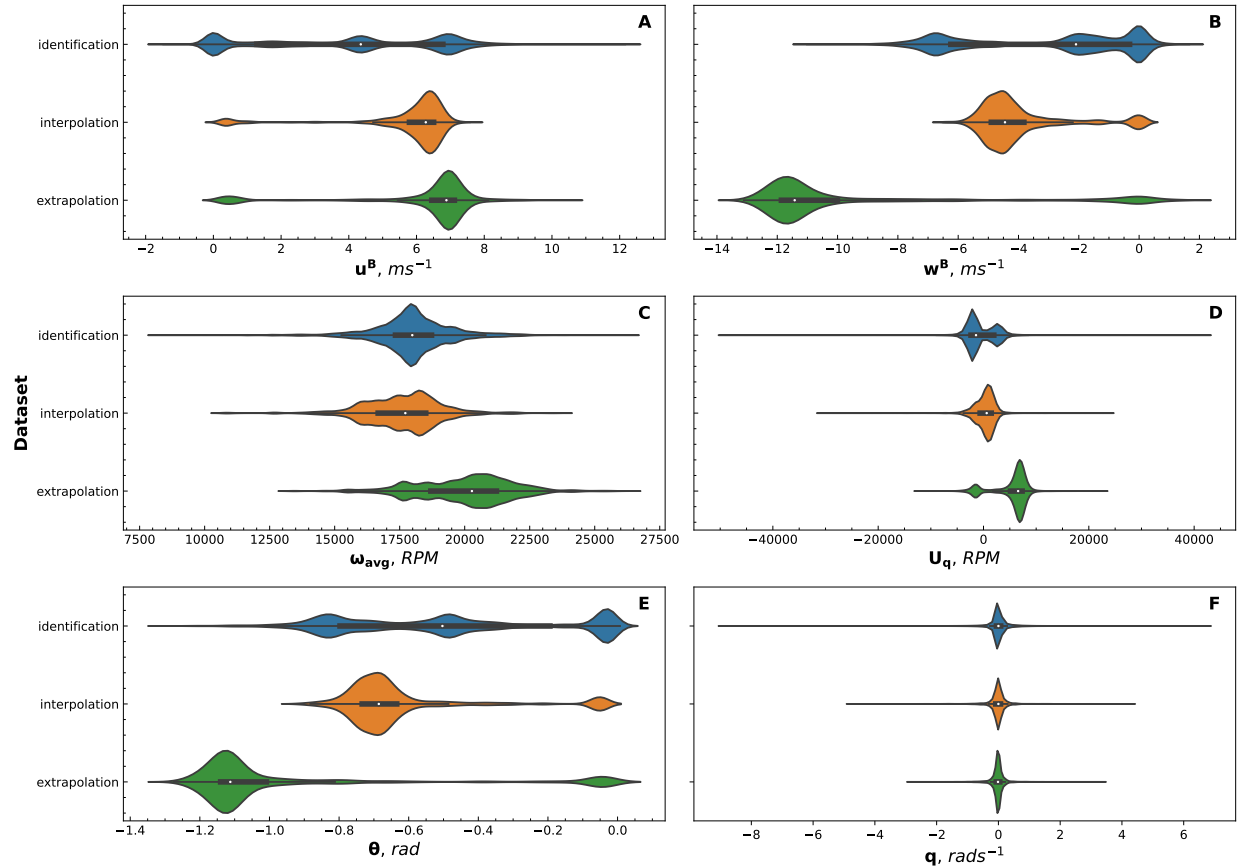
14

Fig. 13 Differences in distribution, across the identification, interpolation, and extrapolation data sets, for key states of the longitudinal *HDBeetle* models.

velocities (subplots **B** and **C** in fig. 13). This is only compounded by the increase in required trim pitch angle at higher velocities (subplot **E** in fig. 13). The comparatively similar state distributions of subplots **A**, **D**, and **F** account for the adequate PICP performance of the F_x and M_y models in table 3 and even afford valid PIs for the ANNs showcasing their well-known generalizability. Nonetheless, the fact that the models are extrapolating is evident through the MPIW.

VI. Conclusion

Through mock data obtained from a high-fidelity quadrotor simulation, all considered prediction interval (PI) estimation techniques (polynomial, ANN bootstrap, and ANN quality driven) are shown to be numerically valid for aerodynamic quadrotor models subject to Gaussian noise statistics. Of the ANN estimation techniques, the bootstrap method appears to be superior in terms of PI quality and practicality. Indeed, the bootstrap method may be directly applied to an existing ensemble of ANN quadrotor models, provided that training datasets are available. Moreover, the polynomial PI results suggest that the width of the estimated PIs are reflective of the approximation power of the underlying model whereby excessively wide PIs may indicate a lack of approximation power.

As the PI estimation techniques produce valid PIs, they are successfully applied to real quadrotor high-speed (up to 14 ms^{-1}) flight data. The sensitivity of the estimated 95% PIs to model interpolation and model extrapolation for each of the techniques is evaluated. Both ANN PI estimation techniques experience a swelling of PI widths for model extrapolation, reflecting a growing model uncertainty. These PIs remain unchanged, or shrink, when interpolating. Such behaviour is desirable in practice as insights into the model confidence and, crucially, the (95%) prediction bounds of the otherwise black-box ANN models can be ascertained and accounted for. This behaviour is much less pronounced for the polynomial PIs, to the point that they are effectively unchanged for both model interpolation and extrapolation. The roots of this lie in the PI estimation itself, where the dominant term - associated with residual error variances - remains

constant after training. Indeed, the model uncertainty term of the polynomial PI is reflective of model uncertainty but its magnitude is negligible in comparison to the measurement uncertainty term. This reveals a limitation to the analytically derived polynomial PIs when applied to model extrapolation.

While the PIs discussed in this paper illustrate their utility, subsequent work should seek to expose further limitations of these PIs and investigate to what extent these PIs remain valid. For example, how do the estimated PIs react to changes in noise statistics - be this the distribution used, or changes to the Gaussian variances and means? Likewise, the relation between model complexity on the PI widths is also an interesting avenue for future work. Although challenging to do practically, it is desirable to also validate these PI estimation techniques using real flight data through repeated flights of a quadrotor in a controlled environment.

A. Appendix

A. Simulation Model Identification

1. Polynomial Candidate Regressors

As these regressor pools can get quite large, the compact notation given in eq. (22) is used to summarize the candidate pool. Here, C_0 denotes the bias vector, $P^2(x, z)$ represents the polynomial basis, and $\{1, a\}$ the interacting factors.

$$
\begin{aligned}
\hat{Y} &= C_0 + P^2(x, z)\{1, a\} \\
&= C_0 + P^2(x, z) \cdot 1 + P^2(x, z) \cdot a \\
\text{where:} \\
P^2(x, z) &= x^2 + 2 \cdot xz + z^2
\end{aligned}
\tag{22}
$$

The chosen candidate regressors for the simulation models are based on the gray-box models of Sun et al. [1, 10]. In the following equations, terms outside the polynomials (i.e. $P^d(\cdot)$) are considered fixed regressors, chosen based on physical principles, and are thus always present in the model.

The candidate force model structure for F_x, F_y, and F_z are given by eq. (23), eq. (24), and eq. (25) respectively.

$$
\hat{F}x = X_0 \cdot \mu_x + P^3(\mu_x, |\mu_y|, \mu_z) \cdot \{1\}
\tag{23}
$$

$$
\hat{F}y = Y_0 \cdot \mu_y + P^3(|\mu_x|, \mu_y, \mu_z) \cdot \{1\}
\tag{24}
$$

$$
\begin{aligned}
\hat{F}z = {} & Z_0 \cdot (\mu_x^2 + \mu_y^2) - Z_1 \cdot (\mu_z - \mu_{v_{in}}) + P^3(|\mu_x|, |\mu_y|, \mu_z) \cdot \{1\} + P^3(|U_p|, |U_q|, U_r) \cdot \{1, \omega_{avg}\} \\
& + P^3(|p|, |q|, r) \cdot \{1, \omega_{avg}\} + P^2(\omega_{avg}) \cdot \{1\}
\end{aligned}
\tag{25}
$$

The candidate moment model structure for M_x, M_y, and M_z are given by eq. (26), eq. (27), and eq. (28) respectively.

$$
\hat{M}x = L_0 \cdot U_p + P^3(\mu_y, \mu_z) \cdot \{1, p, U_p\} + P^3(p, U_p) \cdot \{1, \omega_{avg}\}
\tag{26}
$$

$$
\hat{M}y = M_0 \cdot U_q + P^3(\mu_x, \mu_z) \cdot \{1, q, U_q\} + P^3(q, U_q) \cdot \{1, \omega_{avg}\}
\tag{27}
$$

$$
\begin{aligned}
\hat{M}z = {} & N_0 \cdot U_r + P^3(\mu_x, \mu_y, \mu_z) \cdot \{1, |p|, |q|, r, |U_p|, |U_q|, U_r, \omega_{avg}\} + P^3(|p|, |q|, r) \cdot \{1, \omega_{avg}\} \\
& + P^3(|U_p|, |U_q|, U_r) \cdot \{1, \omega_{avg}\}
\end{aligned}
\tag{28}
$$

2. ANN Input Vectors

Table 4   Input vectors employed for the simulation model identification for both the ANN bootstrap and ANN quality driven techniques.

Model	Input variables
Fx	$\mu_x, |\mu_y|, \mu_z$
Fy	$|\mu_x|, \mu_y, \mu_z$
Fz	$w, |\mu_x|, |\mu_y|, \mu_z, \omega_{avg}, \mu_{v_{in}}, |U_p|, |U_q|, U_r, |p|, |q|, r$
Mx	$\mu_y, \mu_z, U_p, p, \omega_{avg}$
My	$\mu_x, \mu_z, U_q, q, \omega_{avg}$
Mz	$\mu_x, \mu_y, \mu_z, p, q, r, U_p, U_q, U_r, \omega_{avg}$

3. Identified Polynomial Model Structures

Table 5   Identified polynomial force model structures, coefficients, and goodness-of-fit metrics for the simulation model. The rows indicate the order of addition to the model, as selected by the stepwise regression algorithm. Rows in grey denote fixed regressors.

	Fx			Fy			Fz	
Regressor	Coefficient	R2	Regressor	Coefficient	R2	Regressor	Coefficient	R2
bias	-1.431e-04	-	*bias*	-2.187e-05	-	*bias*	-1.654e+00	-
μ_x	-2.498e-01	0.98	μ_y	-2.675e-01	0.97	$\mu_x^2 + \mu_y^2$	3.209e-02	-
$\mu_x^{1.0} \cdot \mu_z^{1.0}$	1.780e-02	0.98	$\mu_y^{1.0} \cdot \mu_z^{1.0}$	2.779e-02	0.98	$-\mu_z + \mu_{v_{in}}$	-1.070e-01	0.01
–	-	-	$|\mu_x|^{1.0} \cdot \mu_y^{1.0}$	-1.844e-02	0.98	$\omega_{avg}^{2.0}$	-1.334e-02	0.85
–	-	-	–	-	-	$\omega_{avg} \cdot U_r^{2.0}$	-1.603e-08	0.87
–	-	-	–	-	-	$\omega_{avg}^{1.0}$	2.746e-01	0.88
–	-	-	–	-	-	$\mu_z^{1.0}$	-5.148e-01	0.89
–	-	-	–	-	-	$\omega_{avg} \cdot |U_q|^{3.0}$	-1.218e-11	0.89

Table 6   Identified polynomial moment model structures, coefficients, and goodness-of-fit metrics for the simulation model. The rows indicate the order of addition to the model, as selected by the stepwise regression algorithm. Rows in grey denote fixed regressors.

	Mx			My			Mz	
Regressor	Coefficient	R2	Regressor	Coefficient	R2	Regressor	Coefficient	R2
bias	-3.838e-05	-	*bias*	5.530e-05	-	*bias*	1.705e-05	-
U_p	2.938e-04	0.89	U_q	2.214e-04	0.85	U_r	-3.321e-06	0.94
$\mu_y^{1.0}$	-2.649e-02	0.99	$\mu_x^{1.0}$	2.950e-02	0.98	$r^{1.0}$	-2.295e-03	0.97
$U_p \cdot \mu_y^{2.0}$	-5.857e-06	0.99	$\mu_x^{1.0} \cdot \mu_z^{1.0}$	1.036e-02	0.99	$|U_q| \cdot \mu_x^{1.0} \cdot \mu_y^{1.0} \cdot \mu_z^{1.0}$	1.981e-06	0.98
–	-	-	$U_q \cdot \mu_z^{2.0}$	-1.298e-05	0.99	$\omega_{avg} \cdot U_r^{1.0}$	1.203e-06	0.98

17

B. Gaussian Noise Statistics

Table 7   Gaussian noise statistics used to contaminate the simulation model inputs and measurements with uncertainty. The noise standard deviations are selected to be of the same magnitude as, but slightly worse than, commonly used sensors such as the MPU6000 [13].

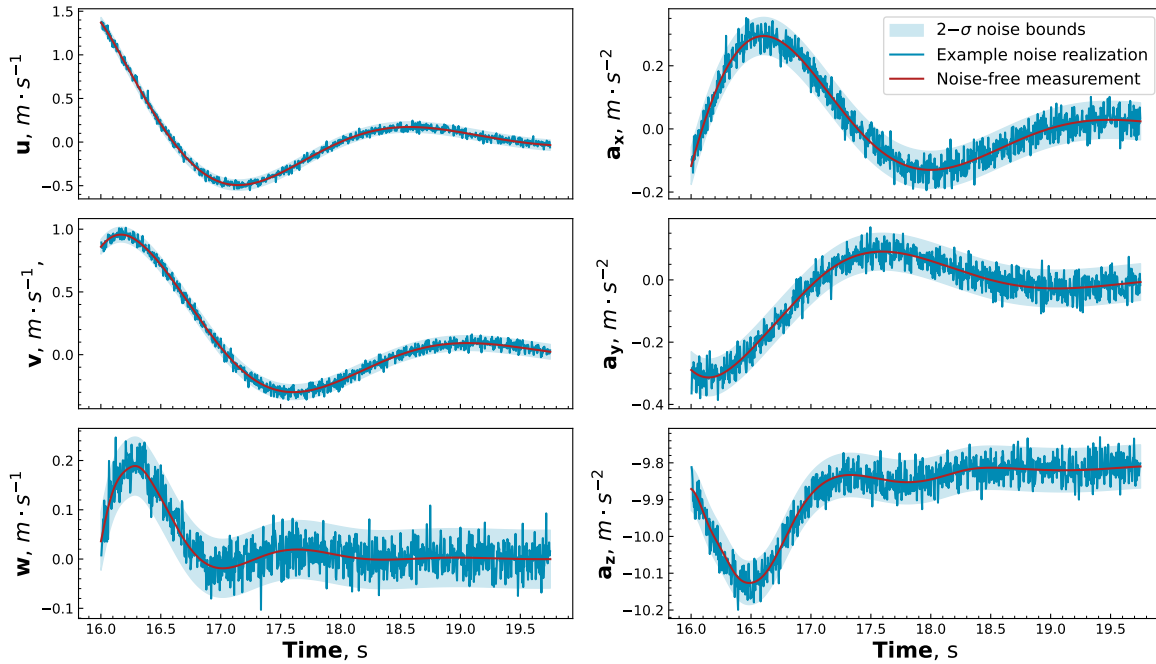|	Rotor speeds [eRPM]	Velocity [ms^{-1}]			Acceleration [ms^{-2}]			Attitude [rad]			Body rates [rads^{-1}]		
	Ω	u	v	w	a_x	a_y	a_z	ϕ	θ	ψ	p	q	r
Mean, μ	0	0	0	0	0	0	0	0	0	0	0	0	0
Standard deviation, σ	5	0.03	0.03	0.03	0.03	0.03	0.03	0.01	0.01	0.01	0.01	0.01	0.01

Fig. 14   Illustration of the addition of Gaussian noise to the noise free measurements (red) of the velocities and accelerations to produce associated noisy realizations (blue). The shaded region represents the 2−σ bounds of the noise statistics used.
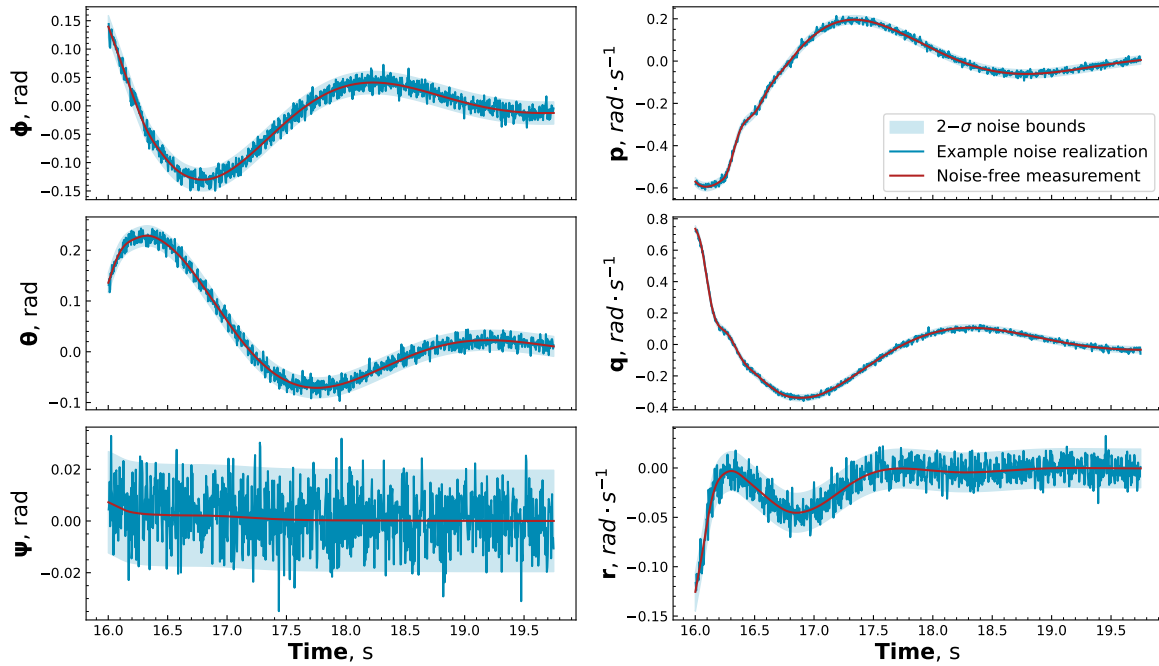
18

Fig. 15   Illustration of the addition of Gaussian noise to the noise free measurements (red) of the attitude and rotational rates to produce associated noisy realizations (blue). The shaded region represents the $2-\sigma$ bounds of the noise statistics used.
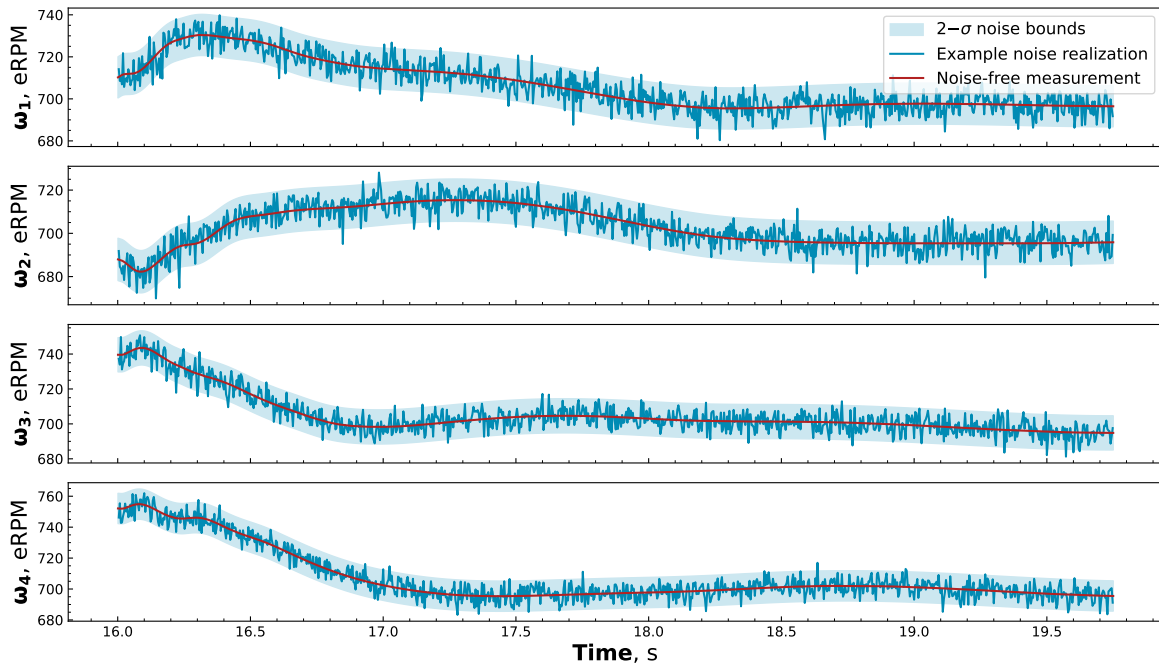

Fig. 16   Illustration of the addition of Gaussian noise to the noise free measurements (red) of the rotor speeds to produce associated noisy realizations (blue). The shaded region represents the $2-\sigma$ bounds of the noise statistics used.

19

C. HDBeetle Quadrotor Model Identification

1. Polynomial Candidate Regressors

The candidate regressors for the longitudinal *HDBeetle* models are chosen based on the grey-box models of Sun et al. [1, 10] and from experimentation. The candidate regressor structures for F_x, F_z, and M_y are summarized in eq. (29), eq. (30), and eq. (31) respectively.

$$
\begin{aligned}
\hat{F}_x = \ & X_0 \cdot u + X_1 \cdot \sin\theta + P^3(u, |v|, w) \cdot \{1, \omega_{avg}, q, U_q, \sin\theta, \cos\theta\} \\
& + P^3(q, |r|) \cdot \{1, \omega_{avg}\} + P^3(\mu_x, |\mu_y|, \mu_z) \cdot \{1, \omega_{avg}, q, U_q, \sin\theta, \cos\theta\} \\
& + P^3(U_q, |U_p|, |U_r|) \cdot \{1, \sin\theta, \cos\theta, \omega_{avg}\} + P^2(\omega_{avg}) \cdot \{\sin\theta\}
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
\hat{F}_z = \ & Z_0 \cdot w + Z_1 \cdot \cos\theta \cdot \cos\phi \\
& + P^3(|u|, |v|, w) \cdot \{1, \omega_{avg}, |p|, |q|, |r|, |U_p|, |U_q|, |U_r|, \sin\theta, \cos\theta, \sin\phi, \cos\phi\} \\
& + P^3(|p|, |q|, |r|) \cdot \{1, \omega_{avg}\} \\
& + P^3(|\mu_x|, |\mu_y|, \mu_z) \cdot \{1, \omega_{avg}, |p|, |q|, |r|, |U_p|, |U_q|, |U_r|, \sin\theta, \cos\theta, \sin\phi, \cos\phi\} \\
& + P^3(|U_q|, |U_p|, |U_r|) \cdot \{1, \sin\psi, \cos\psi, \omega_{avg}\} + P^4(\omega_{avg}) \cdot \{1, \sin\theta, \cos\theta, \sin\phi, \cos\phi\}
\end{aligned}
\tag{30}
$$

$$
\begin{aligned}
\hat{M}y = \ & M_0 \cdot q + M_1 \cdot U_q + P^4(u, |v|, w) \cdot \{1, \omega_{avg}, q, U_q, \sin\theta, \cos\theta\} \\
& + P^4(|p|, q, |r|) \cdot \{1, \omega_{avg}, \sin\theta, \cos\theta\} + P^4(\mu_x, |\mu_y|, \mu_z) \cdot \{1, \omega_{avg}, q, U_q, \sin\theta, \cos\theta\} \\
& + P^4(U_q, |U_p|, |U_r|) \cdot \{1, \sin\theta, \cos\theta, \omega_{avg}\} + P^4(\omega_{avg}) \cdot \{\sin\theta, \cos\theta\}
\end{aligned}
\tag{31}
$$

2. ANN Input Vectors

Table 8   Input vectors employed for the *HDBeetle* longitudinal model identification for both the ANN bootstrap and ANN quality driven techniques.

Model	Input variables
Fx	$u, |v|, w, q, |r|, \omega_{avg}, U_q, \sin\theta, \cos\theta, \mu_x, |\mu_y|, \mu_z, |U_r|, |U_p|$
Fz	$|u|, |v|, w, \omega_{avg}, |p|, |q|, |r|, |U_p|, |U_q|, |U_r|, \sin\theta, \cos\theta, \sin\phi, \cos\phi, |\mu_x|, |\mu_y|, \mu_z$
My	$u, |v|, w, \omega_{avg}, |p|, q, |r|, U_q, \sin\theta, \cos\theta, \mu_x, |\mu_y|, \mu_z, |U_p|, |U_r|$

3. Identified Polynomial Model Structures

Table 9   Identified polynomial longitudinal model structures, coefficients, and goodness-of-fit metrics for the *HDBeetle*. The rows indicate the order of addition to the model, as selected by the stepwise regression algorithm. Rows in grey denote fixed regressors.

Fx Regressor	Coefficient	R2	Fz Regressor	Coefficient	R2	My Regressor	Coefficient	R2
bias	-7.542e-02	-	*bias*	2.413e-01	-	*bias*	6.892e-03	-
u	2.884e-02	-	w	-7.440e-02	-	q	7.872e-04	-
$\sin\theta$	-1.398e-01	0.93	$\cos\theta \cdot \cos\phi$	-6.245e-01	0.45	U_q	8.010e-04	0.73
$\omega_{avg} \cdot u^{1.0}$	-4.537e-03	0.96	$\omega_{avg}^{2.0}$	-2.946e-04	0.95	$\omega_{avg} \cdot \mu_z^{2.0}$	-7.295e-06	0.92
$\cos\theta \cdot u^{3.0}$	1.636e-03	0.97	$\sin\theta \cdot |u|^{1.0} \cdot w^{2.0}$	-3.229e-03	0.96	$\omega_{avg} \cdot \mu_x^{1.0} \cdot \mu_z^{1.0}$	1.338e-04	0.92
—	-	-	—	-	-	$\mu_x^{1.0}$	1.217e-02	0.94

References

[1] Sun, S., de Visser, C., and Chu, Q., "Quadrotor Gray-Box Model Identification from High-Speed Flight Data," *Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 645–661. https://doi.org/10.2514/1.C035135, URL https://doi.org/10.2514/1.C035135.

[2] Powers, C., Mellinger, D., Kushleyev, A., Kothmann, B., and Kumar, V., "Influence of aerodynamics and proximity effects in quadrotor flight," *Experimental robotics*, Springer, 2013, pp. 289–302. https://doi.org/https://doi.org/10.1007/978-3-319-00065-7_21.

[3] Bansal, S., Akametalu, A. K., Jiang, F. J., Laine, F., and Tomlin, C. J., "Learning quadrotor dynamics using neural network for flight control," Institute of Electrical and Electronics Engineers Inc., 2016, pp. 4653–4660. https://doi.org/10.1109/CDC.2016.7798978.

[4] Bauersfeld, L., Kaufmann, E., Foehn, P., Sun, S., and Scaramuzza, D., "NeuroBEM: Hybrid Aerodynamic Quadrotor Model," *CoRR*, Vol. abs/2106.08015, 2021. URL https://arxiv.org/abs/2106.08015.

[5] Schumann, J., Gupta, P., and Nelson, S., "On verification and validation of neural network based controllers," *In Proceedings, International Conference on Engineering Applications of Neural Networks*, 2003, p. 47.

[6] Khosravi, A., Nahavandi, S., Creighton, D., and Atiya, A. F., "Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances," *IEEE Transactions on Neural Networks*, Vol. 22, No. 9, 2011, pp. 1341–1356. https://doi.org/10.1109/TNN.2011.2162110.

[7] Khosravi, A., Nahavandi, S., Creighton, D., and Atiya, A. F., "Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals," *IEEE Transactions on Neural Networks*, Vol. 22, No. 3, 2011, pp. 337–346. https://doi.org/10.1109/TNN.2010.2096824.

[8] Pearce, T., Zaki, M., Brintrup, A., and Neely, A., "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach," , 2018. URL https://arxiv.org/abs/1802.07167.

[9] Pearce, T., Leibfried, F., Brintrup, A., Zaki, M., and Neely, A., "Uncertainty in Neural Networks: Approximately Bayesian Ensembling," , 2020. URL https://arxiv.org/abs/1810.05546.

[10] Sun, S., Wang, X., Chu, Q., and d. Visser, C., "Incremental Nonlinear Fault-Tolerant Control of a Quadrotor With Complete Loss of Two Opposing Rotors," *IEEE Transactions on Robotics*, 2020, pp. 1–15. https://doi.org/10.1109/TRO.2020.3010626.

[11] Svacha, J., Mohta, K., and Kumar, V., "Improving quadrotor trajectory tracking by compensating for aerodynamic effects," *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 860–866. https://doi.org/10.1109/ICUAS.2017.7991501.

[12] Klein, V., and Morelli, E. A., *Aircraft System Identification: Theory and Practice*, 2006. https://doi.org/10.2514/4.861505.

[13] Gonzalez, R., and Dabove, P., "Performance assessment of an ultra low-cost inertial measurement unit for ground vehicle navigation," *Sensors*, Vol. 19, No. 18, 2019, p. 3865.

[14] Armanini, S. F., Karásek, M., de Croon, G. C. H. E., and de Visser, C. C., "Onboard/Offboard Sensor Fusion for High-Fidelity Flapping-Wing Robot Flight Data," *Journal of Guidance, Control, and Dynamics*, Vol. 40, 2017, pp. 2121–2132. https://doi.org/10.2514/1.G002527, URL https://arc.aiaa.org/doi/10.2514/1.G002527.