

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

A comparative study of grey-box structuring techniques for multirotor UAV dynamics under different wind conditions

Tesi di Laurea Magistrale in
Aeronautical Engineering - Ingegneria Aeronautica

Author: **Luca Scorbati**

Student ID: 225144
Advisor: Prof. Marco Lovera
Co-advisors: Enrico Barbiero
Academic Year: 2025-26

Abstract

Accurate dynamic models are essential for the analysis and control design of multirotor Unmanned Aerial Vehicles (UAVs). System identification is a data-driven framework for modeling dynamical systems from experimental data. Open-loop instability requires experiments to be performed under feedback, thus restricting the class of applicable methods. Closed-loop Subspace Identification Methods can be used for this purpose; however, they provide black-box models, whereas it may be desirable to express the dynamics in terms of physically meaningful parameters. This thesis investigates the structured identification of lateral dynamics of a multirotor UAV by comparing two grey-box estimation strategies: $\mathcal{H}_\infty$ model matching and frequency-domain Output-Error minimization. Both procedures start by estimating unstructured models using the optimized Predictor-Based Subspace Identification (PBSID$_{opt}$) algorithm. A dedicated experimental campaign is carried out with the ANT-X drone and the WindShaper facility to reproduce controlled wind conditions in the laboratory. Experiments are performed at different WindShaper power levels to obtain models under varying wind conditions. The selected black-box models are matched with a physically parametrized linear model of the lateral dynamics. The two structured estimation techniques are applied to the same modeling framework, and their accuracy and robustness are assessed through Monte Carlo simulations based on the uncertainty of the unstructured estimates. The results show that both approaches can recover grey-box models reproducing the dominant lateral dynamics in a specified frequency range; however, the Monte Carlo analysis highlights limited robustness of the estimated physical parameters. The comparison suggests that structured model matching can provide useful physically interpretable models, but the reliability of the recovered parameters depends strongly on the quality and uncertainty of the black-box estimate. Overall, the thesis compares two grey-box identification procedures and clarifies their applicability to multirotor UAV lateral dynamics.

Keywords: quadrotor, system identification, grey-box models, $\mathcal{H}_\infty$ model matching, Output-Error minimization

Sommario

Modelli dinamici accurati sono essenziali per l'analisi e la progettazione di Aeromobili a Pilotaggio Remoto (APR) multirotore. L'identificazione di sistemi fornisce un *framework* basato sui dati per modellare sistemi dinamici da misure di dati sperimentali. L'instabilità in anello aperto richiede che gli esperimenti di identificazione vengano effettuati in retroazione, restringendo la classe di metodi di identificazione che possono essere applicati. Una sottoclasse dei Metodi di Identificazione ai Sottospazi può essere usata per questo scopo; tuttavia, fornisce modelli *black-box*, mentre potrebbe essere desiderabile esprimere il modello in termini di parametri fisicamente significativi. Questa tesi indaga l'identificazione strutturata della dinamica laterale di un APR multirotore confrontando due strategie di stima *grey-box*: il *model matching* $\mathcal{H}_\infty$ e la minimizzazione dell'*Output-Error* nel dominio della frequenza. Entrambe le procedure iniziano dalla stima di modelli non strutturati usando l'algoritmo *optimized Predictor-Based Subspace Identification* (PBSID_{opt}). Una campagna sperimentale dedicata è stata effettuata usando il drone ANT-X e il *WindShaper*, al fine di riprodurre condizioni di vento controllate in laboratorio. Gli esperimenti sono stati condotti a diversi livelli di potenza del *WindShaper*, così da ottenere modelli in condizioni di vento variabili. I modelli *black-box* selezionati sono poi approssimati mediante un modello lineare fisicamente parametrizzato della dinamica laterale: le due procedure di stima strutturata sono applicate allo stesso *framework* di modellazione, e la loro accuratezza e robustezza è valutata attraverso un'analisi Monte Carlo basata sull'incertezza delle stime non strutturate. I risultati mostrano che entrambi gli approcci possono recuperare modelli *grey-box* che riproducono la dinamica laterale dominante in un intervallo di frequenze specificato; tuttavia, l'analisi Monte Carlo evidenzia una limitata robustezza dei parametri fisici stimati. Il confronto dunque suggerisce che il *model matching* strutturato può fornire dei modelli fisicamente interpretabili, ma che l'affidabilità dei parametri ottenuti rimane fortemente dipendente dalla qualità e incertezza delle stime *black-box*. Nel complesso, la tesi fornisce un confronto metodologico di due procedure di identificazione *grey-box* e chiarisce la loro applicabilità alla dinamica laterale di un APR multirotore.

Parole chiave: quadrirotore, identificazione di sistema, modelli *grey-box*, *model matching*

$\mathcal{H}_\infty$, minimizzazione *Output-Error*

Contents

Abstract **i**

Sommario **iii**

Contents **v**

Introduction **1**

1 Modeling of quadrotor UAVs **5**
1.1 Quadrotor configuration . . . 5
1.2 Reference frames . . . 6
1.3 Rotation formalism . . . 7
1.4 Flight Dynamics Equations . . . 7
1.4.1 Attitude and position kinematics . . . 7
1.4.2 Translational and rotational motion . . . 8
1.4.3 External forces and moments . . . 10
1.5 Linearized mathematical model . . . 10
1.6 Linear Time-Invariant model . . . 13

2 Black-box PBSID$_{opt}$ algorithm and structured model matching **17**
2.1 Subspace methods . . . 17
2.1.1 PBSID$_{opt}$ algorithm . . . 19
2.1.2 PBSID toolbox and implementation . . . 23
2.2 Structured model matching . . . 24
2.2.1 $\mathcal{H}_\infty$-norm minimization . . . 25
2.2.2 Frequency-domain Output-Error minimization . . . 26
2.3 Prefiltering for type correction . . . 28

3 Experimental activities **31**
3.1 Aerospace Systems and Control Laboratory . . . 31

3.1.1 WindShaper . . . 31
3.1.2 ANT-X drone . . . 33
3.2 Previous works . . . 34
3.3 Experimental procedure setup . . . 35
3.4 Identification and validation datasets . . . 36

4 Validation procedure of unstructured models 39
4.1 Preliminary analysis of the experimental dataset . . . 39
4.2 PBSID$_{opt}$ black-box identification . . . 43
4.3 Results and validation . . . 44
4.3.1 Cross-validation on time histories . . . 45
4.3.2 Frequency-domain validation . . . 51

5 Validation procedure of structured models 55
5.1 $\mathcal{H}_\infty$ model matching setup . . . 55
5.2 Frequency-domain OE minimization setup . . . 57
5.3 Validation . . . 58
5.3.1 $\mathcal{H}_\infty$ model matching . . . 59
5.3.2 Frequency-domain OE model matching . . . 68
5.4 Results . . . 76
5.4.1 Estimated parameters . . . 77
5.4.2 Monte Carlo analysis . . . 78
5.5 Comparison between $\mathcal{H}_\infty$ model matching and frequency-domain Output-Error minimization . . . 86

Conclusions and further developments 89

Bibliography 91

List of Figures 97

List of Tables 103

Introduction

System identification is the field of modeling dynamical systems from experimental data, providing a data-driven alternative or complement to models derived analytically from *first principles* [1]. In aerospace applications, robust identification procedures covering experiment design, input-output data acquisition, and model estimation are essential for obtaining reliable dynamic models, which are then used for simulations, predicting flying qualities, and capturing the relevant dynamics of the system over the time scales and the frequency ranges of interest around various operating points. Furthermore, a reliable plant model is of paramount importance for the development of safe and robust control systems, especially in the case of unstable aircraft such as helicopters and multirotors.
Among the classes of identification techniques, a prominent class is represented by the Subspace Identification Methods (SIMs), which have been used successfully for the estimation of multi-input multi-output (MIMO) systems in a number of industrial applications. The theoretical roots of SIMs lie in the work of Ho and Kalman [2], which first introduced a realization procedure for recovering state-space models from input-output data through Hankel matrices. Modern SIMs were later consolidated with the works of Van Overschee and De Moor with the N4SID approach [3], and Verhaegen with the MOESP algorithm [4].
One limitation of classical open-loop SIMs is that they rely on the assumption that the input is uncorrelated with the noise processes affecting the system. This assumption is generally violated when the system operates under feedback, since the control input depends on measured outputs and may therefore become correlated with process and measurement noise. The work of Van der Veen et al. [5] provides a detailed overview of subspace identification methods designed to operate on data collected under closed-loop conditions. This issue is particularly relevant in practical applications where feedback cannot be removed, for instance because the open-loop plant is unstable or because the system must remain close to an operating condition around which a linear model is meaningful. The authors show that many successful closed-loop subspace identification methods share a common first step based on the estimation of a high-order VARX model which is used to handle the correlation issues introduced by feedback. Starting from this

common predictor framework, different algorithms can then be derived, including direct parametrization, realization-based methods, predictor-based subspace identification, and closed-loop MOESP. Among these approaches, Predictor-Based Subspace Identification (PBSID) algorithms, as described in [6], represent the current state-of-the-art, in particular in its optimized version (PBSID$_{opt}$) which features accurate performance paired with computational efficiency, making them attractive for the identification of systems operating under feedback [7, 8]. The applicability of PBSID$_{opt}$ has been demonstrated in several closed-loop identification problems, both in aerospace applications [9–13] and other fields [5, 14].

The methods described above can be categorized as *black-box* identification methods: the state-space representation is not parametrized *a priori* and it is not physically motivated. On the other hand, the need may arise to identify a model using a specific parametrization derived from the physics of the system: in this case, the derived model is called *grey-box*. Grey-box models are useful in that they exploit prior knowledge of the dynamics of the system, which may be useful in control design applications. A methodology to relate unstructured black-box models to structured grey-box ones is proposed in [15] by Bergamasco and Lovera: in their work, the problem is formulated as an input-output model-matching problem in the frequency domain, by minimizing the $\mathcal{H}_\infty$ norm of the difference between the unstructured and structured models. This avoids the explicit construction of the similarity transformation between the two state-space realizations and leads to an optimization problem that can be tackled with tools developed for fixed-structure control design. Subsequent works followed the same general framework in rotorcraft applications, combining time-domain subspace identification of a black-box state-space model with frequency-domain information obtained from sweep experiments, used to support and enforce the structured model-matching step [16]. This line of work is also related to the broader idea of exploiting black-box models obtained through Subspace Identification Methods to initialize or support the estimation of structured physical models [17, 18]. This $\mathcal{H}_\infty$ model-matching framework has been adopted in previous rotorcraft and multirotor UAV identification studies to recover structured, physically interpretable models from black-box state-space estimates [11–13, 19].

Output-Error (OE) minimization represents a different class of parameter estimation methods used for structured aircraft and rotorcraft models. In this framework, the values of the unknown parameters are adjusted so as to minimize a cost function representing the difference between the available data and the grey-box model, usually through an iterative algorithm. Depending on how the experimental information is represented, the Output-Error criterion can be formulated either in the time domain or in the frequency

domain [20, 21]. In flight dynamics applications, frequency-domain formulations are particularly common, since they allow the identification process to focus on the frequency range where the experimental data are sufficiently informative and where the model is expected to reproduce the relevant dynamics: software packages such as CIFER (Comprehensive Identification from FrEquency Responses) and SIDPAC (System IDentification Programs for AirCraft) provide established tools for deriving, processing, and validating frequency-response information from flight-test data, as well as for supporting subsequent frequency-domain model fitting [21–23].

Different strategies have been proposed in the literature to provide the frequency-domain information required by structured model-matching procedures, both in the OE and in the $\mathcal{H}_\infty$ frameworks. Classical frequency-domain aircraft and rotorcraft identification methods usually start from non-parametric frequency-response estimates obtained from flight-test data, for instance through frequency sweeps and spectral analysis tools available in CIFER [21, 24].

Other approaches combine time-domain and frequency-domain information: Bergamasco et al. propose a procedure in which a black-box state-space model is first estimated from time-domain data using subspace methods, while non-parametric frequency-response estimates are computed from sweep experiments and then enforced during the structured model-matching step [16]. This line of work is related to the idea of exploiting black-box models obtained through Subspace Identification Methods to initialize or support the estimation of structured physical models [17, 18].

The focus of this thesis is on the system identification of a small-scale multirotor UAV, concerning the lateral dynamics under varying flight conditions. An experimental campaign is carried out, with the aim of refining the data acquisition procedure and improving the reliability of the identified models with respect to previous results. Black-box identification is performed using the PBSID$_{opt}$ algorithm, and the results obtained are then used as the reference models for two different structuring procedures, namely the $\mathcal{H}_\infty$ model matching and a frequency-domain Output-Error minimization; after being validated against a dedicated experimental dataset, the two approaches are compared in order to identify the advantages of each method. Finally, the robustness of the results is assessed by means of a Monte Carlo simulation. The thesis is organized as follows:

- Chapter 1 describes the physical modeling of the UAV, leading to the formulation of a linear time-invariant model to be used in the structuring procedure;
- Chapter 2 details the theoretical aspects of subspace identification methods and of the different structuring procedures adopted;

- Chapter 3 explains the experimental activities performed in the ASCL laboratory, describing the laboratory setup and providing the identification and validation datasets;
- Chapter 4 presents the results obtained by the unstructured model estimation via PBSID$_{opt}$, focusing on the analysis of the experimental data to ensure that sufficient information is available and providing the validation procedure to assess the goodness of the results;
- Chapter 5 presents the results obtained with the two structuring procedures, comparing the results obtained with those of the black-box models and assessing the robustness of the parameter estimation by means of a Monte Carlo simulation.

1 | Modeling of quadrotor UAVs

This chapter presents a modeling procedure for a small multirotor UAV, leading to a physical model able to describe its operating behavior under varying flight conditions. The linearized flight dynamics equations will be derived, following which a state-space representation of the model will be defined through a set of physical parameters, in order to obtain a Linear Time-Invariant (LTI) model.

1.1. Quadrotor configuration

The quadrotor configuration is an example of underactuated mechanical system: since it has six degrees of freedom (DOFs) and can only command four independent control inputs, it is not possible to command an instantaneous acceleration of its generalized coordinates in an arbitrary trajectory. The classic quadrotor configuration, which will be considered going forward, consists of four propellers mounted pointing upward, meaning that it can produce a moment in any direction, but can only output a net force (thrust, F_p) in the vertical direction. This leads to an intrinsic coupling between rotational and translational kinematics: in order to change its position, the UAV first needs to generate a moment, which in turn leads to a change in attitude that allows it to translate. The four rotors are divided into clockwise (CW) and counterclockwise (CCW) counter-rotating pairs, mounted on the frame in a symmetric layout.
The basic flight operations are:

- **altitude control**: the flight control system (FCS) acts collectively on each rotor;
- **directional control**: the FCS reduces thrust on a set of rotors and increases it on the others, maintaining the same total F_p, thus producing an imbalance of the angular momentum which results in a yaw motion without changes in altitude;
- **horizontal position control**: the FCS increases thrust on one side and decreases it on the other, causing a change in attitude on the roll or pitch axis leading to translational motion.

In the following physical model derivation, the frame of the quadrotor is assumed to be

rigid, ground and gyroscopic effect are neglected, and aeroelastic phenomena, namely of the propellers, are not modeled.

1.2. Reference frames

Navigation frame $\mathcal{N}$

The origin of the navigation frame is placed in a generic point O_N on the Earth's surface. The first unit vector N points toward the geographic North, the second E points toward the geographic East, while the third D, aligned with the gravity field, points downward. This reference frame is also referred to as North-East-Down, or NED for brevity. Under the assumptions of a flat and non-rotating Earth, this frame can be considered inertial [25].

Body frame $\mathcal{B}$

The body frame is centered in the center of gravity of the aircraft, $O_B \equiv C_G$. The first axis X_B lies in the longitudinal plane of symmetry and points forward, the second Y_B is perpendicular to that plane of symmetry and points to the right. Finally, the third axis Z_B points downward to satisfy the right-hand rule.

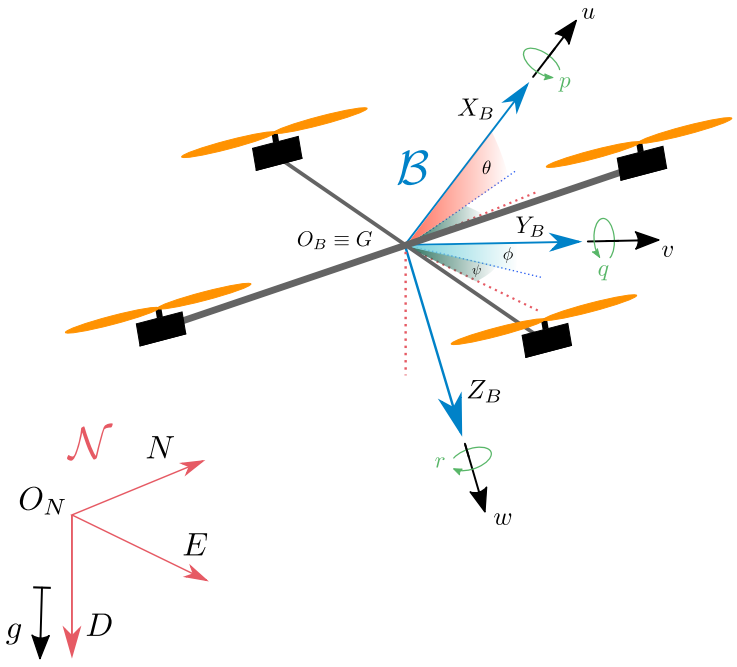


Figure 1.1: Quadrotor kinematics [12].

1.3. Rotation formalism

According to aeronautical convention, the 3-2-1 Euler angles (ϕ, θ, ψ) are used to go from the local $\mathcal{N}$ to the body $\mathcal{B}$ frame through a rigid rotation. Euler angles allow for the minimal representation of a rotation matrix as a sequential composition of 3 planar rotations, namely the yaw angle ψ rotation around the D-axis, pitch angle θ rotation around the E-axis, bank angle ϕ rotation around the N-axis. The rotation around the three axis are described with the following matrices

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \tag{1.1a}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{1.1b}$$

$$R_z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.1c}$$

Applying in the stated order the rotation matrices 1.1, the Euler rotation matrix from $\mathcal{N}$ to $\mathcal{B}$ is obtained:

$$R^{\mathcal{B}}_{\mathcal{N}}(\phi, \theta, \psi) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi C_\theta \\ C_\phi S_\theta C_\psi - S_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix}, \tag{1.2}$$

where $C_\alpha = \cos\alpha$ and $S_\alpha = \sin\alpha$ for the sake of brevity.

1.4. Flight Dynamics Equations

1.4.1. Attitude and position kinematics

The change of attitude of the quadrotor with respect to the inertial frame $\mathcal{N}$ evolves as:

$$\omega^{\mathcal{B}} := \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{1.3}$$

with the angular rates defined in the $\mathcal{B}$ frame.
The *equation of Coriolis* can be used to formulate the time derivative of a generic vector b in the frame $\mathcal{N}$ as [25]:

$$^{\mathcal{N}}\frac{db}{dt} = {}^{\mathcal{B}}\frac{db}{dt} + \omega \times b. \tag{1.4}$$

The velocity of a generic point P of the quadrotor in the frame $\mathcal{N}$ can be written referring to the center of gravity G, considering no relative displacement due to rigid body motion:

$$V_P = V_G + \omega \times (P - G) \tag{1.5}$$

where $V_G = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is the velocity of G. After defining the coordinates of $P = \begin{bmatrix} x & y & z \end{bmatrix}^T$, one can derive its acceleration a_P, as shown in detail in [26], which can be expanded and written in the basis of $\mathcal{B}$ as:

$$\begin{aligned} \ddot{x} &= \dot{u} + \dot{q}z - \dot{r}y + q\left(w + py - qx\right) - r\left(v + rx - pz\right), \\ \ddot{y} &= \dot{v} + \dot{r}x - \dot{p}z + r\left(u + qz - ry\right) - p\left(w + py - qx\right), \\ \ddot{z} &= \dot{w} + \dot{p}y - \dot{q}x + p\left(v + rx - pz\right) - q\left(u + qz - ry\right). \end{aligned} \tag{1.6}$$

thus allowing one to write the first and second cardinal equations of motion.

1.4.2. Translational and rotational motion

The first cardinal equation integrated over the aircraft's body $\mathcal{V}$ states that:

$$\int_{\mathcal{V}} df_P = \int_{\mathcal{V}} a_P \, dm. \tag{1.7}$$

Since $\mathcal{B}$ is centered in G, the static moments defined as

$$(S_x,\, S_y,\, S_z) \;\coloneqq\; \int_{\mathcal{V}} (x,\, y,\, z)\, dm \tag{1.8}$$

are null. Thus, equation (1.7), expressed in the body reference frame, can be written as

$$F = m\dot{V}_G + \omega \times (mV_G) \tag{1.9}$$

where m is the total mass of the aircraft and $F^{\mathcal{B}} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ is the total external force acting on the UAV.
The second cardinal equation integrated over the aircraft's body $\mathcal{V}$ states that:

$$\int_{\mathcal{V}} dm_G = \int_{\mathcal{V}} (P - G) \times a_P \, dm. \tag{1.10}$$

Let us first introduce the inertia tensor, represented in the body frame by the inertia matrix:

$$J_n \coloneqq \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix}. \tag{1.11}$$

whose terms, called moments of inertia, are evaluated in $\mathcal{B}$ and are defined as

$$\begin{aligned} J_{xy} &\coloneqq \int (xy)\, dm, & J_{xz} &\coloneqq \int (xz)\, dm, & J_{yz} &\coloneqq \int (yz)\, dm, \\ J_{xx} &\coloneqq \int (y^2 + z^2)\, dm, & J_{yy} &\coloneqq \int (x^2 + z^2)\, dm, & J_{zz} &\coloneqq \int (x^2 + y^2)\, dm. \end{aligned} \tag{1.12}$$

Since the reference frame is centered in G and the quadrotor, by construction, exhibits bilateral symmetry with respect to its principal axis, the off-diagonal terms of (1.11) are null.
Equation (1.10), expressed in the body reference frame, can be written as

$$M_G \;=\; J_n\dot{\omega} \;+\; \omega \times (J_n\omega) \tag{1.13}$$

where $M_G^{\mathcal{B}} = \begin{bmatrix} \mathcal{L} & \mathcal{M} & \mathcal{N} \end{bmatrix}^T$ is the total moment about G of the external forces around the body frame axes.

1.4.3. External forces and moments

The external forces and moments acting on the UAV are the gravitational and aerodynamic forces, the latter of which also include the effects of propulsion.

Gravitational forces

As a consequence of the Flat-Earth approximation, the gravity field is considered uniform and always orthogonal to the surface. In the NED frame, that means that gravitational forces point downward:

$$F_g^{\mathcal{N}} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \tag{1.14}$$

Its expression in the body frame is obtained by a change of basis through equation (1.2):

$$F_g^{\mathcal{B}} = R_{\mathcal{N}}^{\mathcal{B}}(\phi, \theta, \psi)\, F_g^{\mathcal{N}} = mg \begin{bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{bmatrix}. \tag{1.15}$$

Aerodynamic forces

For the aerodynamics, the functional relation is expressed as

$$F_{aero} = F_{aero}\,(V_G, \omega, \delta, \mathrm{Re}, \mathrm{Ma}), \tag{1.16}$$

$$M_{aero} = M_{aero}\,(V_G, \omega, \delta, \mathrm{Re}, \mathrm{Ma}), \tag{1.17}$$

where $\delta := \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{dir} & \delta_{ver} \end{bmatrix}^T$ is the *control input*, Re is the *Reynolds number* and Ma is the *Mach number*.

1.5. Linearized mathematical model

To derive a linear model, the equations of motion are linearized around a reference or *trim* condition, which is defined as a flight trajectory for which the linear and angular velocity

of the aircraft remain constant for a constant command input. To address the scope of this study, the dynamics must be linearized for different trim conditions that represent flight at different speeds.

The *hovering* condition is a trim condition with null speed, corresponding to $V_{G_0} = 0$, $\omega_0 = 0$, $\phi_0 = 0$, $\theta_0 = 0$ and $\psi_0 = \overline{\psi}$. In addition to this, a general flying condition for the roll dynamics is considered, with $u = 0$, $v \neq 0$, $w \neq 0$, $\omega_0 = 0$, $\phi_0 \neq 0$, $\theta_0 = 0$ and $\psi_0 = \overline{\psi}$. For small perturbations, the solution of the perturbed equations can be approximated by considering the first-order terms of the Taylor expansion around a general $\phi_0 \neq 0$, $\theta_0 \neq 0$, $\psi_0 \neq 0$ trim condition.

Indicating with $\Delta \cdot$ the perturbations, the gravity force is linearized as follows:

$$\begin{aligned} mg \begin{bmatrix} -\sin(\theta_0 + \Delta\theta) \\ \sin(\phi_0 + \Delta\phi)\cos(\theta_0 + \Delta\theta) \\ \cos(\phi_0 + \Delta\phi)\cos(\theta_0 + \Delta\theta) \end{bmatrix} &\approx \underbrace{mg \begin{bmatrix} -\sin\theta_0 \\ \sin\phi_0\cos\theta_0 \\ \cos\phi_0\cos\theta_0 \end{bmatrix}}_{\text{equilibrium equation}} + \\ + mg \begin{bmatrix} -\cos\theta_0 & 0 \\ -\sin\phi_0\sin\theta_0 & \cos\phi_0\cos\theta_0 \\ -\cos\phi_0\sin\theta_0 & -\sin\phi_0\cos\theta_0 \end{bmatrix} &\begin{bmatrix} \Delta\theta \\ \Delta\phi \end{bmatrix} + \mathcal{O}(\Delta^2). \end{aligned} \tag{1.18}$$

Considering roll dynamics ($\theta_0 = 0$), equation (1.18) becomes:

$$F_g \approx mg \underbrace{\begin{bmatrix} 0 \\ \sin\phi_0 \\ \cos\phi_0 \end{bmatrix}}_{\text{equilibrium equation}} + mg \begin{bmatrix} -\Delta\theta \\ \cos\phi_0\,\Delta\phi \\ -\sin\phi_0\,\Delta\phi \end{bmatrix} + \mathcal{O}(\Delta^2). \tag{1.19}$$

The aerodynamic forces are linearized around the trim condition through the use of *Jacobian matrices*:

$$\begin{aligned} F_{aero} &\sim F_{aero}\,(V_G, 0, \delta_0, \mathrm{Re}, \mathrm{Ma}) + J_{F_{V_G}}\Delta V_G + J_{F_\omega}\Delta\omega + J_{F_\delta}\Delta\delta, \\ M_{aero} &\sim M_{aero}\,(V_G, 0, \delta_0, \mathrm{Re}, \mathrm{Ma}) + J_{M_{V_G}}\Delta V_G + J_{M_\omega}\Delta\omega + J_{M_\delta}\Delta\delta, \end{aligned} \tag{1.20}$$

where

$$J_{F_\delta} \coloneqq \begin{bmatrix} 0 & \frac{\partial X}{\partial \delta_{lon}} & 0 & 0 \\ \frac{\partial Y}{\partial \delta_{lat}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial Z}{\partial \delta_{ver}} \end{bmatrix}, \qquad J_{M_\delta} \coloneqq \begin{bmatrix} \frac{\partial L}{\partial \delta_{lat}} & 0 & 0 & 0 \\ 0 & \frac{\partial M}{\partial \delta_{lon}} & 0 & 0 \\ 0 & 0 & \frac{\partial N}{\partial \delta_{ver}} & 0 \end{bmatrix}, \tag{1.21}$$

are the so-called *control derivatives*, which express the UAV's sensitivity with respect to the control inputs, and

$$\begin{aligned} J_{F_{V_G}} &\coloneqq \begin{bmatrix} \frac{\partial X}{\partial u} & 0 & 0 \\ 0 & \frac{\partial Y}{\partial v} & 0 \\ 0 & 0 & \frac{\partial Z}{\partial w} \end{bmatrix}, \qquad J_{F_\omega} \coloneqq \begin{bmatrix} 0 & \frac{\partial X}{\partial q} & 0 \\ \frac{\partial Y}{\partial p} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ J_{M_{V_G}} &\coloneqq \begin{bmatrix} 0 & \frac{\partial L}{\partial v} & 0 \\ \frac{\partial M}{\partial u} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad J_{M_\omega} \coloneqq \begin{bmatrix} \frac{\partial L}{\partial p} & 0 & 0 \\ 0 & \frac{\partial M}{\partial q} & 0 \\ 0 & 0 & \frac{\partial N}{\partial r} \end{bmatrix}. \end{aligned} \tag{1.22}$$

are the so-called *stability derivatives*, which express the UAV's sensitivity to external disturbances.

Finally, the external forces and moments in their linearized forms obtained in equations (1.18) and (1.20) are substituted into equations (1.9) and (1.13), which after neglecting second order terms can be written as:

$$
\begin{aligned}
m\,\Delta\dot{u} &= -mg\,\Delta\theta + \frac{\partial X}{\partial \delta_{lon}}\Delta\delta_{lon} + \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial q}\Delta q,\\
m\,\Delta\dot{v} &= mg\cos\phi_0\,\Delta\phi + \frac{\partial Y}{\partial \delta_{lat}}\Delta\delta_{lat} + \frac{\partial Y}{\partial v}\Delta v + \frac{\partial Y}{\partial p}\Delta p,\\
m\,\Delta\dot{w} &= -mg\sin\phi_0\,\Delta\phi + \frac{\partial Z}{\partial \delta_{ver}}\Delta\delta_{ver} + \frac{\partial Z}{\partial w}\Delta w,\\
J_{xx}\,\Delta\dot{p} &= \frac{\partial \mathcal{L}}{\partial \delta_{lat}}\Delta\delta_{lat} + \frac{\partial \mathcal{L}}{\partial v}\Delta v + \frac{\partial \mathcal{L}}{\partial p}\Delta p,\\
J_{yy}\,\Delta\dot{q} &= \frac{\partial \mathcal{M}}{\partial \delta_{lon}}\Delta\delta_{lon} + \frac{\partial \mathcal{M}}{\partial u}\Delta u + \frac{\partial \mathcal{M}}{\partial q}\Delta q,\\
J_{zz}\,\Delta\dot{r} &= \frac{\partial \mathcal{N}}{\partial \delta_{dir}}\Delta\delta_{dir} + \frac{\partial \mathcal{N}}{\partial r}\Delta r.
\end{aligned}
\tag{1.23}
$$

1.6. Linear Time-Invariant model

From the linearized flight dynamics equations, a state space model can be derived, that is a set of first-order differential equations of *state variables* x. The state variables are the minimum set of system variables that fully describe the system and its response to any given set of *inputs* u:

$$
\dot{x} = A\,x + B\,u, \tag{1.24}
$$

$$
x \coloneqq \begin{bmatrix} v & p & \phi & u & q & \theta & r & \Delta\psi & w \end{bmatrix}^T, \qquad u \coloneqq \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{dir} & \Delta\delta_{ver} \end{bmatrix}^T, \tag{1.25}
$$

$$
A = \begin{bmatrix}
Y_v & Y_p & g\cos\phi_0 & 0 & 0 & 0 & 0 & 0\\
L_v & L_p & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & X_u & X_q & -g & 0 & 0 & 0\\
0 & 0 & M_u & M_q & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & N_r & 0 & 1 & 0\\
0 & 0 & -g\sin\phi_0 & 0 & 0 & 0 & 0 & Z_w
\end{bmatrix}, \qquad
B = \begin{bmatrix}
Y_\delta & 0 & 0 & 0\\
L_\delta & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & X_\delta & 0 & 0\\
0 & M_\delta & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & N_\delta & 0\\
0 & 0 & 0 & Z_\delta
\end{bmatrix}. \tag{1.26}
$$

One may notice from the block diagonal structure of (1.26) that the dynamics around the various axes can be decoupled.
For control purposes, it is useful to relate the state equations to measurable variables through an output equation. In particular, suitable outputs for flight control systems are the ones that can be measured with on-board sensors: angular rates and inertial accelerations in the body frame are directly measured by means of a strap-down Inertial Measurements Unit (IMU). Moreover, the IMU accelerometers measure external accelerations, excluding the contribution of gravity:

$$a_x = \dot{u} + g\sin\theta = X_u\,u + X_q\,q + X_\delta\,\delta_{lon}, \tag{1.27a}$$

$$a_y = \dot{v} - g\sin\phi = Y_v\,v + Y_p\,p + Y_\delta\,\delta_{lat}, \tag{1.27b}$$

$$a_z = \dot{w} - g = Z_w\,w - g + Z_\delta\,\delta_{ver}. \tag{1.27c}$$

Thus, the following structure is considered:

$$\begin{aligned} \dot{x} &= A\,x + B\,u \\ y &= C\,x + D\,u \end{aligned} \tag{1.28}$$

where $y \coloneqq \begin{bmatrix} p & a_y & q & a_x & r & a_z + g \end{bmatrix}^T$ is the output vector, and the output matrix and the feedthrough matrix are:

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ Y_v & Y_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_u & X_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_w \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ Y_\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & X_\delta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_\delta \end{bmatrix}. \tag{1.29}$$

As can be seen from the block-diagonal structure of equations (1.26), (1.29), the dynamics around the X, Y and Z axis can be decoupled as follows:

- *Longitudinal dynamics*

$$\begin{cases} \begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_q & -g \\ M_u & M_q & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_\delta \\ M_\delta \\ 0 \end{bmatrix} \delta_{lon}, \\ \begin{bmatrix} q \\ a_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ X_u & X_q & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ X_\delta \end{bmatrix} \delta_{lon}. \end{cases} \tag{1.30}$$

- *Lateral dynamics*

$$\begin{cases} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & g\cos\phi_0 \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ L_\delta \\ 0 \end{bmatrix} \delta_{lat}, \\ \begin{bmatrix} p \\ a_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ Y_v & Y_p & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ Y_\delta \end{bmatrix} \delta_{lat}. \end{cases} \tag{1.31}$$

- *Vertical dynamics*

$$\begin{bmatrix} \dot{\phi} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -g\sin\phi_0 & Z_w \end{bmatrix} \begin{bmatrix} \phi \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ Z_\delta \end{bmatrix} \delta_{ver}. \tag{1.32}$$

- *Directional dynamics*

$$\begin{cases} \begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} N_r & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} N_\delta \\ 0 \end{bmatrix} \delta_{dir}, \\ r = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix}. \end{cases} \tag{1.33}$$

In the Laplace domain, one can define the transfer function from input u to output y as:

$$G(s) \;=\; C\,(sI - A)^{-1}\,B \;+\; D \tag{1.34}$$

In particular, the Single-Input Multi-Output (SIMO) model for lateral dynamics yields the following transfer functions:

$$G_{\text{lat}}(s) = \begin{bmatrix} G_p(s) \\ G_{a_y}(s) \end{bmatrix} = \begin{bmatrix} \dfrac{s(L_\delta s - L_\delta Y_v + L_v Y_\delta)}{s^3 - (L_p + Y_v)\, s^2 + (L_p Y_v - L_v Y_p)\, s - L_v g \cos\phi_0} \\ \dfrac{Y_\delta s^3 + (L_\delta Y_p - L_p Y_\delta)\, s^2 + L_\delta Y_v g \cos(\phi_0) - L_v Y_\delta g \cos\phi_0}{s^3 - (L_p + Y_v)\, s^2 + (L_p Y_v - L_v Y_p)\, s - L_v g \cos\phi_0} \end{bmatrix}. \tag{1.35}$$

The formulated transfer functions can be used in the estimation problem of the physical parameters of a multirotor UAV. Equation (1.35) shows that, considering only $G_p(s)$, not enough independent information is available on all the physical derivatives appearing in $\vartheta_{lat} = [Y_v \; Y_p \; L_v \; L_p \; Y_\delta \; L_\delta]^T$: the numerator and denominator of $G_p(s)$ constrain only some combinations of these leaving at least one degree of freedom in their estimation. The additional transfer function $G_{a_y}(s)$ introduces further independent relations, thereby making the SIMO formulation more suitable for recovering a physically meaningful grey-box model. Since the lateral dynamics are inherently unstable, the estimated model is expected to exhibit *open-loop instability*; moreover, the transfer function $G_p(s)$ has a zero in the origin (i.e. for $s = 0$), therefore it is not of the zero type (this matter will be dealt with in Section 2.3).

2 | Black-box PBSID$_{opt}$ algorithm and structured model matching

In this chapter, subspace methods are first introduced, along with the challenges which may occur when trying to identify unstable MIMO systems and when operating in closed-loop. The PBSID$_{opt}$ algorithm is then presented as a viable option for dealing with said issues.
Then, two different model-matching methods are introduced to relate the identified black-box models with structured ones.
Finally, prefiltering for type correction is presented as a way to improve the model matching techniques when dealing with non-zero type transfer functions.

2.1. Subspace methods

Subspace Identification Methods (SIMs) are a class of identification techniques which, unlike the classical optimization-based approach, don't require the definition of a model structure depending explicitly on a vector of unknown parameters, nor on the explicit construction of a cost function. Indeed, SIMs first estimate a subspace related to the system state sequence or to the extended observability matrix from structured input-output data matrices. The state-space matrices are then recovered through least-squares [1, 27].

The origin of subspace identification can be understood from deterministic realization [2]. Consider first the noise-free discrete-time LTI system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k). \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times m}$ are respectively the state, input, output and feed-through matrices.
If the system is excited by an impulse input, its impulse response is described by the

sequence of Markov parameters

$$h(0) = D, \qquad h(k) = CA^{k-1}B, \quad k \geq 1. \tag{2.2}$$

These coefficients completely characterize the input-output behavior of the system. By arranging them into a block Hankel matrix, one obtains

$$H_{s,N} = \begin{bmatrix} h(1) & h(2) & \cdots & h(N) \\ h(2) & h(3) & \cdots & h(N+1) \\ \vdots & \vdots & \ddots & \vdots \\ h(s) & h(s+1) & \cdots & h(s+N-1) \end{bmatrix}. \tag{2.3}$$

Using the expression of the Markov parameters in (2.2), the Hankel matrix admits the factorization [27]:

$$H_{s,N} = \Gamma_s \mathcal{C}_N, \tag{2.4}$$

where

$$\Gamma_s = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}, \qquad \mathcal{C}_N = \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix}. \tag{2.5}$$

are the *extended observability matrix* and the *extended controllability matrix*, respectively. Consequently, the column space of $H_{s,N}$ coincides with the observability subspace of the system and can be estimated through a singular value decomposition (SVD) of the Hankel matrix. Once a basis for this subspace has been obtained, the state-space matrices can be reconstructed using linear algebra operations.
It is important to remark that the matrices can be estimated up to a similarity transformation T:

$$\tilde{A} = T^{-1}AT, \quad \tilde{B} = T^{-1}B, \quad \tilde{C} = CT \tag{2.6}$$

The two realizations describe the same input-output behavior, since they are related only by a change of basis in the state space. The realization procedure described above assumes that the Markov parameters, and therefore the impulse response of the system, are directly available. In practical applications, however, system identification is rarely performed from impulse response data, and noise cannot be neglected. Furthermore, in the case of inherently unstable systems, such as multirotor UAVs, input-output measurements are often collected while the plant operates under feedback control, which means that the

input is affected by the measured output and is therefore correlated with the noise [28]. These limitations prevent the direct application of realization-based techniques to most practical identification problems. Modern subspace identification methods therefore aim at estimating the relevant system subspaces directly from noisy input-output data. The following section introduces an example of a predictor-based subspace method that can be applied when the available measurements are collected under closed-loop operating conditions: the key idea is to write the system in predictor form and to then estimate the current state from a finite window of past measured inputs and outputs.

2.1.1. PBSID$_{opt}$ algorithm

The system defined in equation (2.1) can be generalized to a discrete-time, linear, time-invariant system:

$$\begin{aligned} x(k+1) &= A\,x(k) + B\,u(k) + w(k) \\ y(k) &= C\,x(k) + D\,u(k) + v(k), \end{aligned} \tag{2.7}$$

where $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^l$, $u(k) \in \mathbb{R}^{m_u}$, are the states and the sample values of the system's output and input; $w(k) \in \mathbb{R}^n$ and $v(k) \in \mathbb{R}^l$ are respectively process and measurement noise, both with zero mean and finite variance, such that:

$$\mathbb{E}\left[\begin{Bmatrix} w_i \\ v_i \end{Bmatrix} \begin{Bmatrix} w_j & v_j \end{Bmatrix}\right] := \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{ij}, \tag{2.8}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator, while δ_{ij} is the Kronecker delta.
This is also known as a *combined deterministic-stochastic system* [1], due to the presence of both noise and external input.
The system (2.7) can be written in its *innovation form*:

$$\begin{aligned} \tilde{x}(k+1) &= A\,\tilde{x}(k) + B\,u(k) + K\,e(k) \\ \tilde{y}(k) &= C\,\tilde{x}(k) + D\,u(k), \end{aligned} \tag{2.9}$$

where $\tilde{x}(k)$ is the state estimate, $\tilde{y}(k)$ is the output estimate, $K \in \mathbb{R}^{n\times l}$ is the *steady-state Kalman gain* found from solving a *Discrete Algebraic Riccati Equation* (DARE), and $e(k)$ is the *innovation noise*, defined as:

$$e(k) = y(k) - \tilde{y}(k) = y(k) - C\,\tilde{x}(k) - D\,u(k). \tag{2.10}$$

Substituting equation (2.10) into the state equation of (2.9):

$$\tilde{x}(k+1) \;=\; A\,\tilde{x}(k) \;+\; B\,u(k) \;+\; K\,[\,y(k) \;-\; C\,\tilde{x}(k) \;-\; D\,u(k)\,]\,. \tag{2.11}$$

Defining $A_K = \begin{bmatrix} A - KC \end{bmatrix}$, $B_K = \begin{bmatrix} B - KD & K \end{bmatrix}$ and $z(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$, one can finally write the system:

$$\begin{aligned} \tilde{x}(k+1) &= A_K\,\tilde{x}(k) \;+\; B_K\,z(k) \\ y(k) &= C\,\tilde{x}(k) \;+\; D\,u(k) \;+\; e(k). \end{aligned} \tag{2.12}$$

This is known as the *predictor form* [5] of system (2.7).
Defining p as the *past horizon*, the state equation of (2.12) can be forward propagated from time $k-p$ up to time k:

$$\tilde{x}(k) = A_K^p\,\tilde{x}(k-p) \;+\; \begin{bmatrix} A_K^{p-1}\,B_K & A_K^{p-2}\,B_K & \dots & B_K \end{bmatrix} \begin{bmatrix} z(k-p) \\ z(k-p+1) \\ \vdots \\ z(k-1) \end{bmatrix}. \tag{2.13}$$

Given the asymptotic stability of the predictor and assuming a large enough p, then $A_K^p \tilde{x}(k-p) \approx 0$ and the free response term becomes negligible [5]. Equation (2.13) can be rewritten:

$$\tilde{x}(k) \approx L_p\,z_p(k), \tag{2.14}$$

where $L_p = \begin{bmatrix} A_K^{p-1}B_K & A_K^{p-2}B_K \dots B_K \end{bmatrix}$ is the *extended controllability matrix of the predictor* and $z_p(k) = \begin{bmatrix} z^T(k-p) & z^T(k-p+1) \dots z^T(k-1) \end{bmatrix}^T$ is the *past data vector*.
Equation (2.14) can be substituted into the output equation of (2.12), resulting in:

$$y(k) \approx C\,L_p\,z_p(k) \;+\; D\,u(k) \;+\; e(k). \tag{2.15}$$

The same thing can be done on the complete dataset, which yields the matrix equations:

$$X \approx L_p\,Z_p \tag{2.16}$$

$$Y \;\approx\; C\,L_p\,Z_p \;+\; D\,U \;+\; E, \tag{2.17}$$

where

$$X = [x(k)\ x(k+1)\ \dots\ x(k+N-p)]\,, \tag{2.18a}$$
$$Z_p = [z_p(k)\ z_p(k+1)\ \dots\ z_p(k+N-p)]\,, \tag{2.18b}$$
$$Y = [y(k)\ y(k+1)\ \dots\ y(k+N-p)]\,, \tag{2.18c}$$
$$U = [u(k)\ u(k+1)\ \dots\ u(k+N-p)]\,, \tag{2.18d}$$
$$E = [e(k)\ e(k+1)\ \dots\ e(k+N-p)]\,. \tag{2.18e}$$

Matrices CL_p and D can be estimated by solving the least-squares problem:

$$\min_{CL_p, D} \| Y \ - \ CL_p Z_p \ - \ DU \|_F^2 \tag{2.19}$$

The least-squares problem above shows the basic idea behind the estimation of the predictor Markov parameters from input-output data. In the case of the PBSID$_{opt}$ algorithm, this is done by estimating a high-order VARX model, whose coefficients are then interpreted as the Markov parameters of the predictor and rearranged to construct the block matrix $\widehat{\Gamma_f L_p}$ [7].
To reveal the system's order, the algorithm introduces the *future horizon* f and constructs additional shifted regressions for future outputs. In this way, the following block matrix is obtained:

$$\widehat{\Gamma_f L_p} \ = \ \begin{bmatrix} \widehat{CL_p} \\ \widehat{CA_K L_p} \\ \vdots \\ \widehat{CA_K^{f-1} L_p} \end{bmatrix}, \tag{2.20}$$

where Γ_f is the *extended observability matrix of the predictor*:

$$\Gamma_f \ = \ \begin{bmatrix} C \\ CA_K \\ \vdots \\ CA_K^{f-1} \end{bmatrix}, \tag{2.21}$$

Matrix (2.20) is defined in [29] as a *subspace revealing matrix*; the first block row is $\widehat{CL_p}$, whereas each following block row is constructed by shifting the previous one by an increasing order $p, p+1, \dots$ and padding it with zeroes [5]. By multiplying both sides of equation (2.16) for Γ_f, one obtains:

$$\Gamma_f X \ \approx \ \widehat{\Gamma_f L_p} Z_p. \tag{2.22}$$

Performing a SVD on the right-hand side:

$$\widehat{\Gamma_f L_p Z_p} \;=\; \begin{bmatrix}\mathcal{U}_n & \mathcal{U}_\perp\end{bmatrix} \begin{bmatrix}\Sigma_n & 0\\ 0 & \Sigma_\perp\end{bmatrix} \begin{bmatrix}\mathcal{V}_n\\ \mathcal{V}_\perp\end{bmatrix}. \tag{2.23}$$

$\mathcal{U}$ is the matrix of *left singular vectors*, Σ is a diagonal matrix containing the *singular values* and $\mathcal{V}$ is the matrix of *right singular vectors*. Supposing that the correct model order n is selected (e.g., from an evident gap between the n dominant singular values and the smaller ones), the SVD is approximated as:

$$\widehat{\Gamma_f L_p Z_p} \;\approx\; \mathcal{U}_n \, \Sigma_n \, \mathcal{V}_n \tag{2.24}$$

where Σ_n is a diagonal matrix containing only the larger n singular values and $\mathcal{V}_n$ is the corresponding row space. A possible factorization is

$$\hat{\Gamma}_f \;=\; \mathcal{U}_n \, \Sigma_n^{1/2}, \qquad\qquad \hat{X} \;=\; \Sigma_n^{1/2} \, \mathcal{V}_n \tag{2.25}$$

After obtaining the state sequence, the system's matrices can be recovered through least-squares problems. Defining

$$\hat{X}_- = [\hat{x}(k) \quad \hat{x}(k+1) \;\ldots\; \hat{x}(k+N-1)] \qquad \hat{X}_+ = [\hat{x}(k+1) \quad \hat{x}(k+2) \;\ldots\; \hat{x}(k+N)]\,, \tag{2.26}$$

the output and feedthrough matrices can be estimated from the output equation:

$$\min_{C,D} \left\| Y - C\hat{X}_- - DU \right\|_F^2. \tag{2.27}$$

The innovation sequence is estimated as :

$$\hat{E} \;=\; Y \;-\; \hat{C}\hat{X}_- \;-\; \hat{D}U. \tag{2.28}$$

Finally, A, B and K are estimated:

$$\min_{A,B,K} \left\| \hat{X}_+ - A\hat{X}_- - BU - K\hat{E} \right\|_F^2 \tag{2.29}$$

2.1.2. PBSID toolbox and implementation

The PBSID$_{opt}$ algorithm is available in MATLAB through the *PBSID Toolbox*, developed by researchers of the Data-Driven Control group at Technische Universiteit Delft, which provides a set of routines for predictor-based subspace identification of LTI, LPV, Hammerstein and Wiener systems, both in open-loop and closed-loop configurations [30].
The toolbox provides the function `dordvarx`, which estimates the high-order VARX model from measured data and performs the SVD. The output of this step includes the state X, as well as the singular values S. The system's order n can be chosen by plotting said singular values and looking for a gap between them. The estimated state X and the order n are the input of function `dmodx`, which outputs the reduced-order state sequence x. The toolbox allows to impose constraints on the model, such as not considering the presence of direct feedthrough in the model (i.e. imposing $D = 0$), by choosing whether to use the function `dx2abck` or `dx2abcdk`, both of which output the system's matrices referred to equation (2.9). Other useful features implemented in the toolbox are the function `sigscale`, which performs signal scaling with the Root-Mean-Square power not to weight least squares problem when the system has multiple outputs, and the possibility to include options for improving the numerical robustness of the VARX model estimation. *Tikhonov regularization* can be used in place of the ordinary least-squares estimate of the predictor coefficients, and the value of its regularization parameter can be automatically determined by the *Generalized Cross Validation* criterion (more details can be found in [31, 32]).
The procedure is summarized in Algorithm 2.1:

Algorithm 2.1 PBSID$_{opt}$

1: **Input:** Input signal u, output signal y, past window p, future window f
2: **Output:** Estimated system matrices $\hat{A}, \hat{B}, \hat{K}, \hat{C}$.
3: Scale the signals: $[u_n, D_u, y_n, D_y] \leftarrow$ `sigscale`(u, y)
4: Evaluate order of the system:
5: [S, X, VARX, U, Zps] $\leftarrow$ `dordvarx`$(u_n, y_n, f, p,$ `'tikh'`, `'gcv'`$)$
6: Plot singular values: `figure, semilogy`$(S, '*')$
7: Choose model order: n
8: Compute the state sequence: $x \leftarrow$ `dmodx`(X, n)
9: Estimate system matrices: $[\hat{A}_n, \hat{B}_n, \hat{K}_n, \hat{C}_n] \leftarrow$ `dx2abck`(x, u_n, y_n, f, p)
10: Convert the matrices to physical scale:
11: $\hat{A} = \hat{A}_n,\ \hat{B} = \hat{B}_n/D_u,\ \hat{K} = \hat{K}_n/D_u,\ \hat{C} = D_y\hat{C}_n$
12: **Return:** $\hat{A}, \hat{B}, \hat{K}, \hat{C}$.

2.2. Structured model matching

The identification procedure described in the previous section provides an unstructured state space model, i.e. a model whose input-output behavior is estimated from data without imposing a physical parametrization on the matrices. This is one of the main advantages of subspace identification methods, because the state space basis is determined by the algorithm itself and no explicit parametrization of the matrices is required [27]. The consistency of the method up to a similarity transformation T, however, also represents a limitation when one wants to recover a physically-based dynamics model or exploit a prior knowledge of the system's dynamic characteristics. To overcome this limitation, structured model matching is introduced as a way to relate the identified unstructured model to a physically motivated model parametrization.

Let $\hat{G}_{ns}(s)$ denote the unstructured model estimated from input-output data, and $G_s(s,\vartheta)$ a structured model whose matrices depend on a set of unknown physical parameters $\vartheta \in \mathbb{R}^{n_\vartheta}$. The model matching problem consists in finding a value of ϑ such that the difference between the two input-output operators is small according to a chosen norm or error criterion:

$$\vartheta^\star = \arg\min_{\vartheta\in\Theta} J\left(\hat{G}_{ns}(s), G_s(s,\vartheta)\right) \tag{2.30}$$

where Θ collects possible bounds or constraints on the physical parameters. In this sense, the procedure is a grey-box identification step: the experimental information is contained in $\hat{G}_{ns}(s)$, while the model class is constrained by the analytical structure of $G_s(s,\vartheta)$ [33]. Classical structured state-space identification approaches may explicitly include the similarity transformation T as an additional unknown, but this leads to bilinear and generally non-convex optimization problems [34]; furthermore, it is not suitable when the orders of the two models are different, which is a typical situation as the physical model provides an approximated description of the true system dynamics.

In [35], the authors suggest modifying the objective function in such a way that it penalizes directly a measure of the deviation between the models. Possible choices would be the $\mathcal{H}_2$-norm or the $\mathcal{H}_\infty$-norm of the difference between the transfer functions of the two systems.

2.2.1. $\mathcal{H}_\infty$-norm minimization

The methodology presented in [15] reformulates the problem (2.30) as the minimization of the $\mathcal{H}_\infty$-norm of the difference between the two models:

$$\vartheta^\star = \arg\min_{\vartheta\in\Theta} \left\| \hat{G}_{ns}(s) - G_s(s,\vartheta) \right\|_\infty . \tag{2.31}$$

The problem in equation (2.31) is still a nonlinear, non-convex, non-smooth optimization, however T does not need to be constructed explicitly, so the optimization variables are only the physical parameters in ϑ. The problem in (2.31) can be interpreted as a special case of fixed-structure $\mathcal{H}_\infty$ tuning. The *standard form* for the synthesis, depicted in Figure 2.1, consists of a fixed LTI model $P(s)$ and a structured tunable controller $K(s,\vartheta)$. The problem can be formulated as follows:

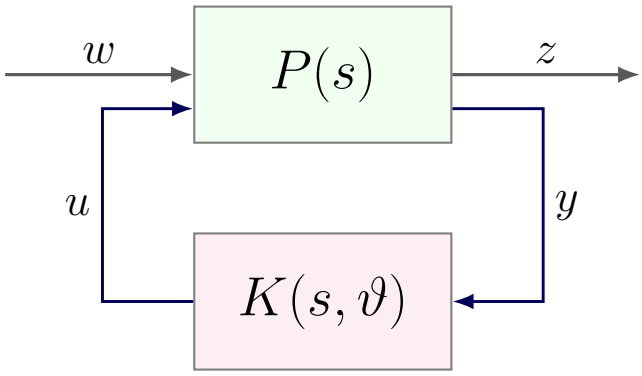


Figure 2.1: Standard form for structured $\mathcal{H}_\infty$ synthesis.

$$\begin{aligned} &\min_{\vartheta} \max_{\omega\in[0,\infty)} \bar{\sigma}\left(T_{w\to z}\left(P\left(j\omega\right),\vartheta\right)\right) \\ &\text{subject to: } K \text{ stabilizes } P \text{ internally.} \end{aligned} \tag{2.32}$$

$T_{w\to z}\left(P,K\right)$ is the closed-loop transfer function, while $\bar{\sigma}(\cdot)$ is the maximum singular value: in the SISO case it corresponds to the peak gain over frequency, whereas in the MIMO case it represents the peak induced 2-norm of the frequency response. The optimization problem converges to a local solution, given a suitable initial guess [36].
The model matching problem in (2.31) can be rearranged into the standard form of Figure 2.1, where $G_s(s,\vartheta)$ plays the role of the tunable controller $K(s,\vartheta)$, as seen in Figure 2.2:

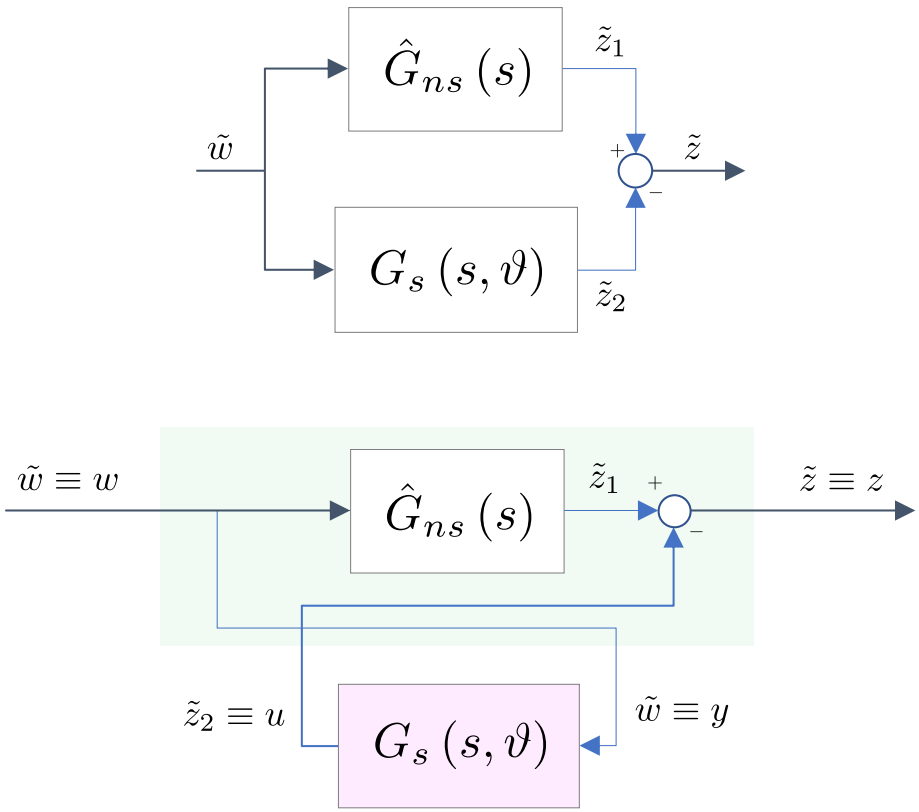


Figure 2.2: Model matching structure and equivalent standard form.

Formulating the problem in this way allows the use of tools developed for $\mathcal{H}_\infty$ controller synthesis embedded in the *Robust Control Toolbox* of MATLAB [37], with some important specifications: the removal of the stability constraint, which is meaningless for model matching, and the imposition of a focus bandwidth for the optimization. Both of these options are available through the function `systune`.

2.2.2. Frequency-domain Output-Error minimization

Problem (2.30) can also be formulated within the Output-Error (OE) minimization framework. In the standard time-domain OE formulation, the measured output is:

$$y_m(k) = y\left(k, \vartheta^\star\right) + v(k), \tag{2.33}$$

where $v(k)$ is measurement noise and $y\left(k, \vartheta^\star\right)$ denotes the noise-free output generated by the assumed model structure for the unknown true parameter vector $\vartheta^\star$. Under Gaussian white measurement noise assumptions, maximizing the likelihood of the measured data is

equivalent to minimizing the sum of squared output errors [33]:

$$J(\vartheta) = \frac{1}{2\sigma^2} \sum_{k=1}^{N} \|y_m(k) - y(k, \vartheta)\|_2^2 , \tag{2.34}$$

where the scalar variance σ^2, for multiple-output systems, is replaced by a weighting matrix related to the output-noise covariance, leading to a weighted quadratic criterion. The OE problem can also be formulated in frequency domain [21, 23, 38, 39]: given a frequency grid $\Omega = \{\omega_1, \ldots, \omega_{N_\omega}\}$, samples of the unstructured model's frequency response function (FRF) are obtained:

$$\hat{G}_{ns}(j\omega_k), \qquad k = 1, \ldots, N_\omega. \tag{2.35}$$

The cost function then becomes:

$$J_f(\vartheta) = \sum_{k=1}^{N} \left\| W(\omega_k) \left[\hat{G}_{ns}(j\omega_k) - G_s(j\omega_k, \vartheta) \right] \right\|_F^2 , \tag{2.36}$$

where $W(\omega_k)$ is a frequency dependent weight. Problem (2.30) becomes:

$$\vartheta^\star = \arg\min_{\vartheta \in \Theta} J_f(\vartheta) . \tag{2.37}$$

The problem can be solved using the tools present in the *System Identification Toolbox* of MATLAB. After converting the sampled values of $\hat{G}_{ns}(s)$ into an `idfrd` object, the structured model is implemented by means of a user-defined ordinary differential equation (ODE) function, returning the state-space matrices [40]:

$$\begin{aligned} \dot{x}(t) &= A(\vartheta)\, x(t) \;+\; B(\vartheta)\, u(t) \\ y(t) &= C(\vartheta)\, x(t) \;+\; D(\vartheta)\, u(t), \end{aligned} \tag{2.38}$$

as an `idgrey` object. Note from (2.38) that the system can be defined either in continuous or discrete-time. Finally, a local optimization from initial guess ϑ_0 is performed, via an iterative minimization of the loss function $J_f(\vartheta)$ using numerical search methods [41]: more on the available search methods can be found at [42].

2.3. Prefiltering for type correction

The type of a SISO transfer function is the number of poles or zeros located at +1 on the unit circle for discrete-time models, or at the origin for continuous-time models. Subspace identification methods such as PBSID$_{opt}$ always provide an unstructured model in discrete-time whose transfer functions do not have zeros or poles on the frontier of the unit circle. Structured continuous-time models, on the other hand, could have transfer functions with type $\neq 0$, as in the case of transfer function (1.35) of the quadrotor UAV presented in Chapter 1.6.
To prevent difficulties in the model matching algorithm's convergence due to having to deal with transfer functions of different type, a special prefiltering for type correction can be applied on the identification dataset [43].
Given a system with $j = 1, \ldots, m$ inputs and $i = 1, \ldots, l$ outputs, input data should be filtered independently on each j-th channel, while output data should be filtered independently on each i-th channel [12]. In Chapter 1.6, the system obtained from physical modeling has been decoupled into SIMO models, therefore $m = 1$; given a filter design $F_u(s, \omega_c)$ for the input in continuous-time, the filter for the output can be derived as:

$$F_{y_i}(s, \omega_c, g_i, \omega_F) \coloneqq F_u(s, \omega_c) \left(\frac{s}{1 + s/\omega_F} \right)^{g_i}, \tag{2.39}$$

where g_i is the type of the transfer function from input u to output y_i, frequency ω_c is the design parameter of the input filter and frequency ω_F is the design parameter of the output filter. In the framework of this thesis, the input filter has been defined as a first order low-pass filter with cut-off frequency ω_c:

$$F_u(s, \omega_c) = \left(\frac{1}{1 + s/\omega_c} \right), \tag{2.40}$$

and the outputs filters are:

$$\begin{aligned} F_p(s, \omega_c, \omega_F) &= \frac{1 + s/\omega_F}{s(1 + s/\omega_c)}, \\ F_{a_y}(s, \omega_c) &= F_u(s, \omega_c). \end{aligned} \tag{2.41}$$

The filtered continuous-time system has transfer function:

$$G_{F_i}(s) = G_i(s) \left(\frac{s}{1 + s/\omega_F} \right)^{g_i}. \tag{2.42}$$

To recover the original system in discrete-time, the bilinear transform (*Tustin's method* [44]) is applied:

$$\hat{G}_i(z) = \left\{ \left(\frac{1 + s/\omega_F}{s} \right)^{g_i} \right\}_{s=\frac{2}{T}\frac{z-1}{z+1}} \hat{G}_{F_i}(z), \tag{2.43}$$

where T is the sampling period.

3 | Experimental activities

This chapter describes the laboratory equipment and the experimental activities conducted to collect the flight-dynamics data used by the PBSID$_{opt}$ algorithm. The experimental procedure was originally defined in previous work [13]; however, different combinations of setup parameters were tested in this thesis to improve the identification results and provide a more reliable starting point for the structured model-matching procedures of Chapter 5.

3.1. Aerospace Systems and Control Laboratory

The Flying Arena for Rotorcraft Technologies (Fly-ART) facility, part of the Aerospace Systems & Control Laboratory (ASCL) is an indoor environment used for experimental activities, including the testing of UAV platforms. With a flight volume of 12 m x 6 m x 4 m, it is equipped with a *Motion Capture System* (*Mocap*) for tracking the rigid body motion of the UAV, thus estimating position and attitude of the drone. It is connected to a *Ground Control System* (*GCS*) dedicated to sending and retrieving information from the quadcopter during test sessions. The communication protocol with the drone is based on the *Robot Operating System* (*ROS*).

3.1.1. WindShaper

Developed by *WindShape SA* [45] to replicate real weather and wind conditions for testing of flying vehicles in a laboratory setting [46–48], the WindShaper is a modular wind generator composed of fan modules that can be combined to create a wind wall of any size and shape. Each fan can be controlled independently, which allows for fine control of flow properties, and the low inertia of small fans enables fast transients. Methodologies for effective testing of UAVs using the WindShaper, as well as indications about wind speed and turbulence distribution, are provided in [49].

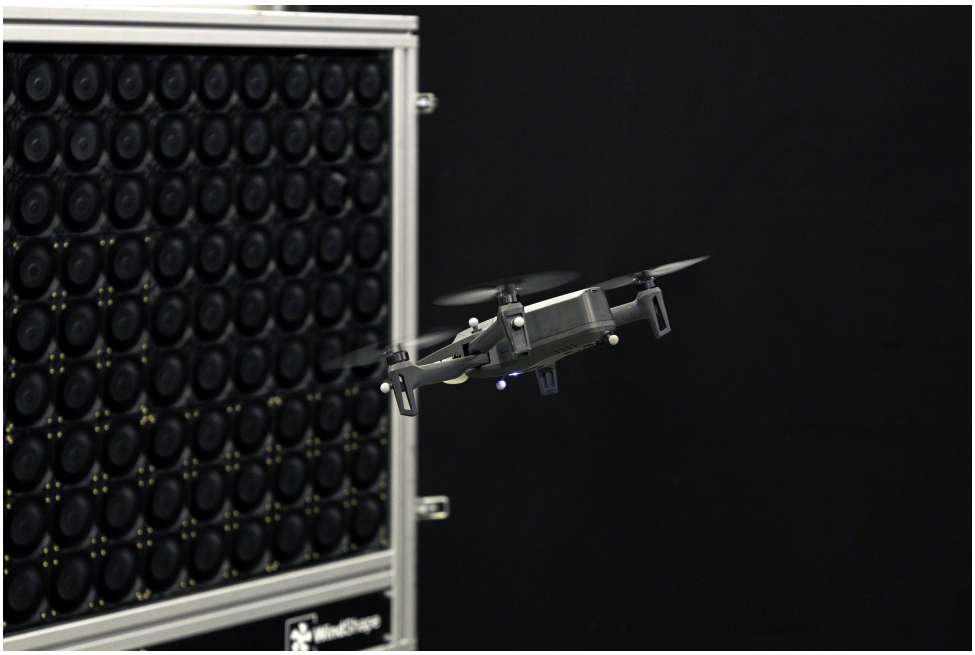


Figure 3.1: Drone in free flight in the WindShaper wind field, from [49].

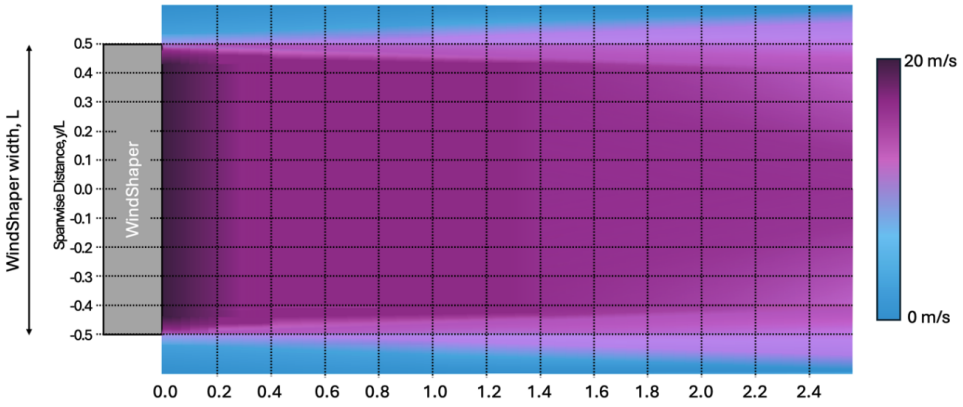


Figure 3.2: Flowfield distribution downstream of the WindShaper, from [49].

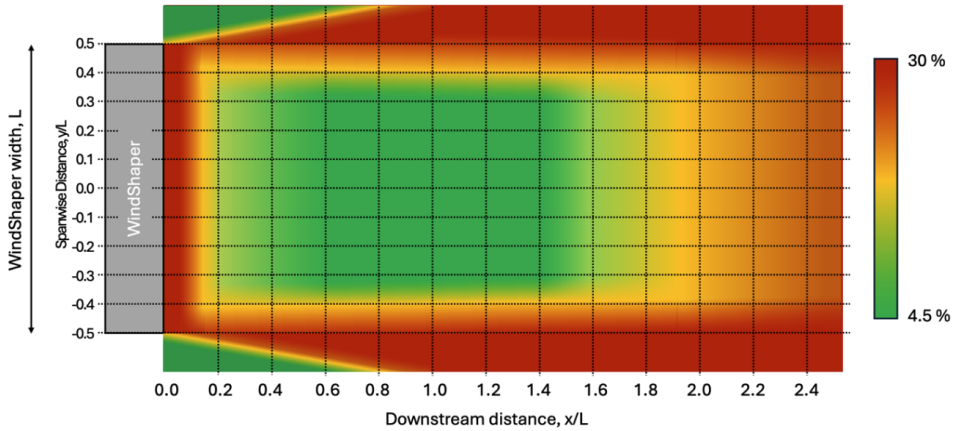


Figure 3.3: Turbulence distribution downstream of the WindShaper, from [49].

Given this information, it was decided to place the drone at a distance of 1.3 m from the WindShaper to ensure both the safety of operations and the accuracy of the experiment. The WindShaper used at ASCL is about 1.22 m wide, therefore the ratio between downstream distance and width is $x/L = 1.07$, meaning that the experiments were conducted within the efficiency zone defined in Figure 3.2 and 3.3.

3.1.2. ANT-X drone



Figure 3.4: ANT-X drone.

Figure 3.4 shows the drone used for the experiments of this thesis. This section describes the ANT-X drone lab environment. The drone is equipped with an *Inertial Measurement Unit* (*IMU*) with a three axis accelerometer and a three axis gyroscope. It is also comprised of a *Flight Control Unit* (*FCU*) which controls the motors' speed through the *Electronic Speed Controllers* (*ESCs*). The *FCU* firmware is the *PX4 autopilot*, which implements an *Extended Kalman Filter* that provides real-time state estimation using inertial and *Mocap* measurements available from the communication node between the PX4 firmware and the *GCS*, the *Companion computer*.

3.2. Previous works

Relevant studies performed at ASCL include *Multirotor UAV model identification under varying flight conditions* [13], which developed the experimental procedure described in 3.3, considering a SISO system, and obtained an estimation of the parameters of the lateral dynamics under varying flight conditions. The results obtained by the author are reported in Tables 3.1 and 3.2.

Power [%]	p	f	n	VAF [%]	FIT [%]
0	13	10	4	85.68	62.16
5	21	9	4	44.10	25.24
10	25	18	4	57.07	34.48
15	13	13	4	89.23	67.19
20	20	20	4	91.52	70.87
25	12	12	4	95.99	79.97
30	13	11	7	86.85	63.74
35	13	11	5	96.06	80.14
40	15	13	5	97.19	83.23

Table 3.1: Black-box cross-validation results for different WindShaper power levels and PBSID$_{opt}$ optimal *hyperparameters*, considering the SISO system with input δ_{lat} and output p [13].

Power [%]	Y_v [1/s]	Y_p [m/(s rad)]	L_v [rad/(s m)]	L_p [1/s]	Y_δ [m/s^2]	L_δ [rad/s^2]
0	-1.1581	-0.0956	7.6591	2.7895	19.3983	1566.7860
5	0.4393	2.7669	14.5670	11.0860	109.9722	2334.1708
10	-1.4386	3.5313	11.5528	9.8690	-374.4022	2810.0603
15	-6.9363	-1.9178	19.3186	4.5498	237.5458	851.6019
20	-10.1547	-2.1805	25.7752	3.5267	522.5630	296.5221
25	-9.0112	-3.4063	15.3272	1.1510	529.9263	500.5118
30	-4.7310	-5.5791	11.4384	-1.8756	706.4103	207.2811
35	-12.5013	-1.8954	37.8422	-0.3154	490.4982	172.5661
40	-16.9019	-1.3375	56.4942	-1.7863	572.9344	-246.1630

Table 3.2: Nominal structured parameter estimates for different WindShaper power levels, considering the SISO grey-box model with input δ_{lat} and output p [13].

3.3. Experimental procedure setup

The identification experiment is the same as the one performed in [13]. To perform said experiment, the following steps are carried out in the specified order to ensure operational safety:

- Take off at a distance of 1.3 m from the WindShaper and climb to an altitude of 1.3 m, then stabilize the position.
- Activate the WindShaper by setting the desired power percentage.
- Once the transient has decayed, the position and attitude controllers restore the new trim conditions.
- Inject the automated frequency-sweep disturbance signal.
- After the signal has ended, land the drone safely with the WindShaper still active.
- Once the drone has landed, power off the WindShaper.

In order to gather input-output data at varying flight conditions, the experimental procedure was repeated for WindShaper power percentage ranging from 0% to 40%, with intervals of 5%.
In [50], the effect of wind stress on quadrotor dynamics is investigated by analyzing the coherence between the estimated wind-induced disturbances and the measured wind velocity. The authors found that the dominant influence appears in the dynamics aligned with the wind direction. For this reason, the present work restricts the analysis to the effect of wind on the lateral dynamics, with the experimental setup arranged so that the roll motion is excited by a flow direction perpendicular to the roll axis.

One of the critical aspects of the work done in [13] was the low accuracy obtained by the PBSID$_{opt}$ algorithm for models at low WindShaper power levels, namely at 5% and 10%. For this reason, although the experimental procedure remains the same in terms of setup and steps followed, different experimental parameters were used. In particular, the distance from the WindShaper was decreased from 1.8 m to 1.3 m, in order to better position the UAV in the efficiency zone defined by Figures 3.2 and 3.3. The length of the experiment was increased from 70 s to 120 s, and different excitation signals were used to check if they could provide higher accuracy in terms of VAF index (which will be defined in Section 4.2) for the black-box identification algorithm.

The excitation signal δ_{sweep} is a normalized moment disturbance which, after being summed to the control input, results in δ_{lat}. δ_{sweep} can be either a sine sweep with a 10% amplitude,

a duration of 120 s and a frequency ranging from 0.3 to 8 rad/s, or a pseudo-random binary signal (PRBS) of the same duration and with bandwidth 0.315–8 rad/s. If the identified model exhibits poles that fall outside the bandwidth of the identification signal, the experiment is repeated with a bandwidth going up to 16 rad/s, in order to properly identify and validate the model.

The same experimental procedure is repeated to obtain a cross-validation dataset, which is used to find the optimal combination of *hyperparameters* for the PBSID$_{opt}$ algorithm, as described in Section 4.2. In the cross-validation experiment, a different signal is injected under the same flight conditions: in the case of the sine sweep, the signal has a frequency bandwidth of 0.3–6 rad/s (up to 15 rad/s for higher WindShaper powers) and a duration of 100 s, while for the PRBS cross-validation signal a different random sequence is used, with bandwidth 0.315–15 rad/s (16 rad/s in case of higher WindShaper powers) and the same duration as the identification dataset.
In light of the different experimental setup parameters, the PRBS signal was chosen because it achieved higher cross-validation performance.

3.4. Identification and validation datasets

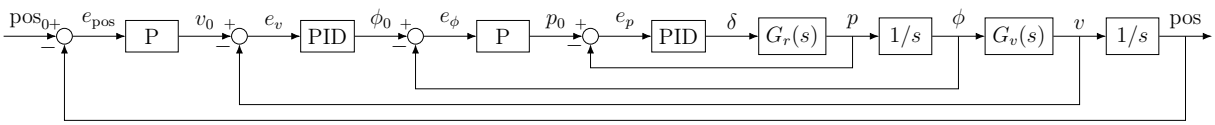


Figure 3.5: Cascade control structure.

The identification experiments are performed in closed loop due to the intrinsic instability of the quadrotor. The controller is made up of a P controller for the drone's position in cascade with a PID controller for the velocity, followed by a P controller for attitude in cascade with a PID controller for the drone's angular rate. The overall control structure is shown in Figure 3.5, where $G_r(s)$ is the roll rate dynamics transfer function while $G_v(s)$ is that of velocity dynamics.

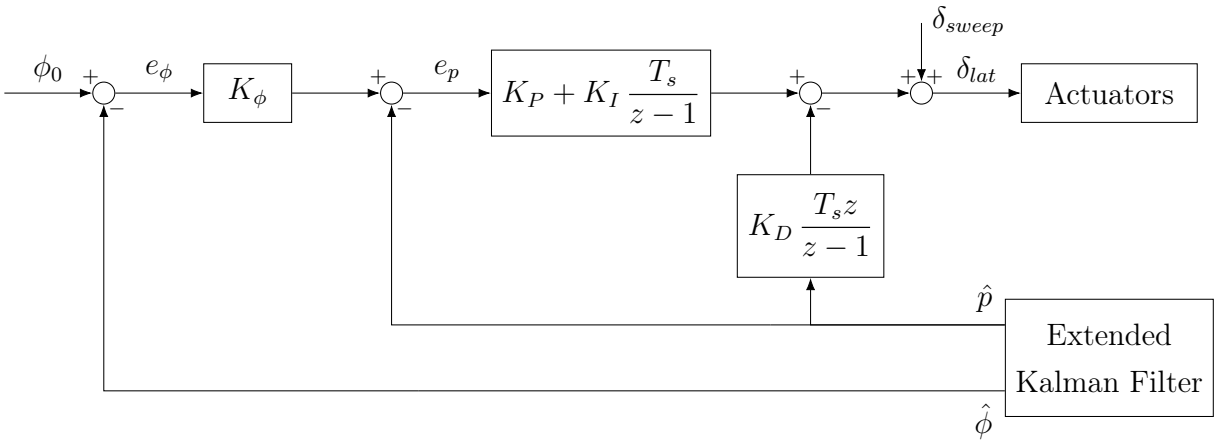


Figure 3.6: Lateral dynamics control system.

Details of the lateral dynamics control system can be seen in Figure 3.6: roll-angle $\hat{\phi}$ and roll-rate $\hat{p}$ measurement are estimated by an *Extended Kalman Filter* (EKF). The control action of the P/PID cascade controller is summed to the disturbance δ_{sweep} to obtain δ_{lat}. *Input moment* δ_{lat}, estimated *roll rate* p and measured *acceleration* a_y constitute the identification dataset for the PBSID$_{opt}$ algorithm. Figure 3.7 shows an example of such dataset, while Figure 3.8 shows the cross-validation signal for the same WindShaper power.

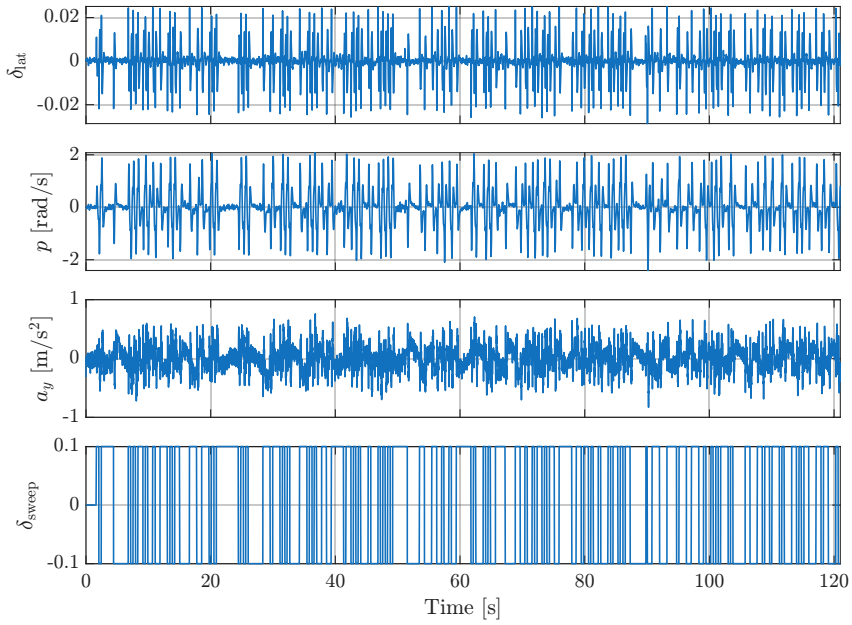


Figure 3.7: Dataset δ_{lat}, p, a_y and excitation signal δ_{sweep} used for the identification of lateral dynamics at 20% of WindShaper power.

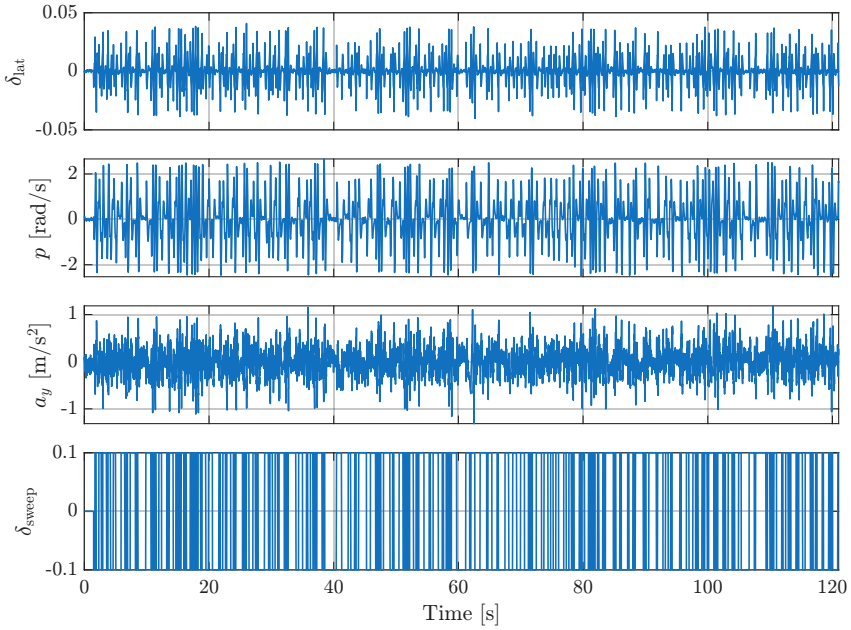


Figure 3.8: Dataset δ_{lat}, p, a_y and excitation signal δ_{sweep} used for the validation of lateral dynamics at 20% of WindShaper power.

4 | Validation procedure of unstructured models

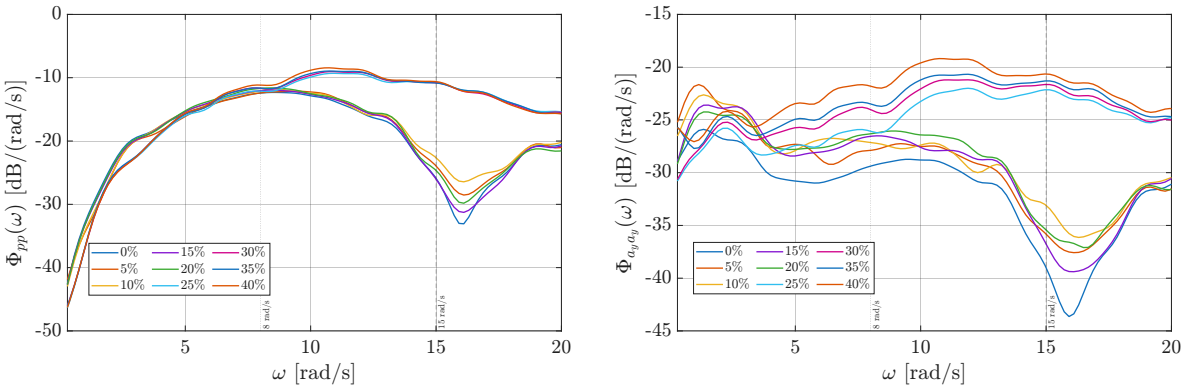
This chapter describes the procedure adopted to estimate and validate the unstructured models obtained with the PBSID$_{opt}$ algorithm. First, the experimental datasets are analyzed in the frequency domain to assess the level of excitation of the measured signals and the input-output correlation within the bandwidth of interest. Then, the black-box identification procedure is introduced, with particular attention to the selection of the main PBSID$_{opt}$ *hyperparameters*, namely the past window p, the future window f and the model order n. The optimal models are selected through a grid search based on the cross-validation performance on an independent dataset.

The quality of the resulting unstructured models is then evaluated by combining cross-validation on independent data with frequency-domain comparisons against non-parametric FRF estimates. These validation steps provide the basis for selecting the unstructured models that will be used in the structured model-matching procedures of Chapter 5.

4.1. Preliminary analysis of the experimental dataset

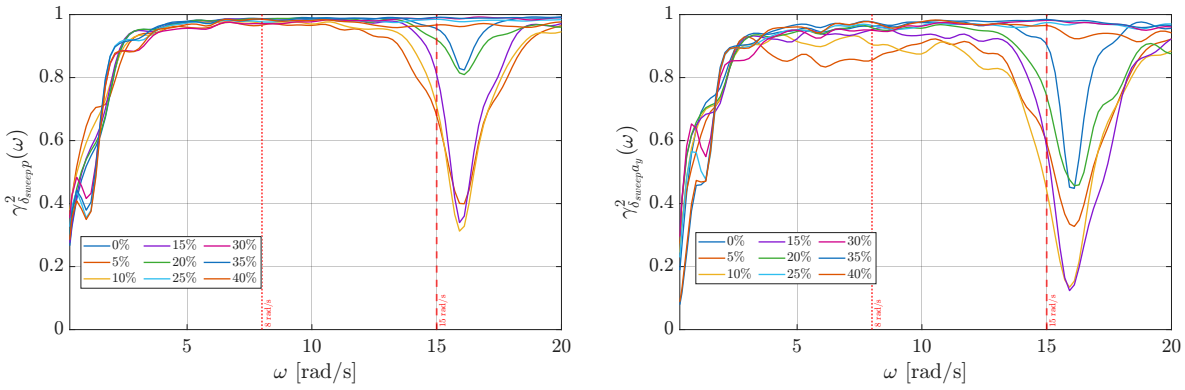
Before applying the black-box identification procedure, the experimental data are inspected in order to verify that they are suitable for the estimation of the lateral dynamics. The power spectral density (PSD) of the signal, often denoted with the letter $\Phi(\omega)$, is used to verify where the signal's energy is located in frequency. The PSD can be estimated using Welch's method [51].

Figure 4.1 shows the PSD for the two measured outputs p and a_y, obtained after the injection of the PRBS excitation signal δ_{sweep}: the energy of the excitation is mainly concentrated in the frequency interval targeted during the experiment, namely 0.315–8 rad/s for WindShaper powers below 20% and up to 16 rad/s for higher wind speeds, as explained in Section 3.3.
The magnitude-squared coherence is used to quantify how much of the output at a given



(a) Output p. (b) Output a_y.

Figure 4.1: Power spectral density of the outputs for varying WindShaper power.



(a) Output p. (b) Output a_y.

Figure 4.2: Input-output coherence between δ_{sweep} and y for varying WindShaper power.

frequency can be linearly associated with the input; indeed, it is defined as the ratio of the energy of the deterministic output response of the system (assuming that a linear relation exists with the input signal) with respect to the total energy of the output considering noise. Since the experiments are performed in closed-loop, the coherence is computed with respect to the externally injected PRBS excitation signal $r(t) = \delta_{sweep}$ rather than with respect to the commanded input δ_{lat}, which also contains the feedback contribution generated by the controller [12]. Figure 4.2 shows that both p and a_y exhibit satisfactory levels of coherence in the frequency bandwidth of the experiment: a common practice in flight-test system identification is to consider the bandwidth where $\gamma^2_{ry} > 0.6$ [21], which translates to considering the frequency range 1.5–8 rad/s.

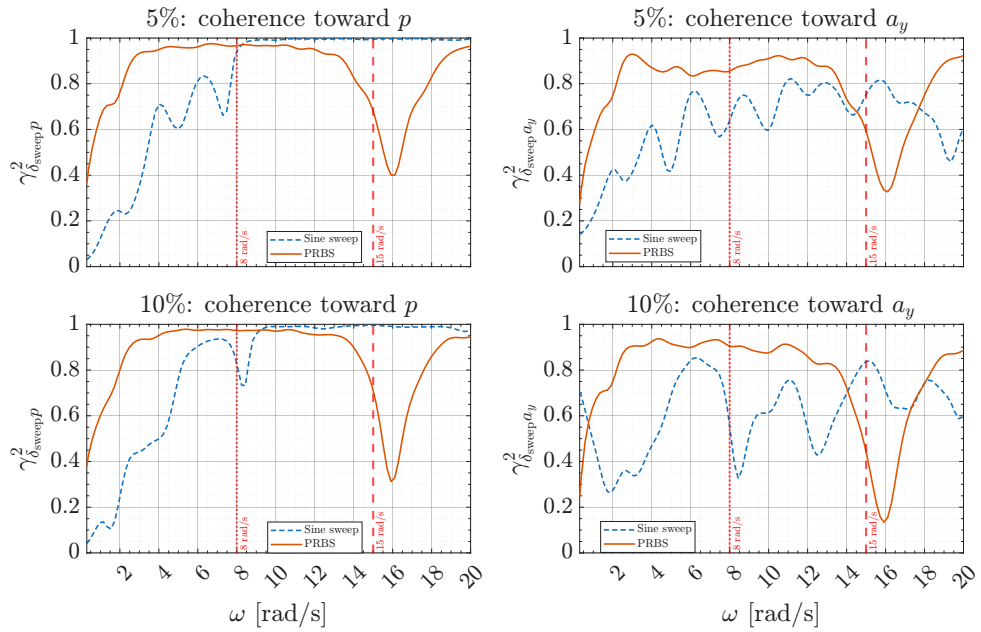
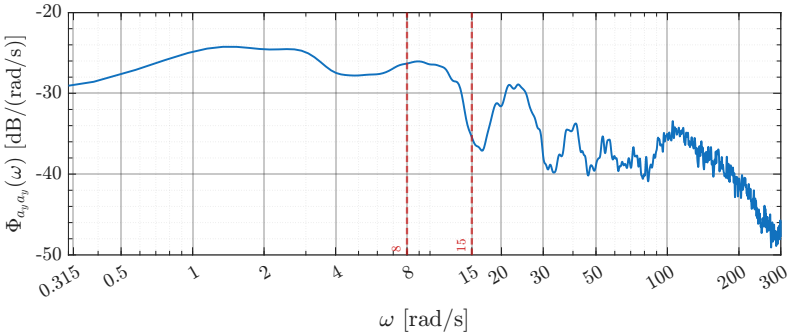


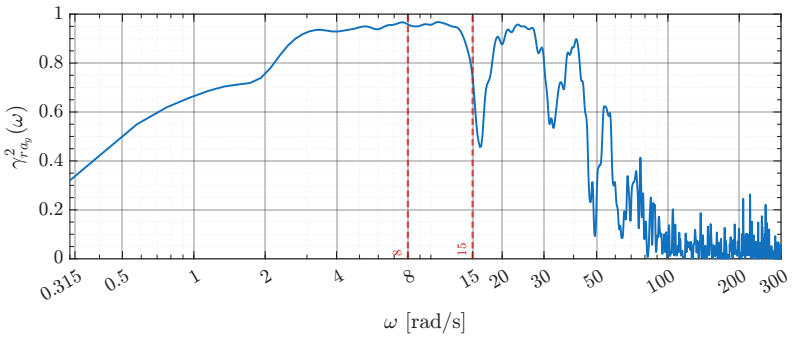
Figure 4.3: Comparison of input-output coherence between δ_{sweep} and y at 5% and 10% of the previous experiments [13] and the current ones.

Figure 4.3 shows the coherence estimate obtained in [13], using a sine sweep as excitation signal, compared with the one computed for the experiments described in Section 3.3, using a PRBS as excitation signal: in particular, the 5% and 10% WindShaper power levels are shown, since they correspond to the cases with the lowest coherence estimates and, consequently, the poorest cross-validation results. The new experiments show higher coherence over most of the frequency bandwidth of interest, supporting the choice of the PRBS as excitation signal for the present identification campaign.
Lastly, filtering is used to reduce components that are not informative of the rigid-body lateral dynamics, such as high-frequency sensor noise, aliasing remnants, or structural/actuator dynamics outside the identification bandwidth. This is particularly important for SIMO systems, since using a_y as additional output for the identification introduces significant additional noise, which has to be attenuated via filtering. Figure 4.4a shows that the signal has significant energy content beyond the frequency threshold used for the identification, however the coherence $\gamma^2_{ra_y}$ decreases significantly for $\omega > 50$ rad/s, as seen in Figure 4.4b; indeed, for values of ω above this threshold, the energy content is mostly uncorrelated with the injected excitation and is therefore not informative for the identification of the lateral input-output dynamics.

Then, a low-pass filter can be applied to all input and output signals, ensuring that the filter bandwidth is wider than the frequency range used for model validation and model matching: a conservative choice is $\omega_c = 100$ rad/s, which ensures that the filter does



(a) Power-Spectral-Density of a_y.



(b) Coherence from δ_{sweep} to a_y.

Figure 4.4: Power-Spectral-Density and coherence analysis for the lateral acceleration signal at 20% WindShaper power.

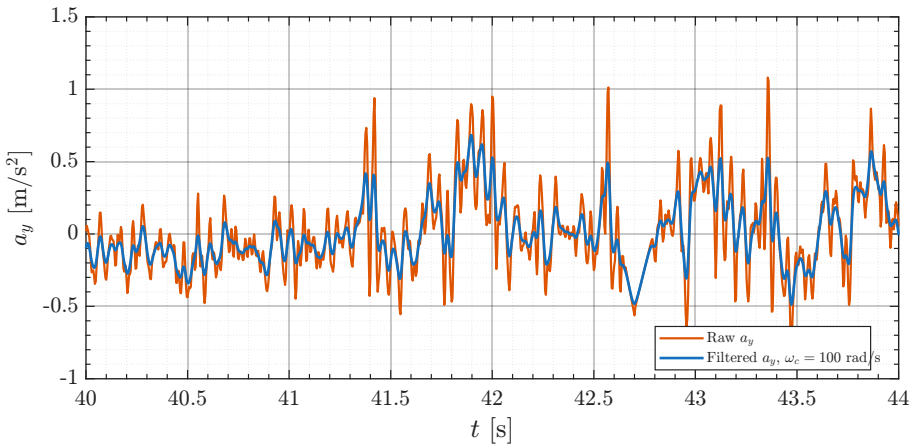


Figure 4.5: Raw and filtered a_y, with $\omega_c = 100$ rad/s.

not introduce significant attenuation in the dynamics relevant for identification, while still reducing high-frequency measurement noise and unmodelled components outside the validation bandwidth. Figure 4.5 shows the signal a_y before and after filtering.

4.2. PBSID$_{opt}$ black-box identification

The first step for the PBSID$_{opt}$ algorithm is the selection of the *hyperparameters* p (past window), f (future window) and n (model order), always ensuring that $p \geq f \geq n$. In order to select the optimal combination of *hyperparameters*, a grid search is carried out: a first criterion for the choice of n derives from the singular value decomposition, as stated in Equation (2.24), which shows a noticeable gap after the fifth singular value (Figure 4.6): therefore, $n = 5$ was selected as a lower bound for n. The upper bound for the model order is 10, since it has emerged that an increase in the model order can lead to *overfitting*, while for p and f an upper bound of 50 is chosen, since it was observed that increases beyond this range did not lead to significant improvements in model accuracy.
For each triplet (n, p, f), an unstructured model is estimated and its quality is evaluated by the Variance Accounted For (VAF) index, computed independently for each output channel:

$$\mathrm{VAF}(y_{val}) = \max\left(0,\ 1 - \frac{\mathrm{Var}(y_{val} - \hat{y})}{\mathrm{Var}(y_{val})}\right) \times 100. \tag{4.1}$$

The index is computed against the dedicated cross-validation dataset in which, as explained in Section 3.4, the values of input $u_{val} = \delta_{lat,val}$, and outputs $y_{val} = [p_{val}, a_{y,val}]^T$ are collected: the estimated output $\hat{y}$ is obtained by simulating the response of the identified model using $\delta_{lat,val}$ as input. It is worth noting that, to deal with the instability of the system, the simulation is performed in the frequency domain, as detailed in Sec-

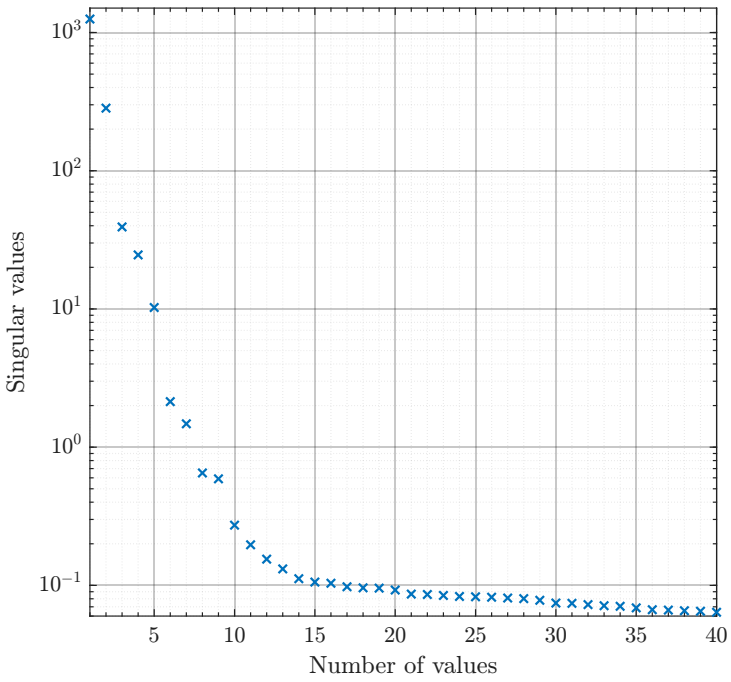


Figure 4.6: Singular value decomposition, $ABCK$ model, $p = f = 50$; a gap can be spotted after the first five singular values.

tion 4.3.1. Since it was observed that $\text{VAF}_{a_y} < \text{VAF}_p$, the adopted criterion for choosing the best model was the following: first, for every WindShaper power, a minimum acceptable value for VAF_p was chosen (e.g. $\text{VAF}_p > 90\%$). Among the models satisfying this constraint, the selected model was the one maximizing VAF_{a_y}, so as to favor models capable of reproducing also the noisier lateral-acceleration response.

4.3. Results and validation

This section presents the validation results of the unstructured models estimated by the PBSID_{opt} algorithm. For each flight condition, the combination of *hyperparameters* selected from the grid search is shown in Table 4.1.

Power [%]	n	p	f	VAF$_p$ [%]	VAF$_{a_y}$ [%]
0	7	17	12	97.01	49.93
5	5	8	8	70.66	51.27
10	7	9	8	93.43	45.26
15	7	11	8	98.75	61.92
20	7	12	11	98.40	67.89
25	7	20	20	99.49	72.37
30	7	13	13	99.35	72.75
35	7	10	8	99.06	60.87
40	7	20	13	99.25	49.24

Table 4.1: PBSID$_{opt}$ *hyperparameters* and cross-validation performance of the unstructured models.

The validation index is higher for p than it is for a_y, consistently with the results obtained in hover in [12]. This result is somewhat expected, given the fact that the measure of a_y, although filtered, still contains a non-negligible amount of noise. Comparing the results of Table 4.1 with the ones obtained in [13] shown in Table 3.1, the VAF is higher for each WindShaper power, thus corroborating the choice of a PRBS as identification signal; furthermore, there is a noticeable improvement in the models at 5% and 10%. The results obtained for 5% power show the lowest VAF on output p: this is likely due to the irregular flow produced by the WindShaper at lower power levels, which may have an effect more similar to an aerodynamic disturbance. In the following sections, only the validation plots for these selected models are reported.

4.3.1. Cross-validation on time histories

As stated in Section 4.2, the main criterion for the choice of the optimal *hyperparameters* is the VAF index: in order to obtain the estimated output vector $\hat{y}$, the identified model's response must be simulated using as input the measured $\delta_{lat,val}$ from the cross-validation dataset. Although the VAF is computed using data in the time domain, the open-loop instability of the quadrotor means that a direct time-domain simulation by numerical integration may be affected by divergence caused by numerical errors, residual noise, or small inconsistencies in the initial conditions. In [12], this is avoided by implementing a linearized version of the drone's control system and performing the simulation in closed-loop. In [13], on the other hand, a different approach is followed, as the model's response is obtained by performing the simulation in the frequency domain: this only requires

algebraic operations on the finite Fourier spectra of the signals, and the simulated response is reconstructed as a finite harmonic superposition [20].

The procedure can be summarized as follows. Let $u_{val}(k) = \delta_{lat,val}(k)$ be the measured validation input, sampled with sampling time T_s and composed of N samples. First, its Discrete Fourier Transform (DFT) is computed:

$$U_{val}(\omega_j) = \mathrm{DFT}\left\{u_{val}(k)\right\}, \qquad \omega_j = \frac{2\pi j}{NT_s}, \quad j = 0, \ldots, N-1. \tag{4.2}$$

Then, the frequency response of the identified discrete-time model is evaluated on the same frequency grid ω_j, or equivalently on the corresponding points of the unit circle

$$z_j = e^{\mathrm{i}\omega_j T_s}. \tag{4.3}$$

For each output channel, the simulated output spectrum is obtained by multiplying the model frequency response by the input spectrum:

$$\hat{Y}_i(\omega_j) = \hat{G}_i(z_j)\, U_{val}(\omega_j), \qquad i = 1, \ldots, l. \tag{4.4}$$

Finally, the simulated time history is recovered by applying the inverse Fourier transform:

$$\hat{y}_i(k) = \mathrm{IDFT}\left\{\hat{Y}_i(\omega_j)\right\}. \tag{4.5}$$

This procedure provides the simulated output time histories $\hat{p}$ and $\hat{a}_y$, which can then be directly compared with the measured validation outputs and used to compute the VAF index. Figures 4.7–4.15 show a portion of the time histories of the compared signals. Possible local discrepancies can be attributed to measurement noise, unmodelled nonlinear effects, closed-loop interactions, and the finite-length nature of the frequency-domain simulation.

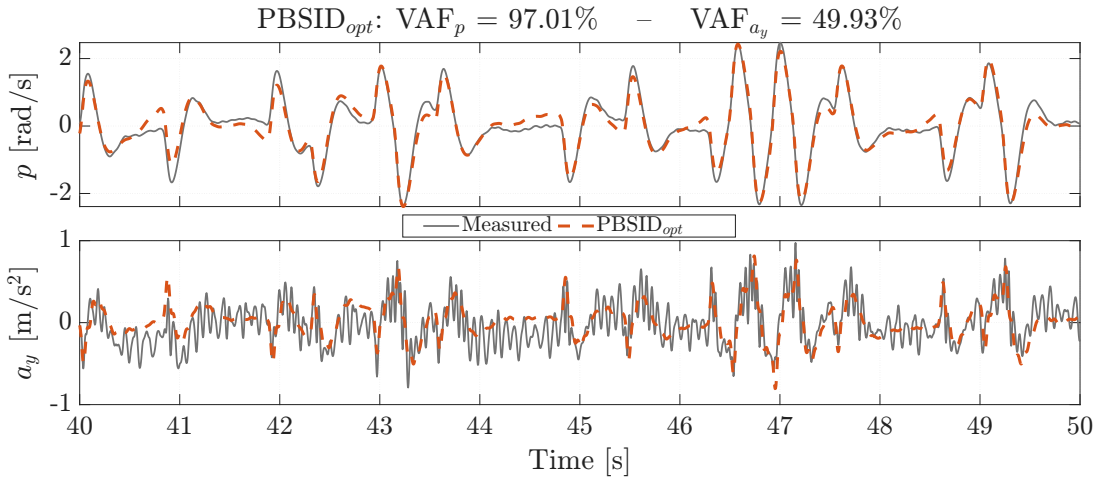


Figure 4.7: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 0% WindShaper power.

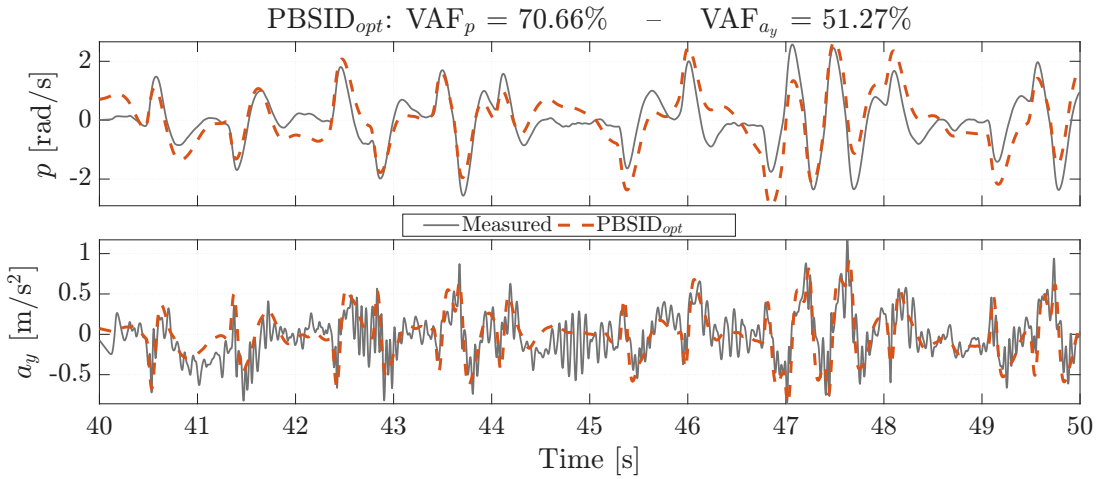


Figure 4.8: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 5% WindShaper power.

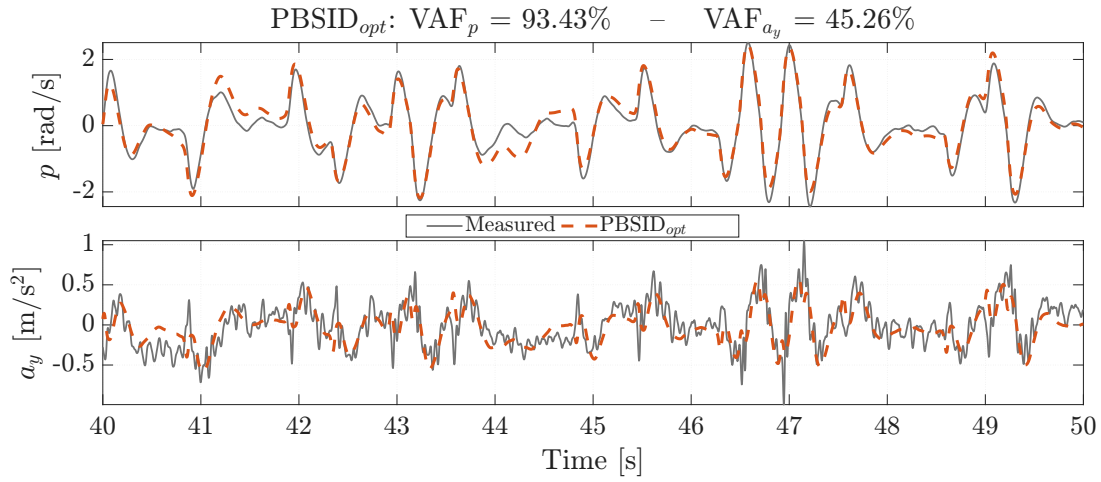


Figure 4.9: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 10% WindShaper power.

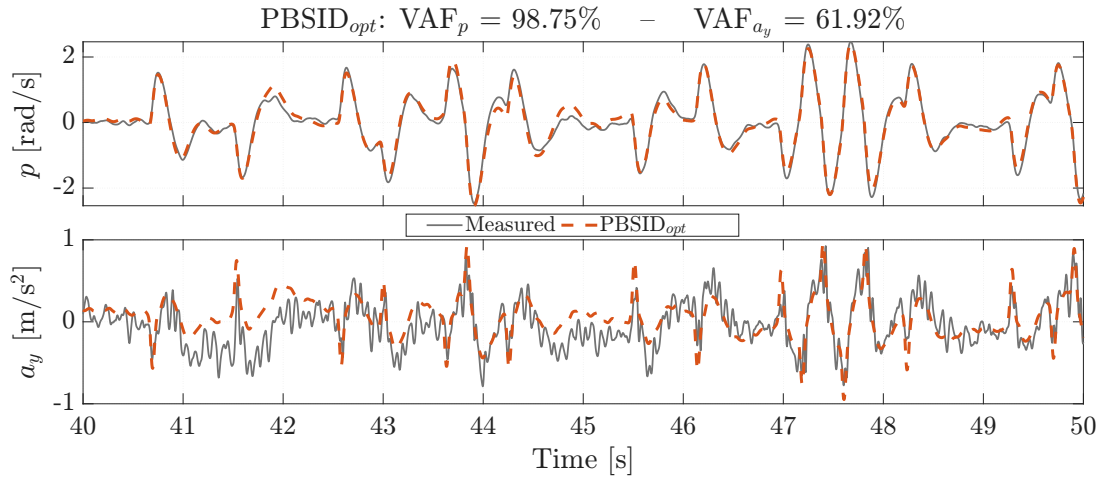


Figure 4.10: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 15% WindShaper power.

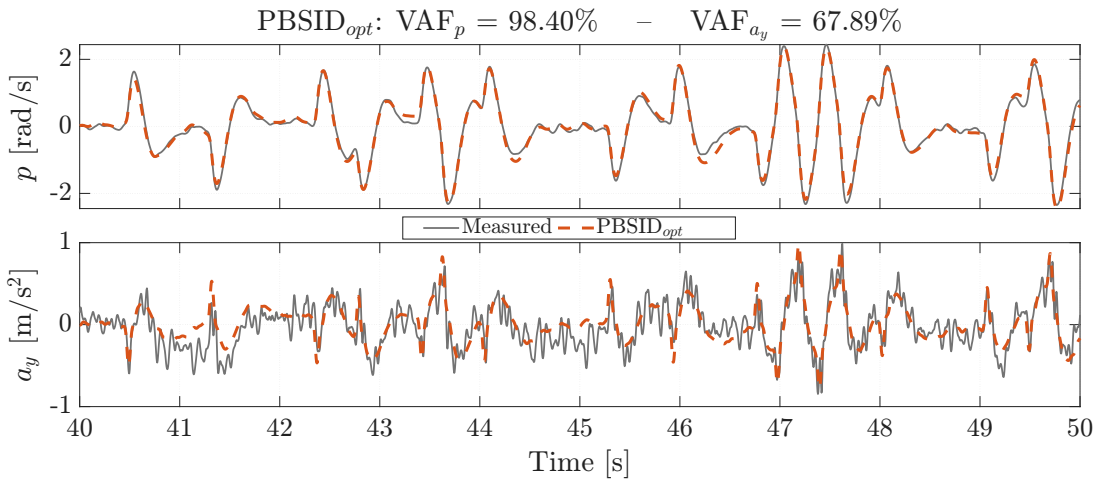


Figure 4.11: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 20% WindShaper power.

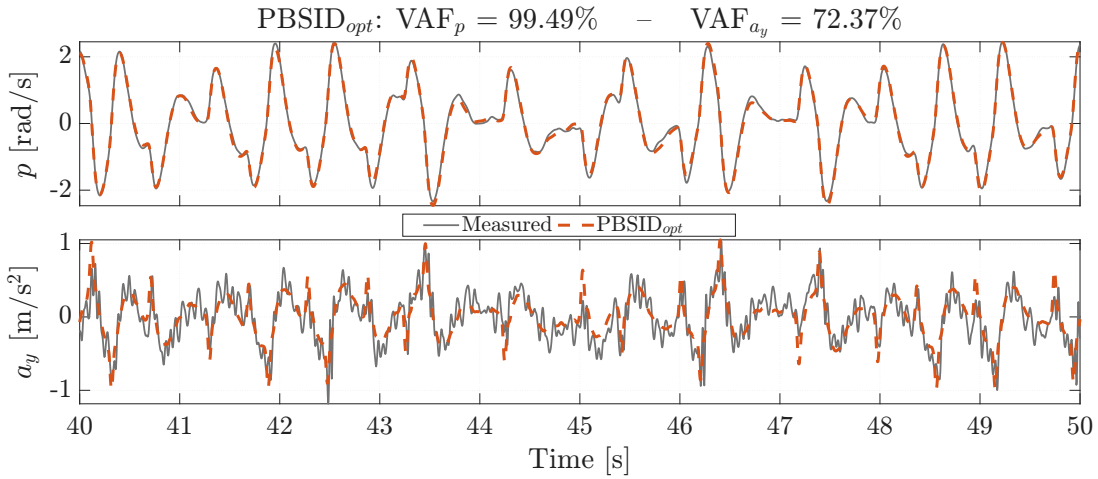


Figure 4.12: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 25% WindShaper power.

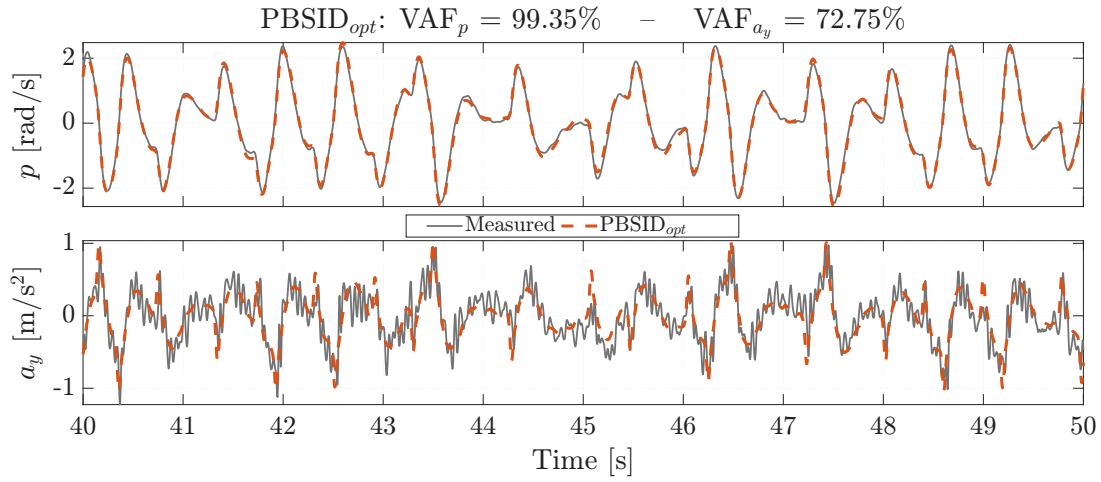


Figure 4.13: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 30% WindShaper power.

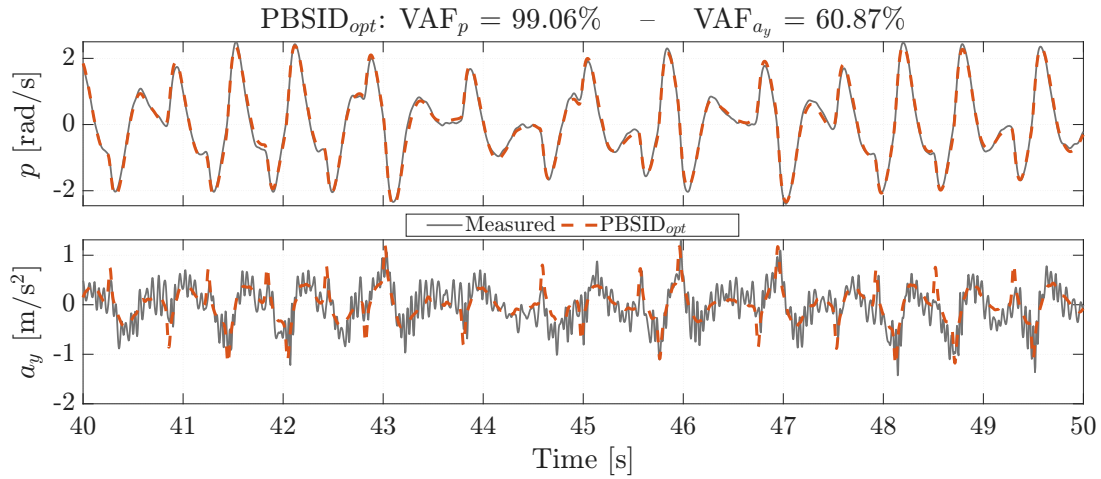


Figure 4.14: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 35% WindShaper power.

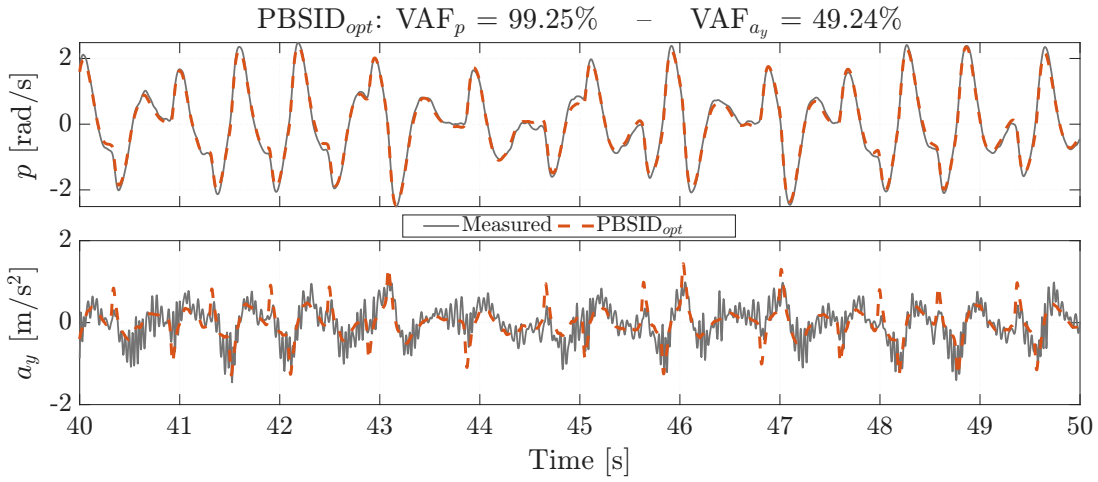


Figure 4.15: Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 40% WindShaper power.

4.3.2. Frequency-domain validation

The frequency-domain behavior of the identified model can be assessed qualitatively by comparing its frequency response function (FRF) with a non-parametric estimate obtained from experimental input-output data. The non-parametric FRF is obtained by means of the MATLAB function `spaavf`, available in the *PBSID Toolbox* [30]: this function performs spectral analysis with frequency averaging, providing a smoothed estimate of the plant frequency response. The averaging operation is carried out locally in frequency and allows the variance of the spectral estimate to be reduced, at the cost of a lower frequency resolution.

For each selected combination of PBSID$_{opt}$ *hyperparameters*, the frequency response of the identified black-box model is evaluated at the same frequency points as the non-parametric estimate; the comparison is performed for both transfer functions from the lateral control input δ_{lat} to the measured outputs p and a_y.

The resulting Bode diagrams are reported in Figures 4.16–4.24. These are consistent with the results from Table 4.1, which show that the identified models are better at predicting the behavior of the output p rather than that of output a_y. Furthermore, the mismatch between the frequency responses is mainly observed in the frequency range where the corresponding coherence is lower.

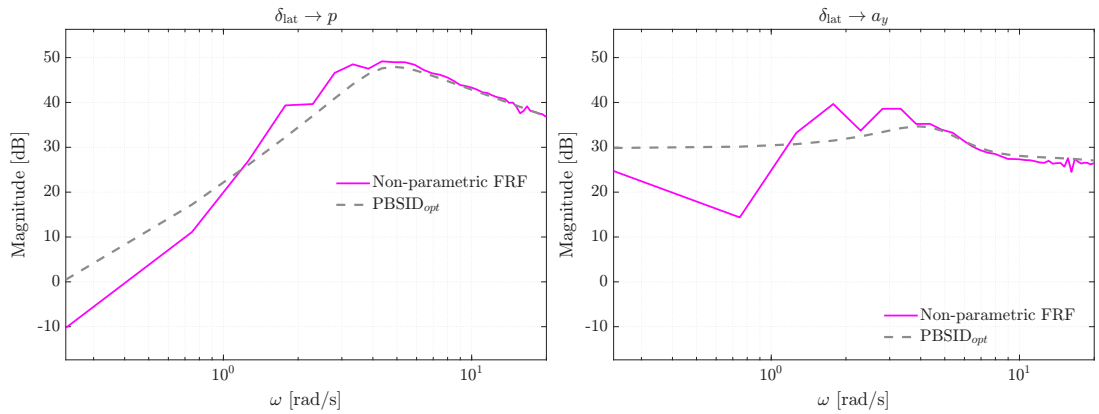


Figure 4.16: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 0% WindShaper power.

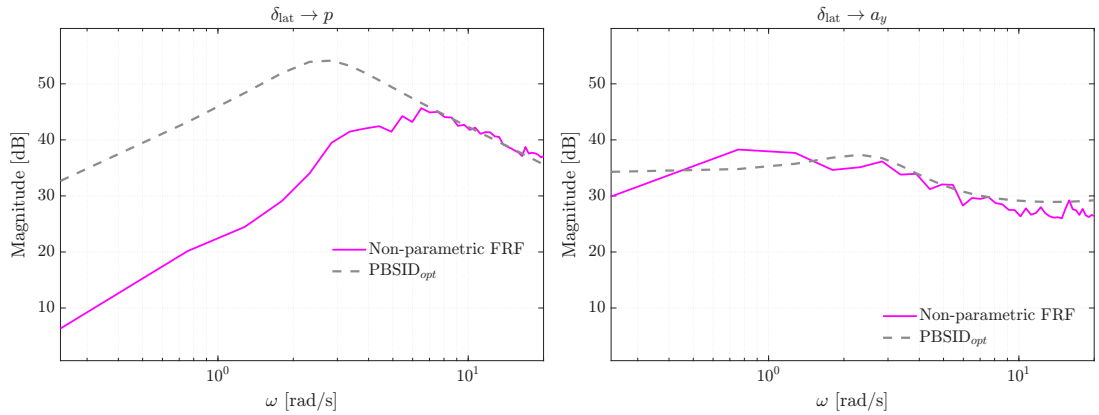


Figure 4.17: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 5% WindShaper power.

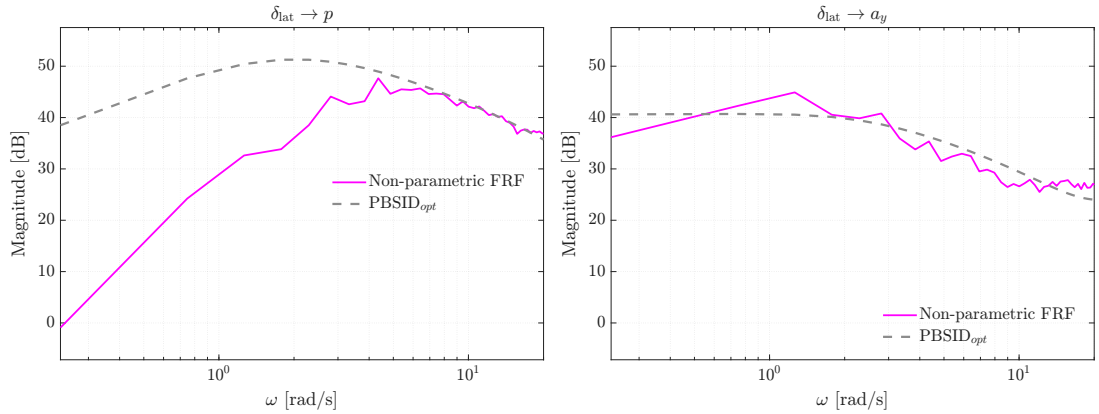


Figure 4.18: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 10% WindShaper power.

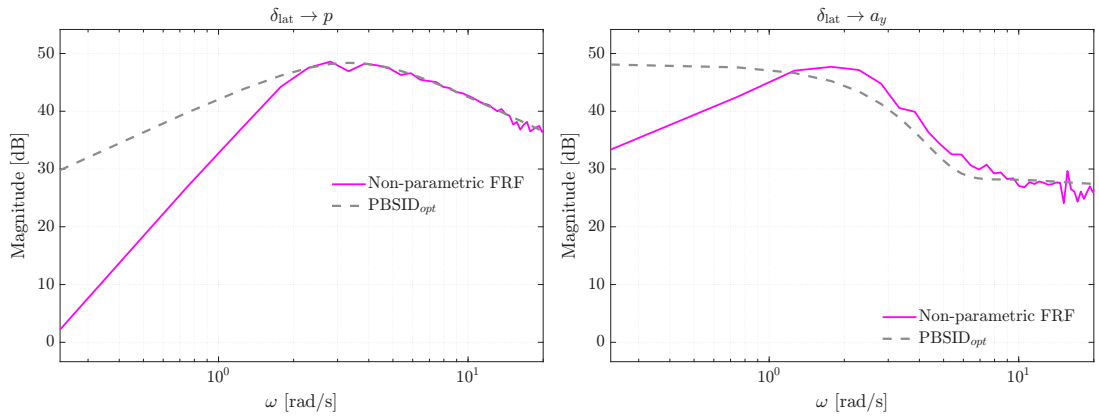


Figure 4.19: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 15% WindShaper power.

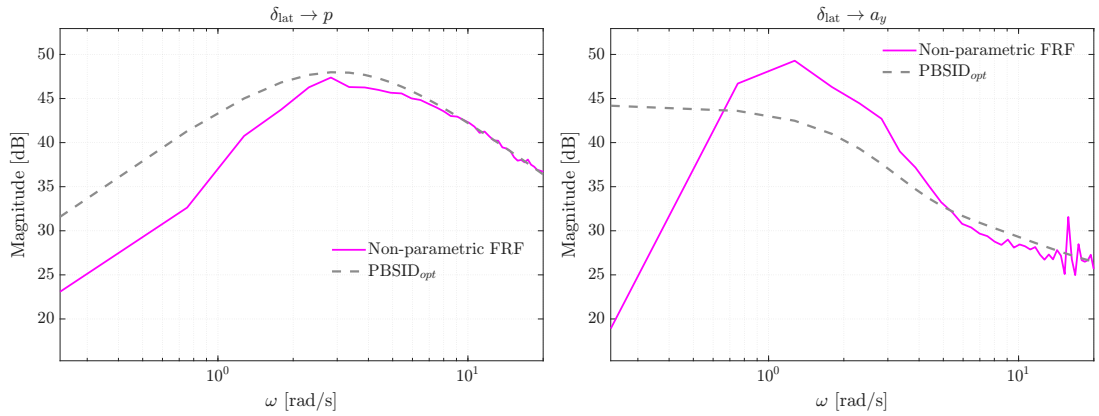


Figure 4.20: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 20% WindShaper power.

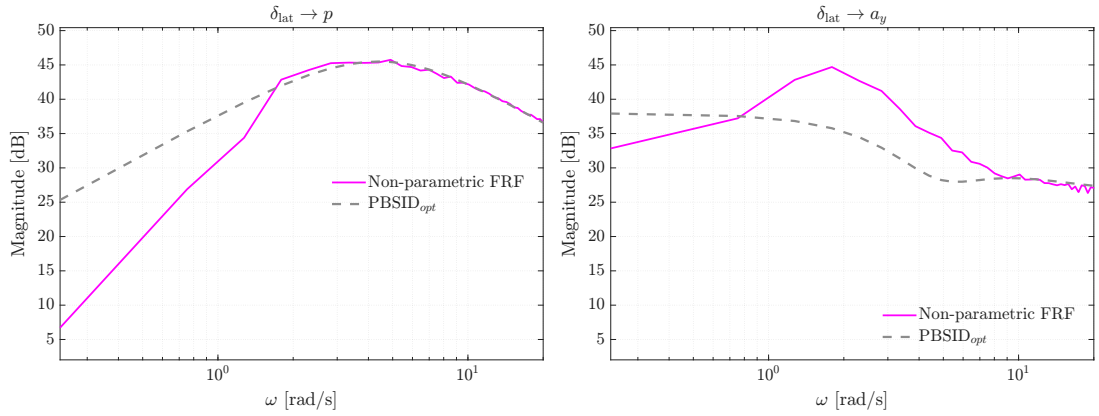


Figure 4.21: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 25% WindShaper power.

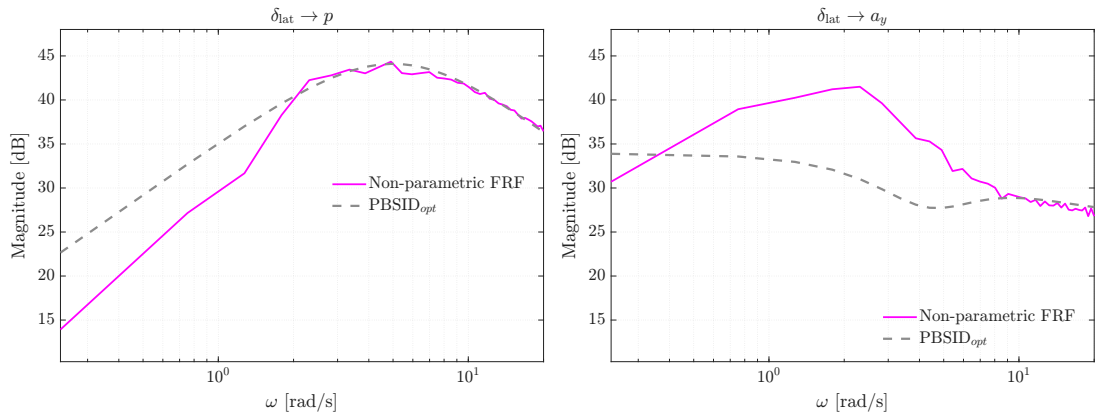


Figure 4.22: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 30% WindShaper power.

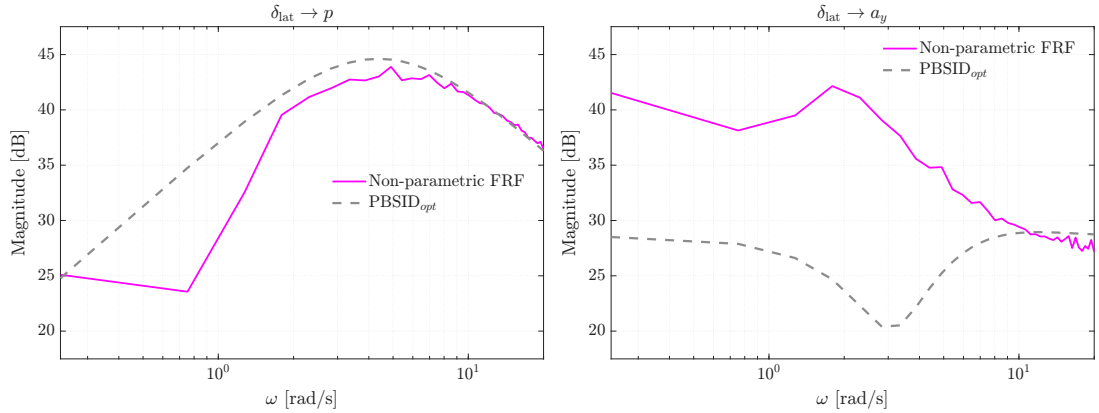


Figure 4.23: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 35% WindShaper power.

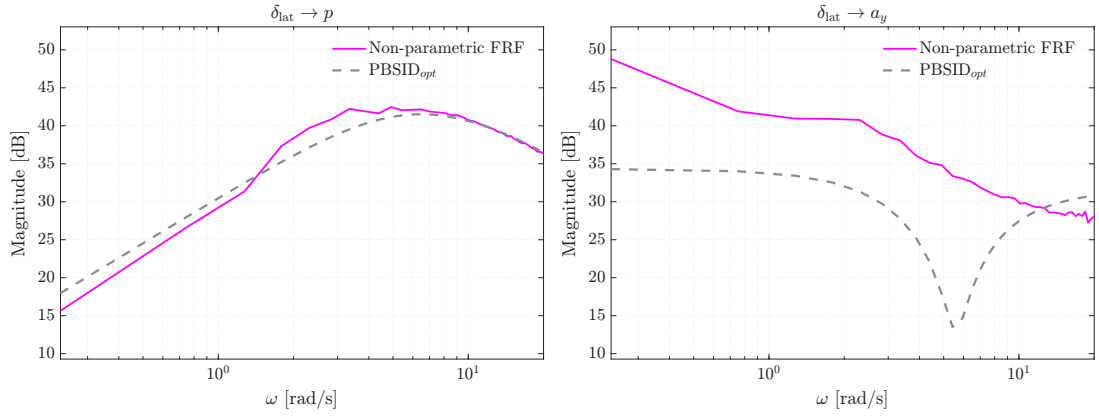


Figure 4.24: Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 40% WindShaper power.

5 | Validation procedure of structured models

This chapter describes the implementation of the different model-matching strategies, the validation procedure adopted to assess the quality of the grey-box models, and the results obtained. First, the $\mathcal{H}_\infty$ model-matching problem described in Section 2.2.1 is implemented. Then, the new approach based on frequency-domain OE minimization, detailed in Section 2.2.2, is implemented. The results are evaluated using the same criterion described in Chapter 4 for the black-box models, namely the VAF index for simulations conducted in the frequency domain. Finally, a Monte Carlo analysis is conducted to assess the robustness of the parameter estimates.

5.1. $\mathcal{H}_\infty$ model matching setup

The $\mathcal{H}_\infty$ model-matching algorithm aims at matching the unstructured model provided by the PBSID$_{opt}$ algorithm and selected in Chapter 4 with the structured model derived from the flight dynamics equations introduced in Chapter 1. In the case of lateral dynamics identification, the SIMO model defined in Equation (1.31) is characterized by the set of parameters:

$$\vartheta_{lat} = [Y_v \; Y_p \; L_v \; L_p \; Y_\delta \; L_\delta]^T , \tag{5.1}$$

which can be implemented in MATLAB as tunable real parameters through the function `realp`. The structured grey-box model $G_s(s, \vartheta)$ also depends on the trim roll angle ϕ_0, whose value can be determined from experimental data.

As described in Section 2.1.2, both the unstructured and structured input-output maps are normalized to avoid weighting the optimization algorithm for the two transfer functions involved in the lateral dynamics: it is worth noticing that the scale factors considered in this phase should be associated with the original dataset signals before applying the prefiltering for type correction. Since the structured system is obtained in continuous-time, the unstructured models obtained are converted from discrete-time to continuous-

time using *Tustin's method*.

Since the model-matching problem is non-convex and non-smooth, the final estimate may depend on several implementation choices. First, since the experiments are performed in a given frequency range, the $\mathcal{H}_\infty$ norm minimization should also be focused within the same bandwidth: indeed, outside of this frequency band the matching between the two transfer functions becomes less relevant. From the analysis on input-output coherence of Figure 4.2, the focus bandwidth is set to 1.5–8 rad/s for models up to 20% WindShaper power, and up to 16 rad/s for higher power levels. The stopping criteria for the optimization algorithm are defined in terms of *relative tolerance* and *maximum iterations*. Finally, since the solver may converge to a local minimum point, the initial guess on the physical parameters can greatly influence the final results, therefore a common practice is to perform multiple additional random initializations as a robustness check. Furthermore, the final result $\vartheta^\star$ is used as a suitable initial guess for the following flight speed, as suggested in [13].
The procedure is summarized in Algorithm 5.1:

Algorithm 5.1 PBSID$_{opt}$–$\mathcal{H}_\infty$ model matching

1: **Input:** Unstructured discrete-time model $\hat{G}_{ns}(z)$, initial guess ϑ_0, scale factors D_u, D_y.
2: **Output:** Estimated structured continuous-time model $\hat{G}_s\left(s, \hat{\vartheta}\right)$.
3: Convert the system to continuous time:
4: $\hat{G}_{ns}(s) \leftarrow$ `d2c`$(\hat{G}_{ns}(z),$ `'tustin'`$)$;
5: Scale the unstructured system: $\hat{G}_{ns}^{sc}(s) = D_y^{-1}\,\hat{G}_{ns}(s)\,D_u$
6: Define the structured grey-box model $G_s(s, \vartheta)$
7: Scale the structured model: $G_s^{sc}(s, \vartheta) = D_y^{-1}\,G_s(s, \vartheta)\,D_u$
8: Build the interconnected system $E\,(s, \vartheta)$ from Figure 2.2 using `connect`
9: Define the requirements Req and additional optimization options Opt
10: Run the optimization:
11: $\hat{E}\left(s, \hat{\vartheta}\right) \leftarrow$ `systune` $(E\,(s, \vartheta)\,, \text{Req}, \text{Opt})$
12: Extract the tuned parameters: $\hat{\vartheta} \leftarrow \hat{E}\left(s, \hat{\vartheta}\right)$
13: Convert the estimated system to physical scale:
14: $\hat{G}_s\left(s, \hat{\vartheta}\right) = D_y\,\hat{G}_s^{sc}\left(s, \hat{\vartheta}\right)\,D_u^{-1}$
15: **Return:** $\hat{G}_s\left(s, \hat{\vartheta}\right)$.

5.2. Frequency-domain OE minimization setup

The frequency-domain Output-Error minimization procedure is introduced as an alternative structured model-matching approach to the $\mathcal{H}_\infty$ norm setup described in the previous section. As stated in Section 2.2.2, the model-matching problem is solved by minimizing the cost function defined in Equation (2.36). The structured grey-box model $G_s(s, \vartheta)$ is the same as the one introduced in the previous section, and it depends on the lateral-dynamics parameter vector defined in Equation (5.1). The frequency response of the identified black-box model is evaluated on a finite grid $\Omega = \{\omega_1, \ldots, \omega_{N_\omega}\}$, whose lower and upper bounds are chosen consistently with the frequency range considered for the $\mathcal{H}_\infty$ model-matching problem. The corresponding complex frequency-response samples are:

$$\hat{\mathcal{G}}_k \coloneqq \hat{G}_{ns}(j\omega_k) \qquad k = 1, \ldots, N_\omega. \tag{5.2}$$

Therefore, unlike the $\mathcal{H}_\infty$ setup, whose objective is related to the worst-case mismatch over the considered frequency interval, the frequency-domain OE formulation minimizes a cost function built from the mismatch between the unstructured and structured frequency responses at the selected frequency samples.

In MATLAB, the complex frequency-response samples are stored in an `idfrd` object. The structured grey-box model is stored as a continuous-time `idgrey` object, implemented by means of a user-defined ordinary differential equation (ODE) function, which takes as input the initial guess vector ϑ_0, as well as auxiliary parameters such as the trim roll angle ϕ_0.

The minimization is carried out using the MATLAB function `greyest`, which estimates the parameters of a linear grey-box model using either time-domain or frequency-domain data: in this work, the same convergence settings adopted for the $\mathcal{H}_\infty$ model-matching procedure, i.e. the *relative tolerance* and *maximum iterations*, are used whenever possible, in order to make the comparison between the two structured identification approaches as consistent as possible. The function `greyestOptions` allows the user to choose the numerical search method used for the iterative parameter estimation. If no specific search method is selected, a combination of line search algorithms is tried in sequence at each iteration, and the first descent direction leading to a reduction in estimation cost is used. The procedure is summarized in Algorithm 5.2:

Algorithm 5.2 PBSID$_{opt}$–frequency-domain OE model matching

1: **Input:** Unstructured discrete-time model $\hat{G}_{ns}(z)$, initial guess ϑ_0, scale factors D_u, D_y.
2: **Output:** Estimated structured continuous-time model $\hat{G}_s\left(s, \hat{\vartheta}\right)$.
3: Convert the system to continuous time:
4: $\hat{G}_{ns}(s) \leftarrow$ `d2c`$(\hat{G}_{ns}(z),$ `'tustin'`$)$;
5: Scale the unstructured system: $\hat{G}^{sc}_{ns}(s) = D_y^{-1}\,\hat{G}_{ns}(s)\,D_u$
6: Convert the transfer function into a `idfrd` object: $\hat{G}_{id}(s) \leftarrow$ `idfrd` $\left(\hat{G}^{sc}_{ns}(s), \Omega\right)$
7: Define the structured grey-box model: $G_s(s,\vartheta) \leftarrow$ `idgrey` (@`odefun`, ϑ_0, `aux`)
8: Scale the structured model: $G^{sc}_s(s,\vartheta) = D_y^{-1}\,G_s(s,\vartheta)\,D_u$
9: Define the optimization options Opt
10: Run the optimization:
11: $\hat{G}^{sc}_s\left(s,\hat{\vartheta}\right) \leftarrow$ `greyest` $\left(\hat{G}_{id}(s), G^{sc}_s(s,\vartheta), \text{Opt}\right)$
12: Convert the estimated system to physical scale:
13: $\hat{G}_s\left(s,\hat{\vartheta}\right) = D_y\,\hat{G}^{sc}_s\left(s,\hat{\vartheta}\right)\,D_u^{-1}$
14: **Return:** $\hat{G}_s\left(s,\hat{\vartheta}\right)$.

5.3. Validation

This section presents the validation results of the structured models estimated by the two different model-matching algorithms. The validation criteria used are the same ones that were defined in Section 4.3.

Let $\hat{G}_{ns}(s)$ denote the continuous-time black-box model selected in Chapter 4, and let $G_{\mathcal{H}_\infty}(s, \hat{\vartheta}_{\mathcal{H}_\infty})$ and $G_{OE}(s, \hat{\vartheta}_{OE})$ denote the grey-box models obtained respectively, through the $\mathcal{H}_\infty$ model-matching procedure of Algorithm 5.1 and the frequency-domain Output-Error minimization procedure of Algorithm 5.2; both grey-box models share the same structured parametrization defined in Equation (1.31), namely from input δ_{lat} to outputs p and a_y. The validation is first quantified in terms of the VAF index defined in Equation (4.1): the simulated time histories are obtained using the same procedure defined in Section 4.3.1. The numerical results for both methods are summarized in Table 5.1.

Power [%]	PBSID$_{opt}$		$\mathcal{H}_\infty$		Frequency-domain OE	
	VAF$_p$ [%]	VAF$_{a_y}$ [%]	VAF$_p$ [%]	VAF$_{a_y}$ [%]	VAF$_p$ [%]	VAF$_{a_y}$ [%]
0	97.01	49.93	94.38	41.33	93.98	37.30
5	70.66	51.27	0.00	0.00	68.53	23.34
10	93.43	45.26	36.14	0.00	70.07	0.00
15	98.75	61.92	62.83	0.00	76.32	0.00
20	98.63	60.87	52.39	0.00	79.70	42.71
25	99.49	72.37	62.59	0.00	78.45	45.26
30	99.35	72.75	61.02	59.78	66.31	54.91
35	99.06	60.87	51.84	54.70	58.33	60.30
40	99.25	49.24	63.98	21.75	70.13	53.96

Table 5.1: Cross-validation performance of the grey-box models obtained with $\mathcal{H}_\infty$ and frequency-domain OE methods.

The same trend observed in the validation of the unstructured models is confirmed, with higher VAF values for the output p than for the output a_y, due to the noise component present in the second output even after filtering.

5.3.1. $\mathcal{H}_\infty$ model matching

The validation results obtained with the $\mathcal{H}_\infty$ model matching are presented first. The resulting time histories obtained from the simulation in the frequency domain are compared with the measured outputs and with the response of the unstructured models in Figures 5.1–5.9. At 0% WindShaper power (Figure 5.1), the optimization algorithm performed as expected and was able to obtain satisfactory results, with only a slight decrease in VAF value for both roll rate and acceleration. At higher WindShaper power levels, however, the results obtained were not satisfactory: in particular, while most identified grey-box models were at least able to reproduce the roll rate channel, although with a decrease in accuracy from the PBSID$_{opt}$ identified models, the one at 5% power (Figure 5.2) failed to reproduce either output channel.

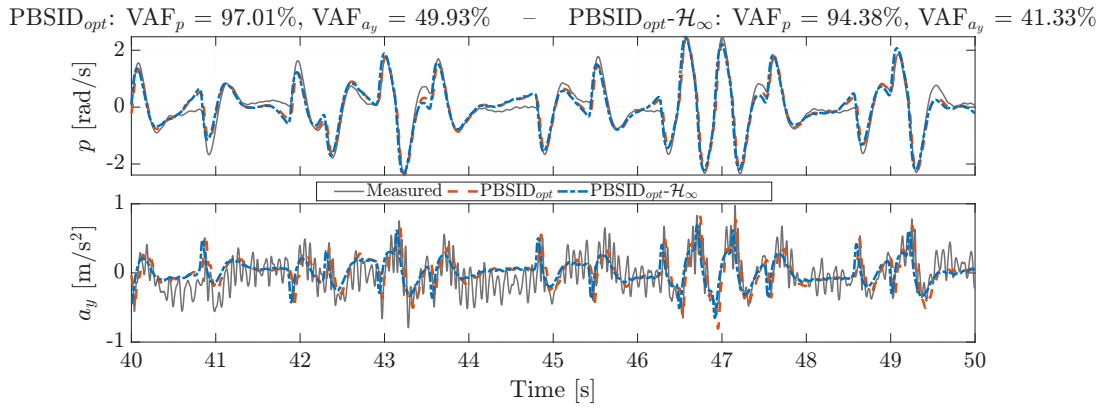


Figure 5.1: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 0% power.

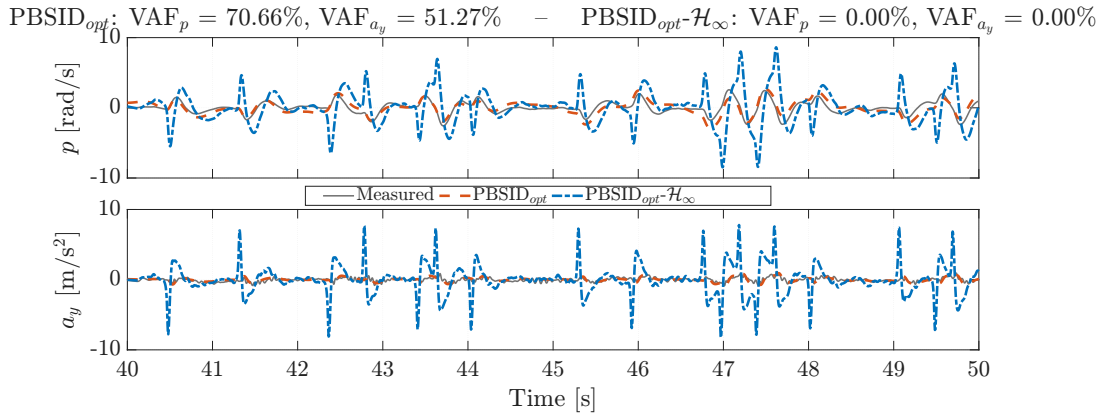


Figure 5.2: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 5% power.

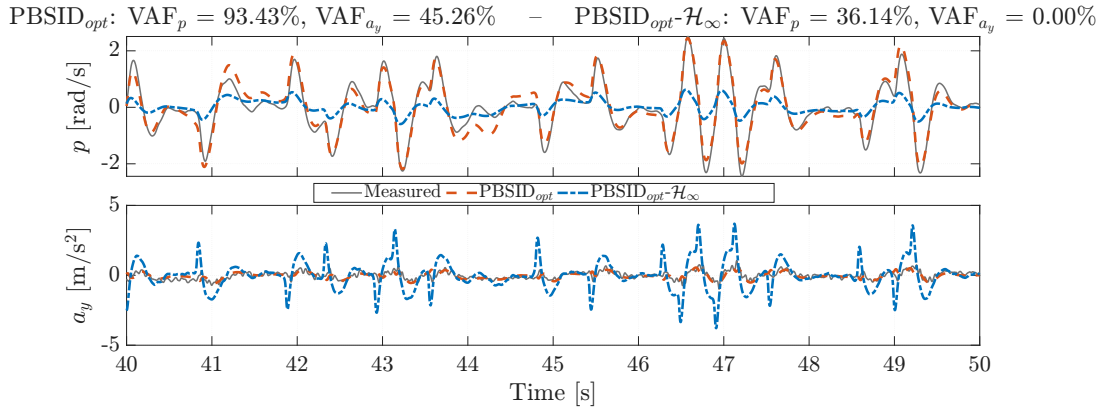


Figure 5.3: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 10% power.

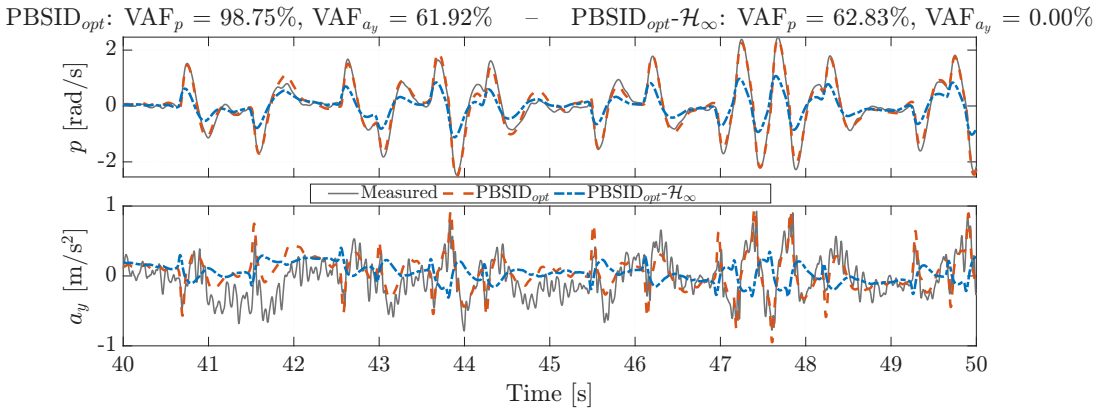


Figure 5.4: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 15% power.

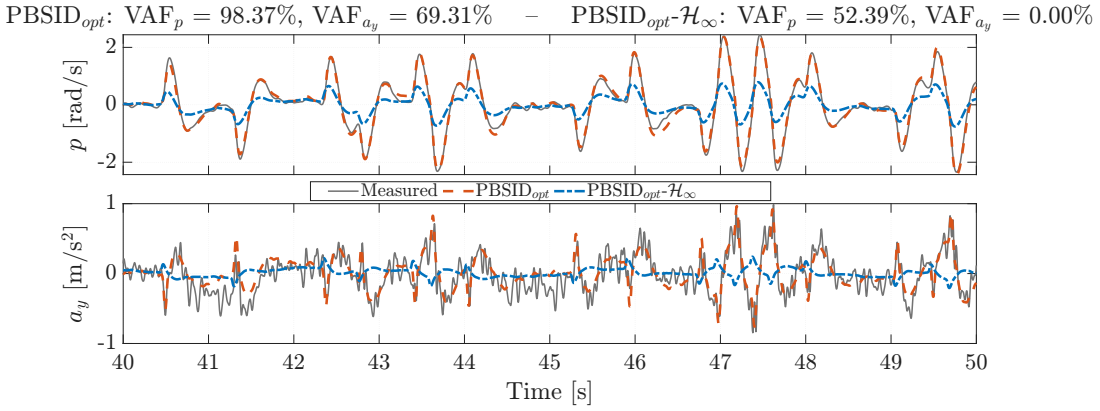


Figure 5.5: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 20% power.

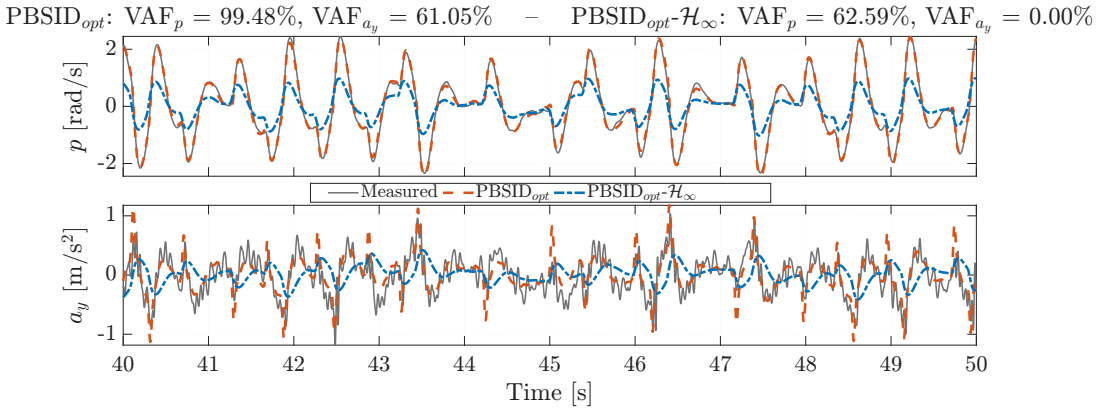


Figure 5.6: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 25% power.

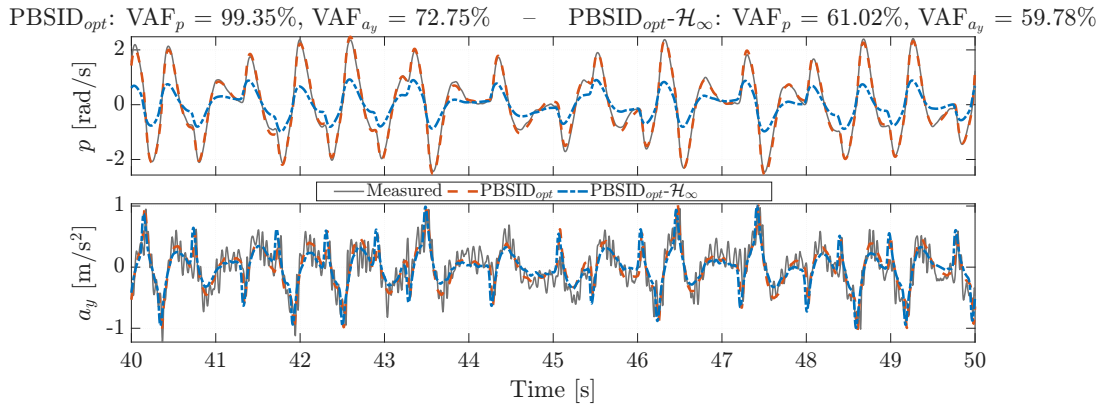


Figure 5.7: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 30% power.

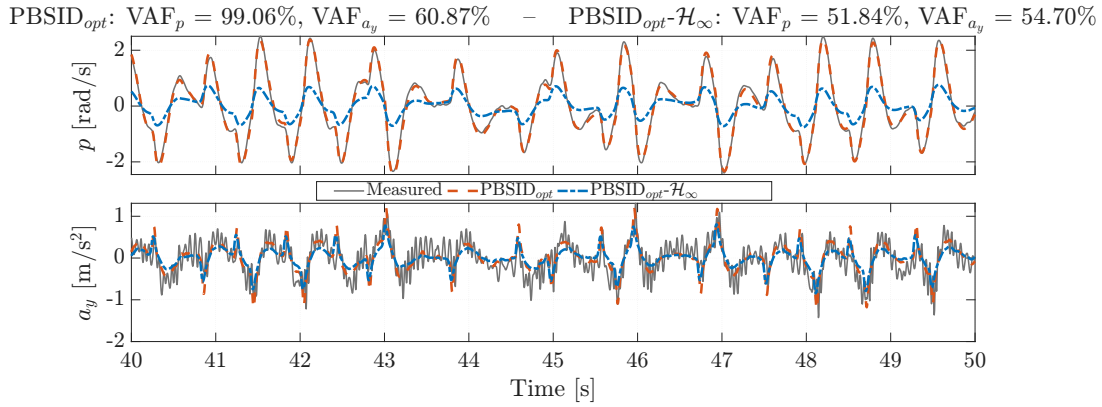


Figure 5.8: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 35% power.

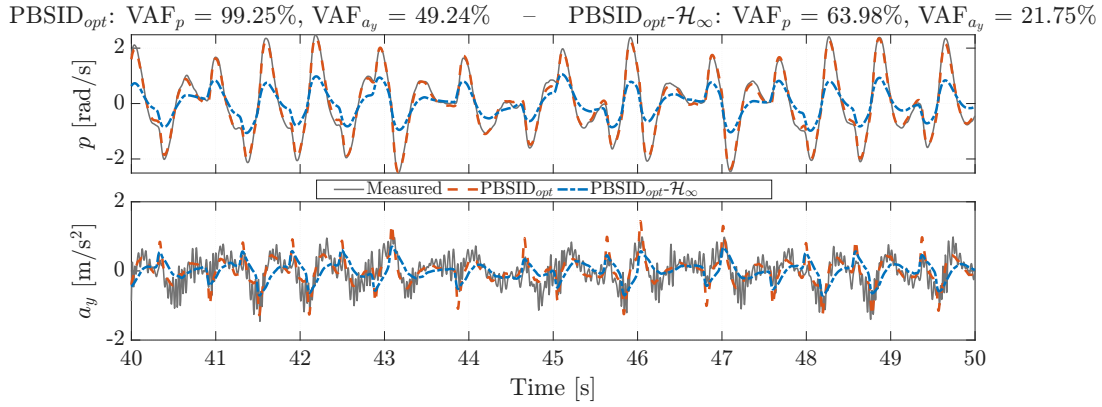
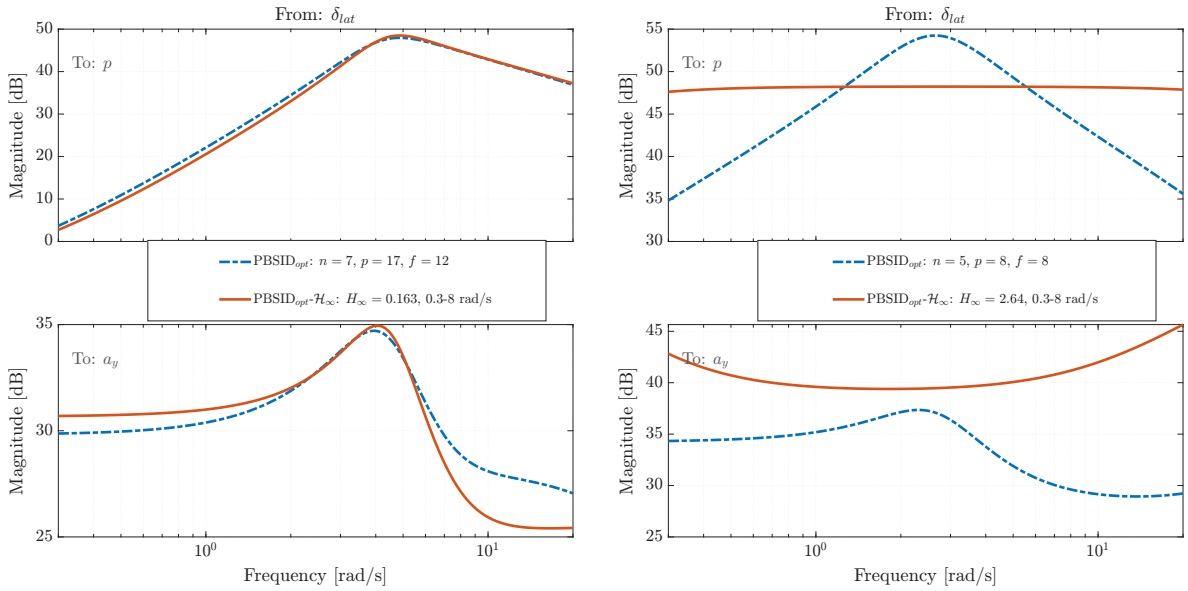


Figure 5.9: Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 40% power.

Figures 5.10–5.14 show the FRF of the grey-box models compared with the ones of the estimated black-box models. In each figure, the value of the soft constraint returned by `systune` is shown: the function converts the requirements given as input into a normalized

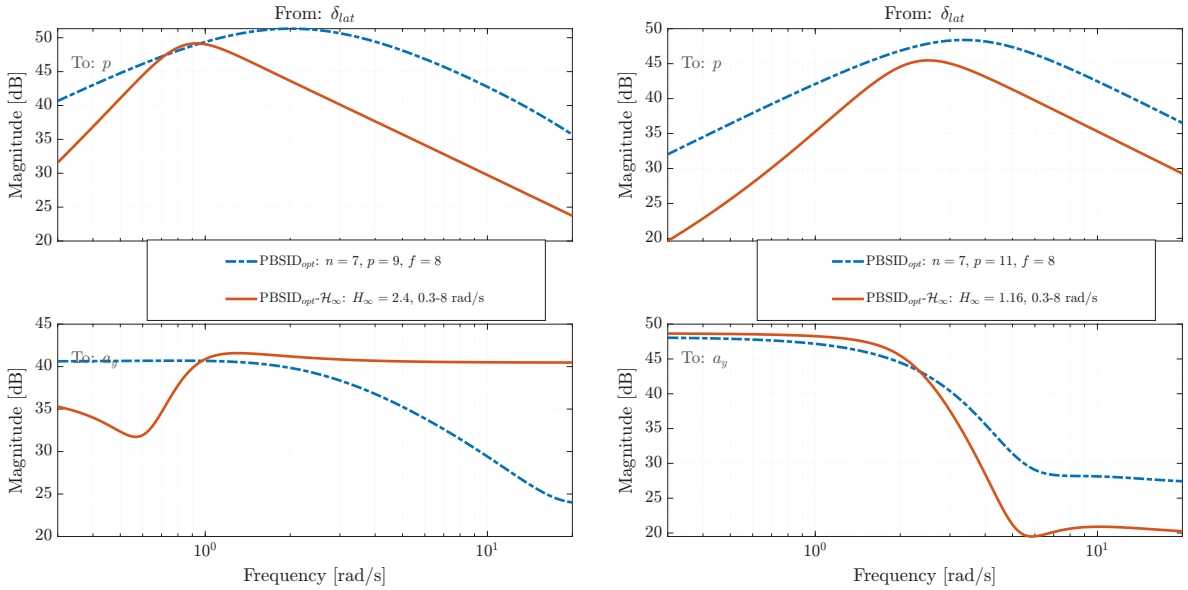
function $f(\vartheta)$ to be minimized, and the soft constraint γ is the best achieved value, which is considered satisfied when $\gamma < 1$. Again, the model at 0% power is the most accurate, as corroborated by the final soft constraint value $\gamma = 0.163$, while the one at 5% power is the worst one with a value of $\gamma = 2.64$.



(a) 0% power.

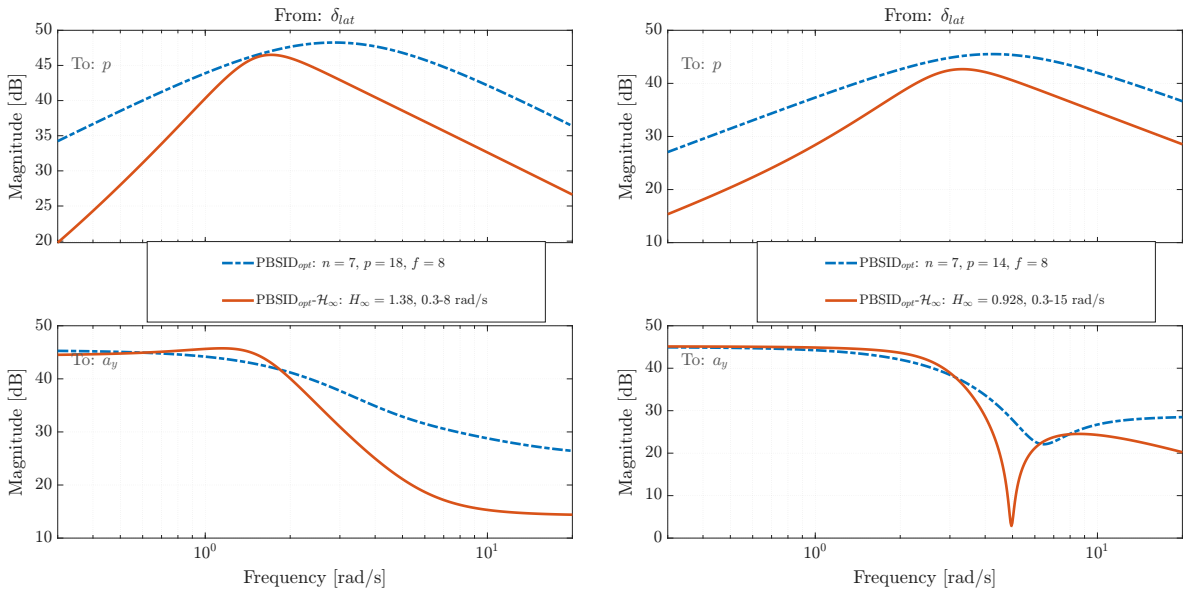
(b) 5% power.

Figure 5.10: Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 0% and 5% power.



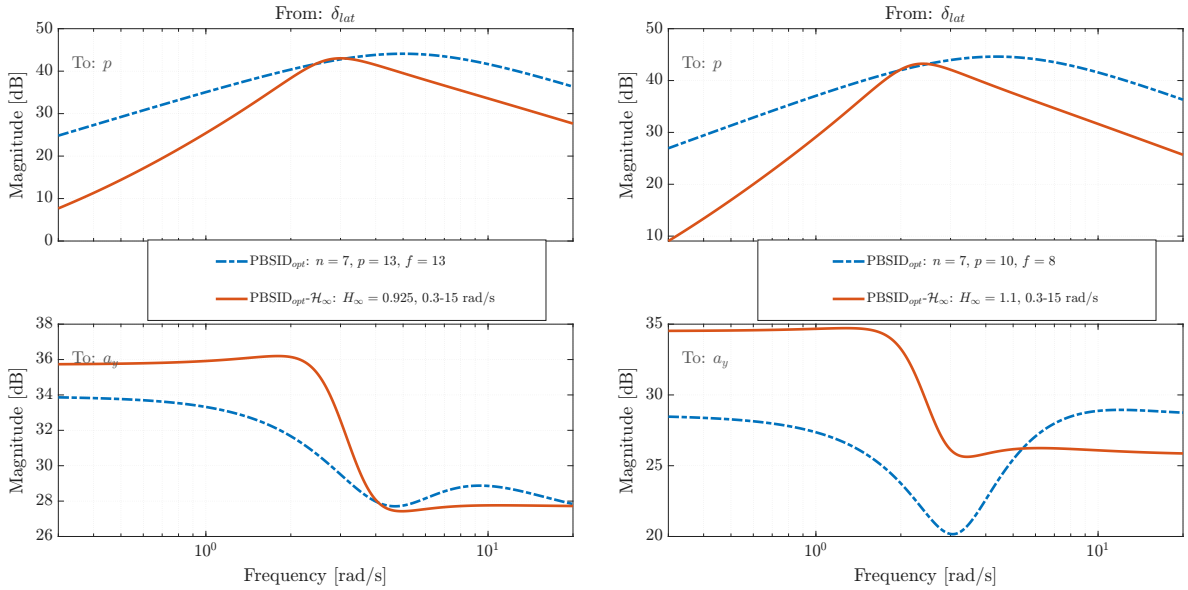
(a) 10% power. (b) 15% power.

Figure 5.11: Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 10% and 15% power.



(a) 20% power. (b) 25% power.

Figure 5.12: Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 20% and 25% power.



(a) 30% power. (b) 35% power.

Figure 5.13: Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 30% and 35% power.

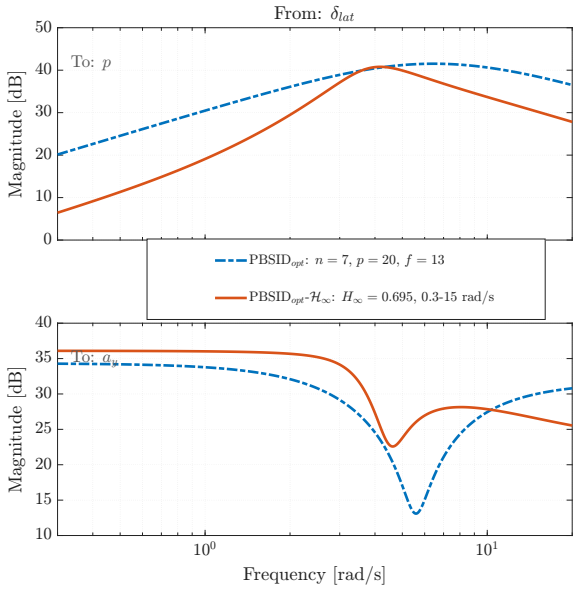
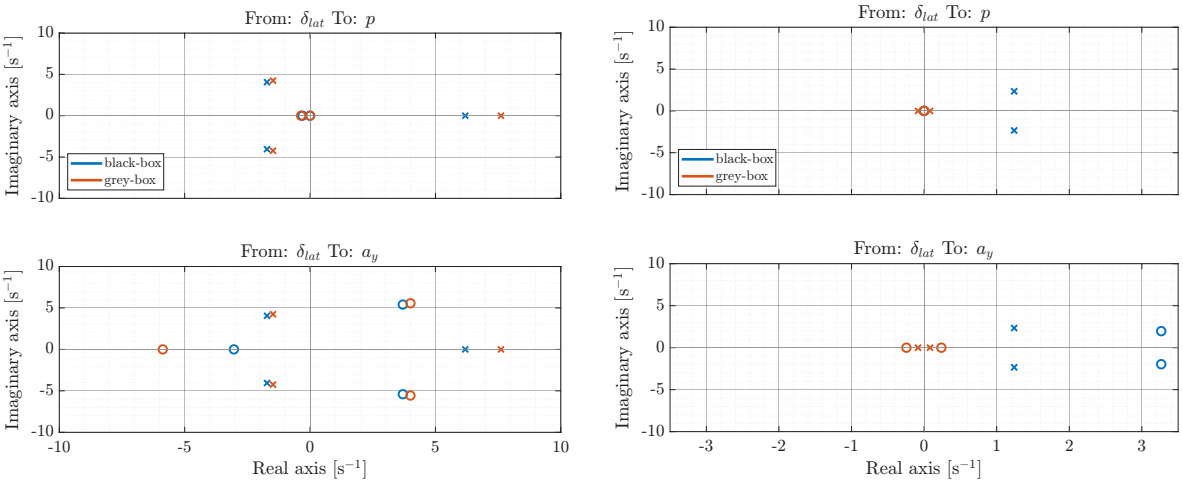


Figure 5.14: Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box model at 40% power.

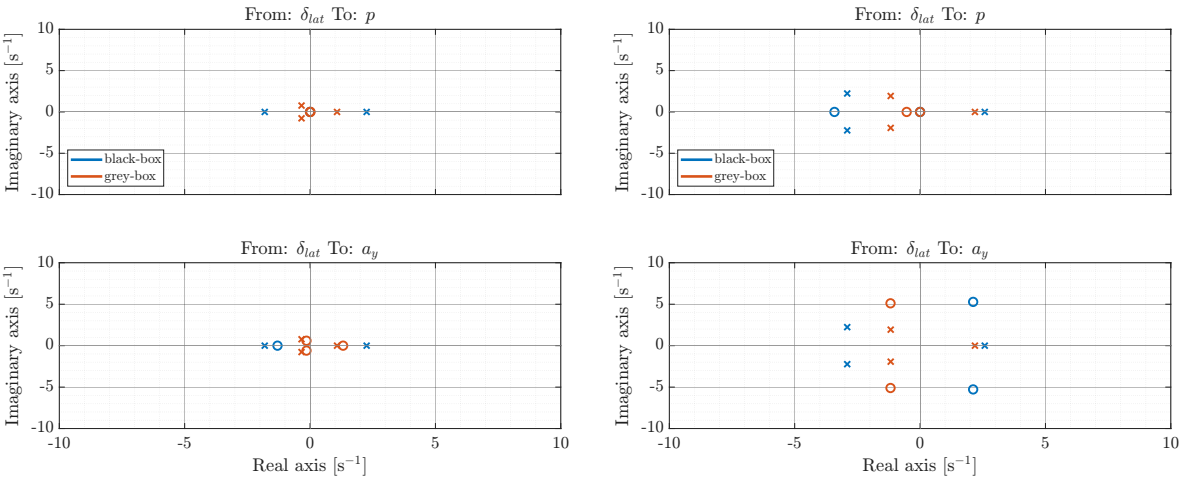
Additionally, the pole-zero maps for both the unstructured and structured models are reported in Figures 5.15–5.19: the comparison focuses on the poles and zeros inside or close to the frequency bandwidth of interest, while high-frequency poles and zeros introduced by

discretization, conversion and type-correction in the black-box models are omitted. The general structure of the poles and zeros is visible in both the structured and unstructured models.



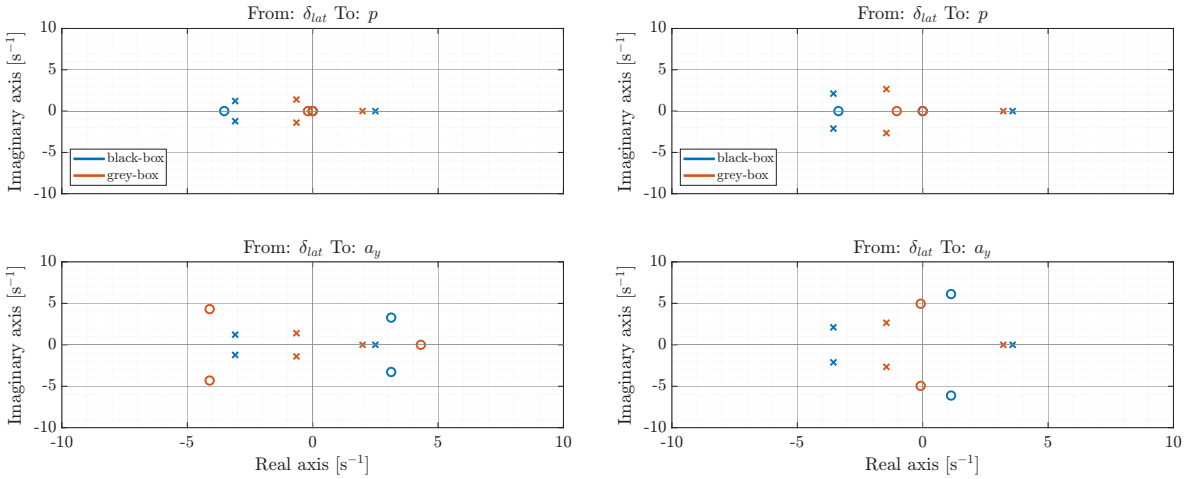
(a) 0% power. (b) 5% power.

Figure 5.15: Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 0% and 5% power.



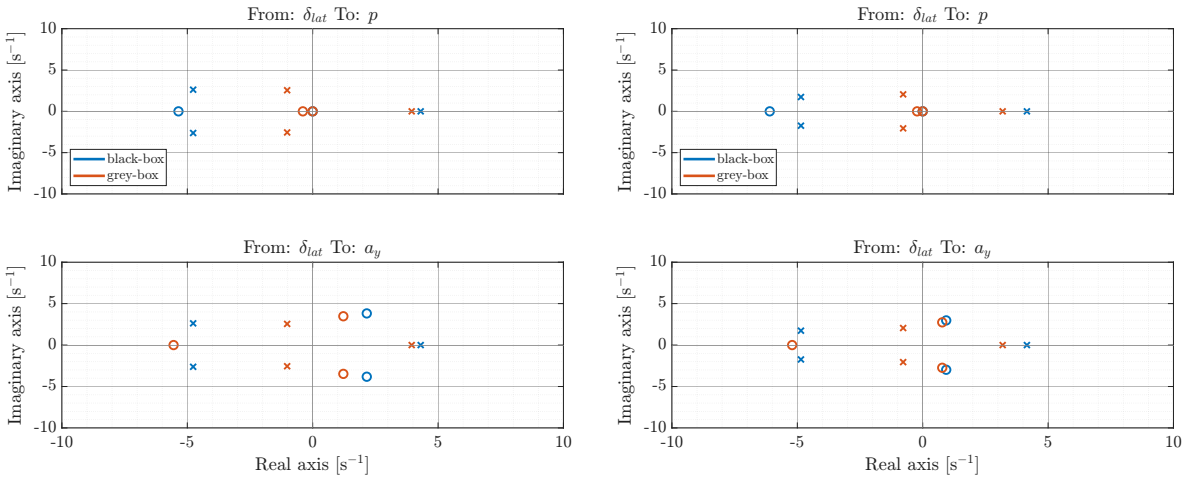
(a) 10% power. (b) 15% power.

Figure 5.16: Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 10% and 15% power.



(a) 20% power. (b) 25% power.

Figure 5.17: Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 20% and 25% power.



(a) 30% power. (b) 35% power.

Figure 5.18: Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 30% and 35% power.

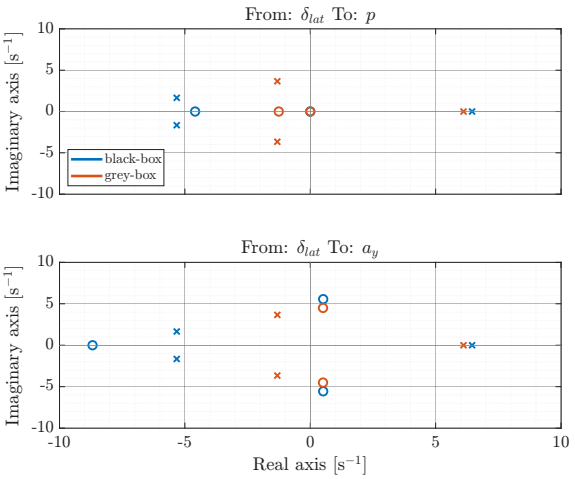


Figure 5.19: Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box model at 40% power.

5.3.2. Frequency-domain OE model matching

The same validation sequence is then repeated for the grey-box models obtained through frequency- domain Output-Error minimization. Figures 5.20–5.28 show that the method is able to better capture the behavior of both outputs, in particular of the roll rate p. Notable exceptions are the models obtained at 10% and 15% power, which are unable to capture the behavior of the lateral acceleration a_y, as shown by the time history in Figures 5.22–5.23.

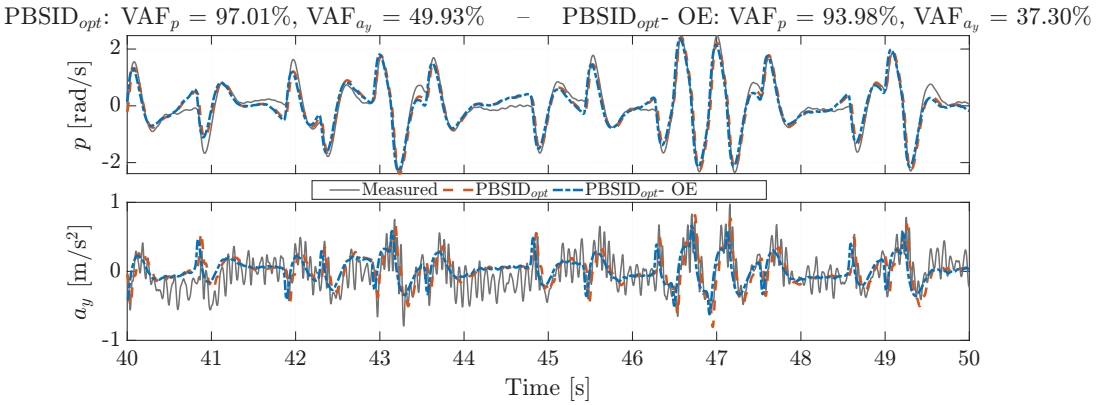


Figure 5.20: Time-domain cross-validation of the OE grey-box model at 0% power.

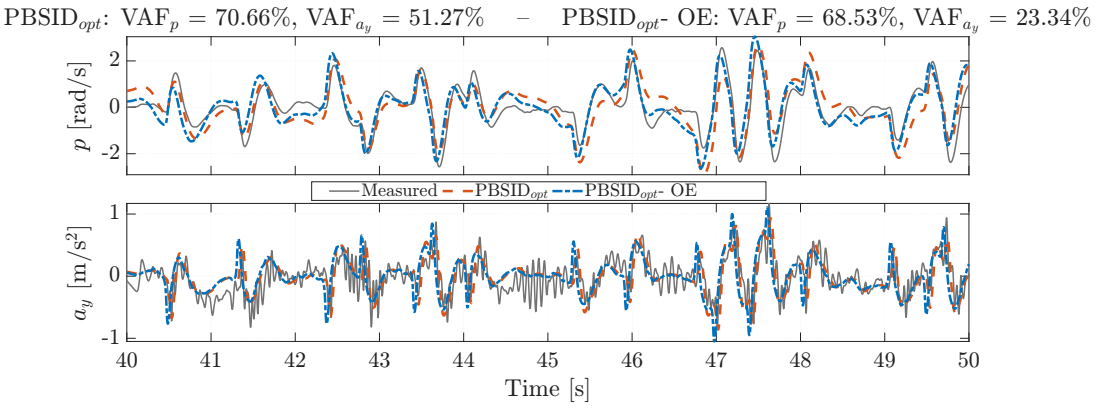


Figure 5.21: Time-domain cross-validation of the OE grey-box model at 5% power.

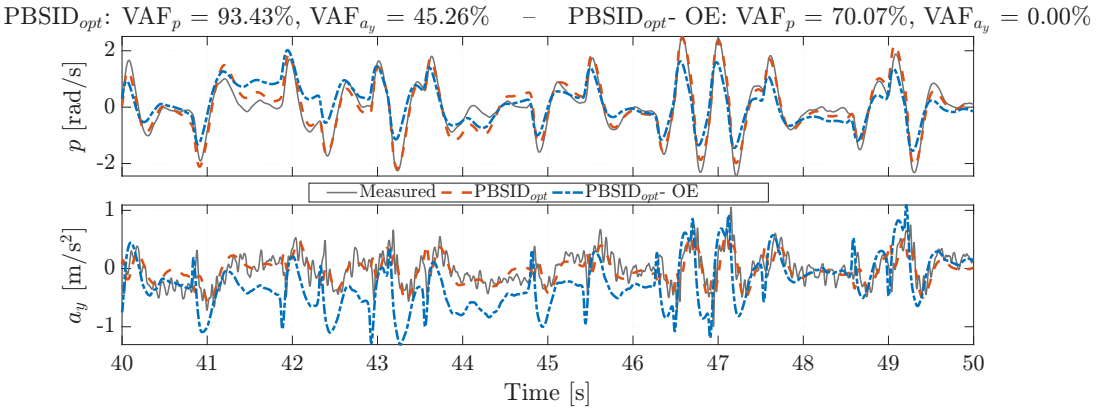


Figure 5.22: Time-domain cross-validation of the OE grey-box model at 10% power.

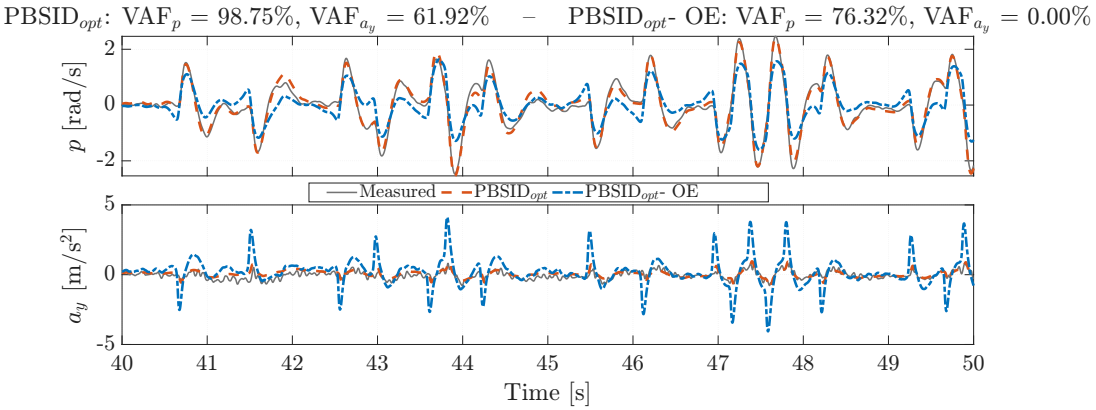


Figure 5.23: Time-domain cross-validation of the OE grey-box model at 15% power.

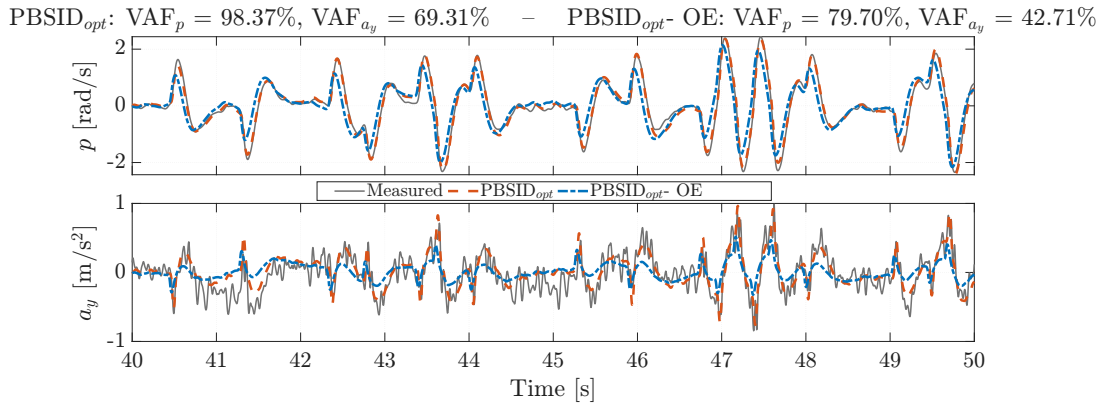


Figure 5.24: Time-domain cross-validation of the OE grey-box model at 20% power.

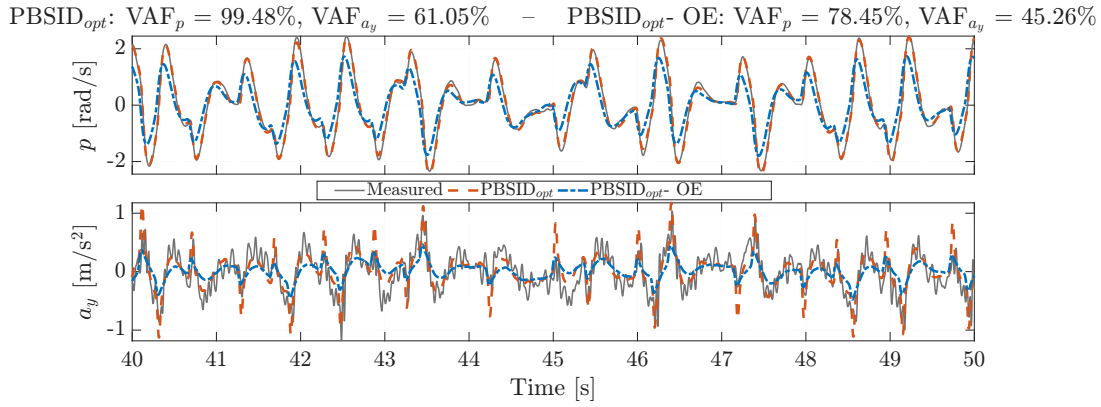


Figure 5.25: Time-domain cross-validation of the OE grey-box model at 25% power.

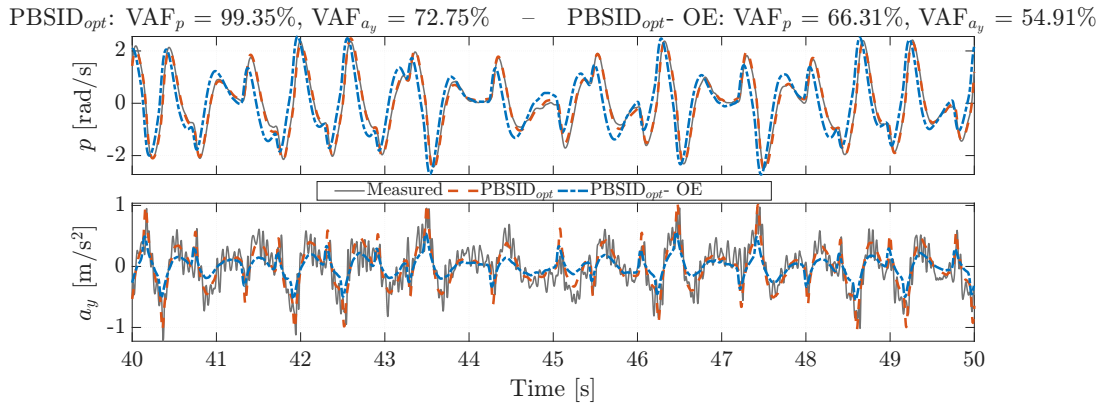


Figure 5.26: Time-domain cross-validation of the OE grey-box model at 30% power.

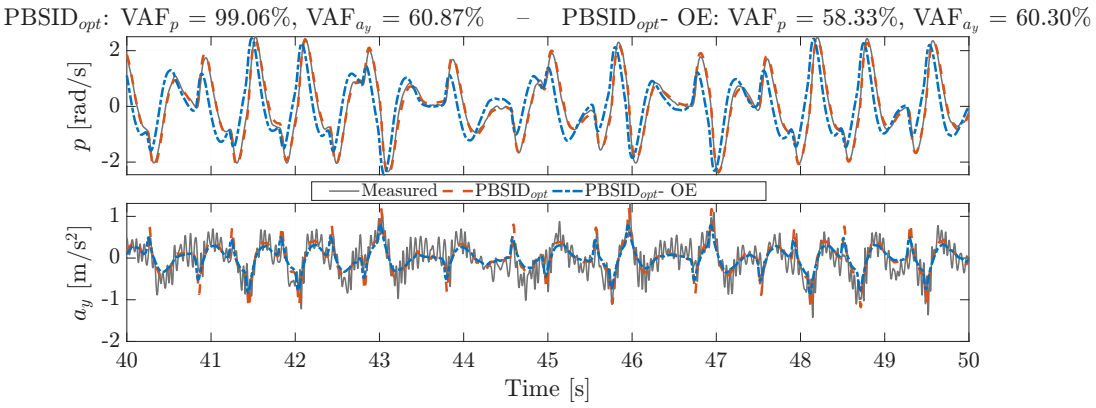


Figure 5.27: Time-domain cross-validation of the OE grey-box model at 35% power.

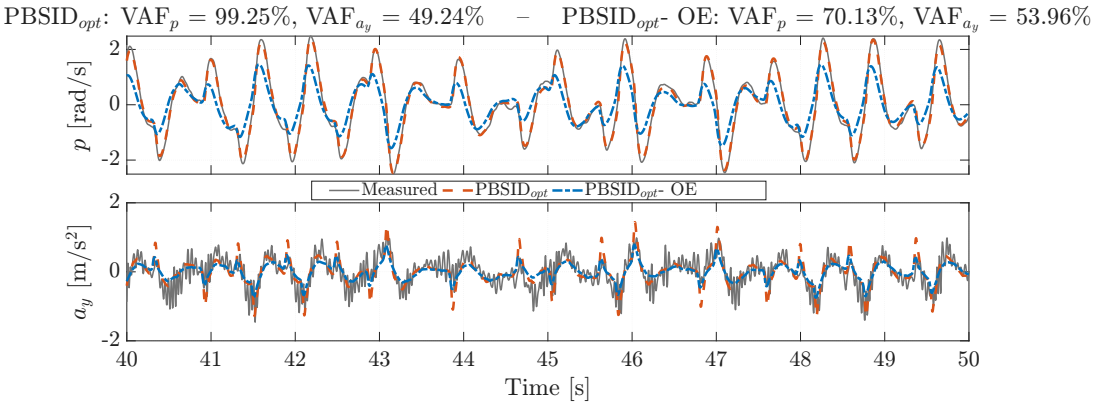
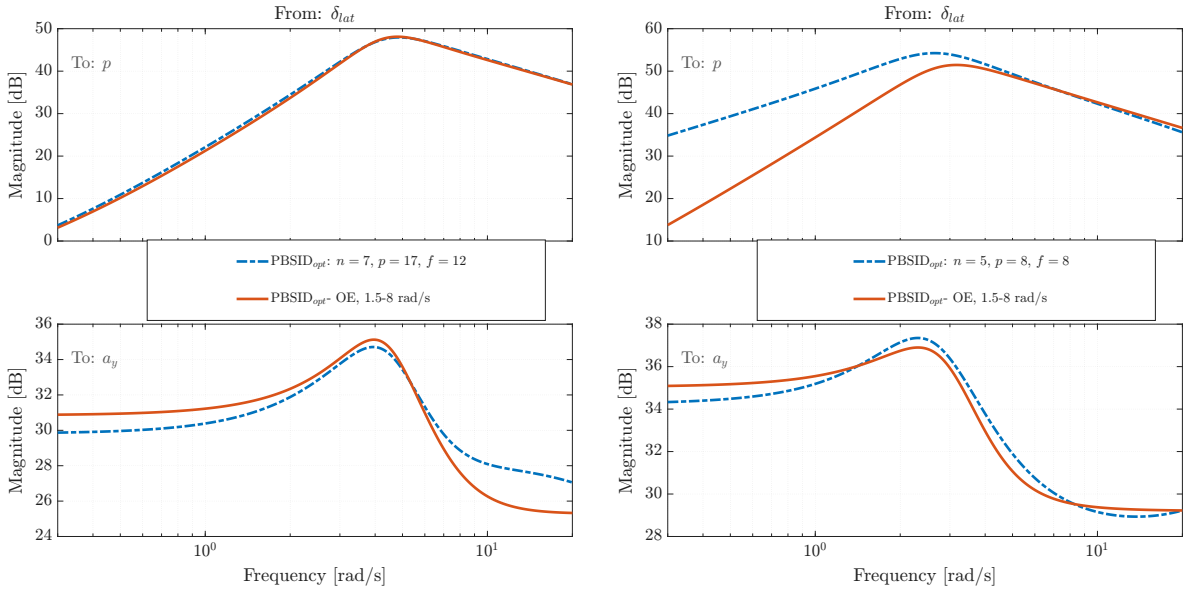


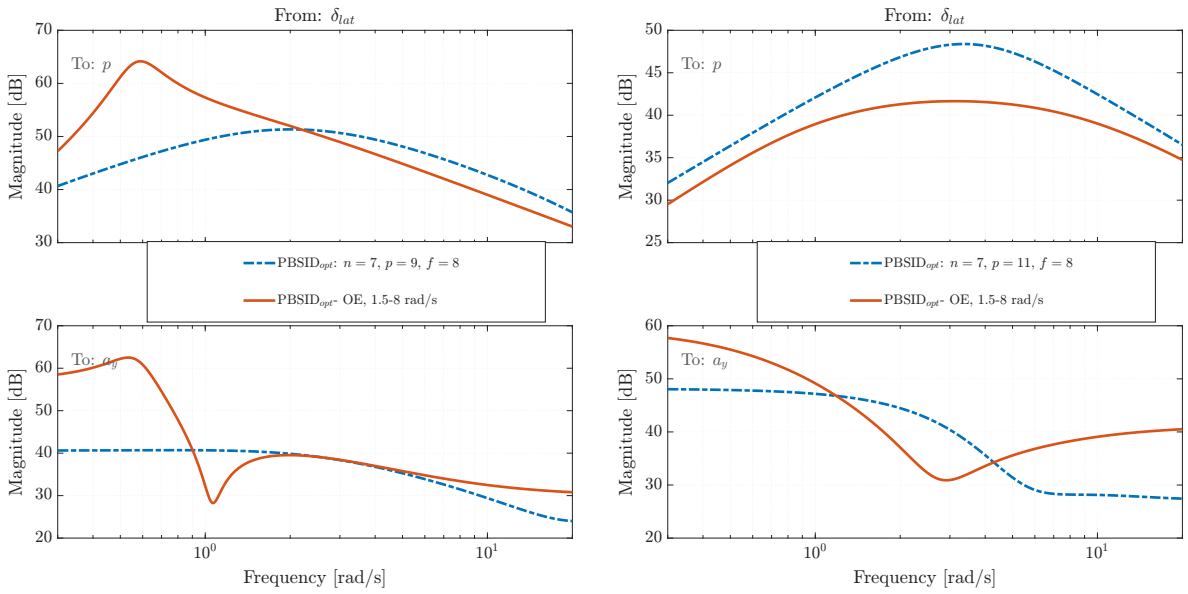
Figure 5.28: Time-domain cross-validation of the OE grey-box model at 40% power.

The same conclusions can be drawn from the frequency response functions, which show that the models at 10% and 15% perform poorly on the lateral acceleration channel (Figure 5.30) due to the intrinsically noisier nature of the measured output. Furthermore, the magnitude plots show that the biggest discrepancies for the FRF from δ_{lat} to p are in the range of frequencies where the coherence value is the lowest, as shown in Figure 4.2.



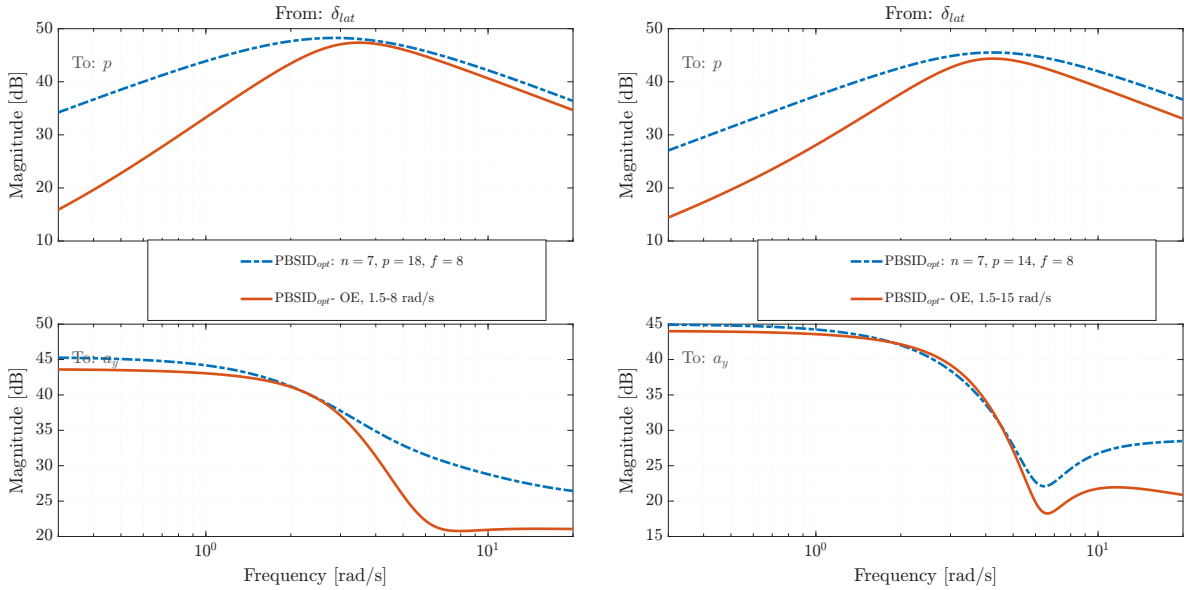
(a) 0% power. (b) 5% power.

Figure 5.29: Frequency response function of the black-box and frequency-domain OE grey-box models at 0% and 5% power.



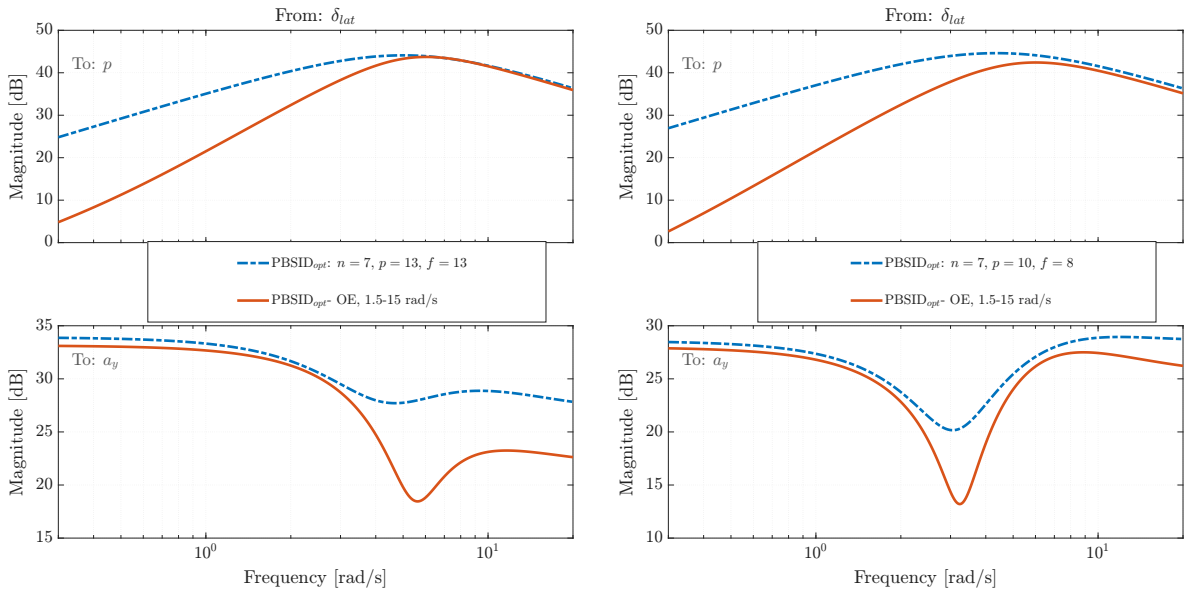
(a) 10% power. (b) 15% power.

Figure 5.30: Frequency response function of the black-box and frequency-domain OE grey-box models at 10% and 15% power.



(a) 20% power. (b) 25% power.

Figure 5.31: Frequency response function of the black-box and frequency-domain OE grey-box models at 20% and 25% power.



(a) 30% power. (b) 35% power.

Figure 5.32: Frequency response function of the black-box and frequency-domain OE grey-box models at 30% and 35% power.

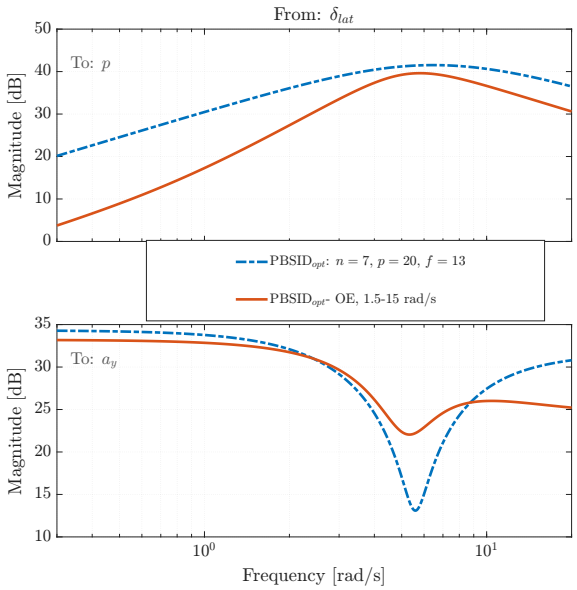
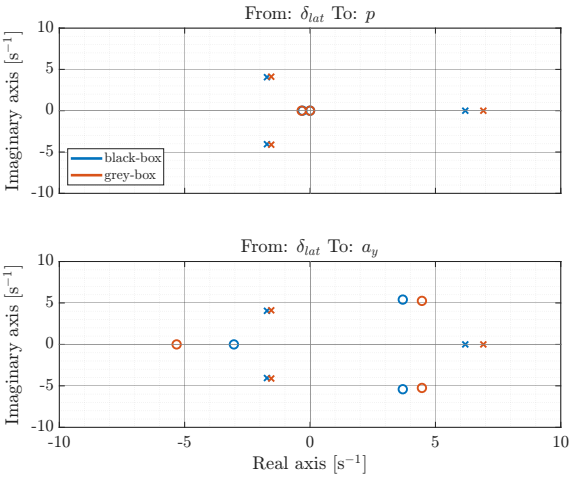
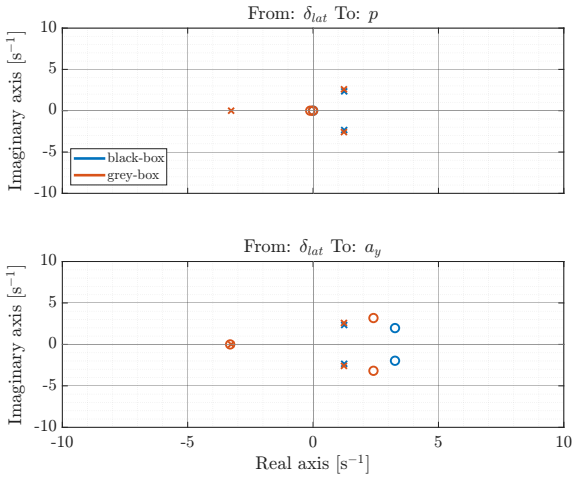


Figure 5.33: Frequency response function of the black-box and frequency-domain OE grey-box model at 40% power.

The pole-zero maps obtained with the frequency-domain OE algorithm are presented in Figures 5.34–5.38: there is a clear correspondence between the poles and zeros of the structured and unstructured models, with the largest differences in the models of Figure 5.35.

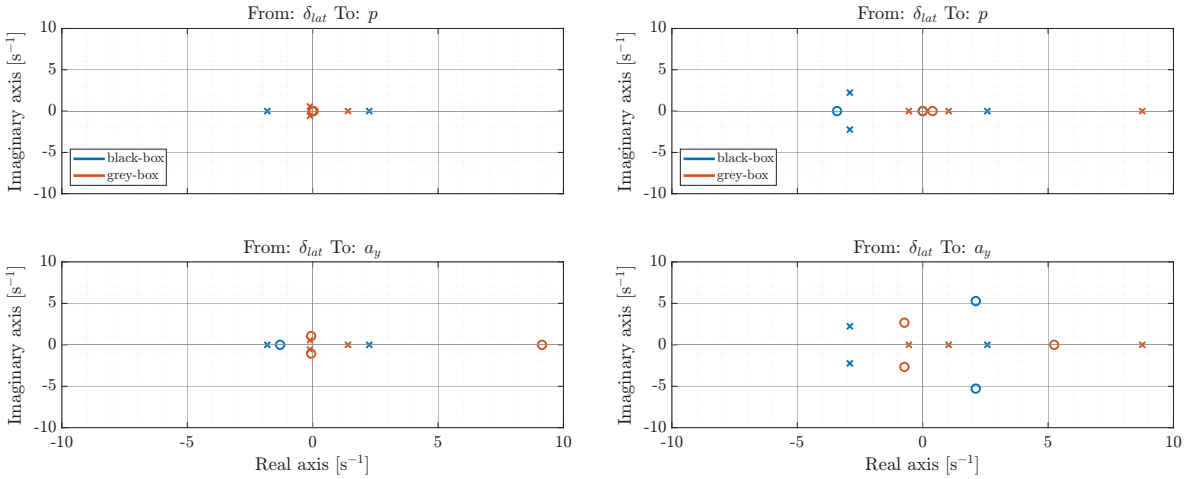


(a) 0% power.



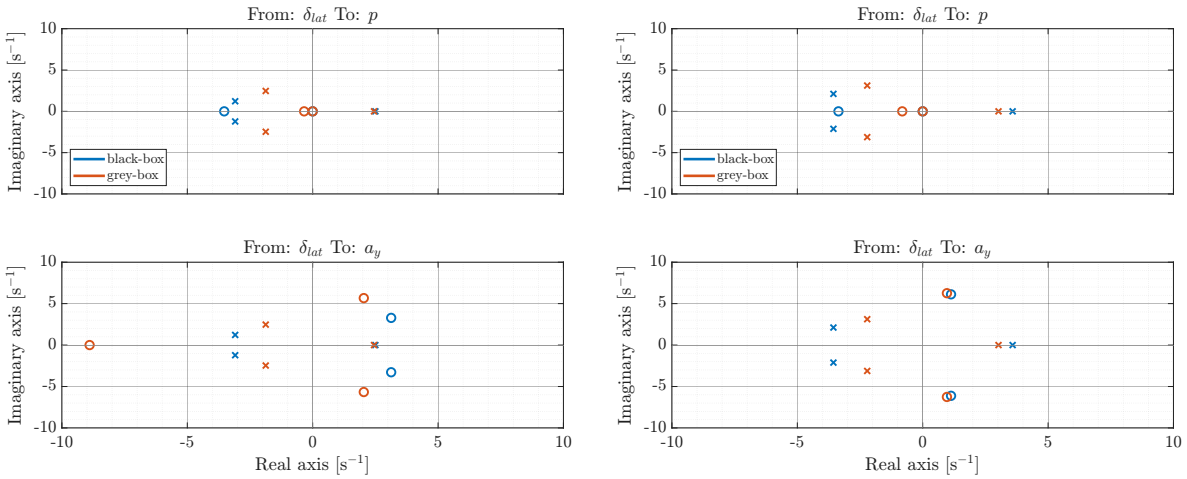
(b) 5% power.

Figure 5.34: Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 0% and 5% power.



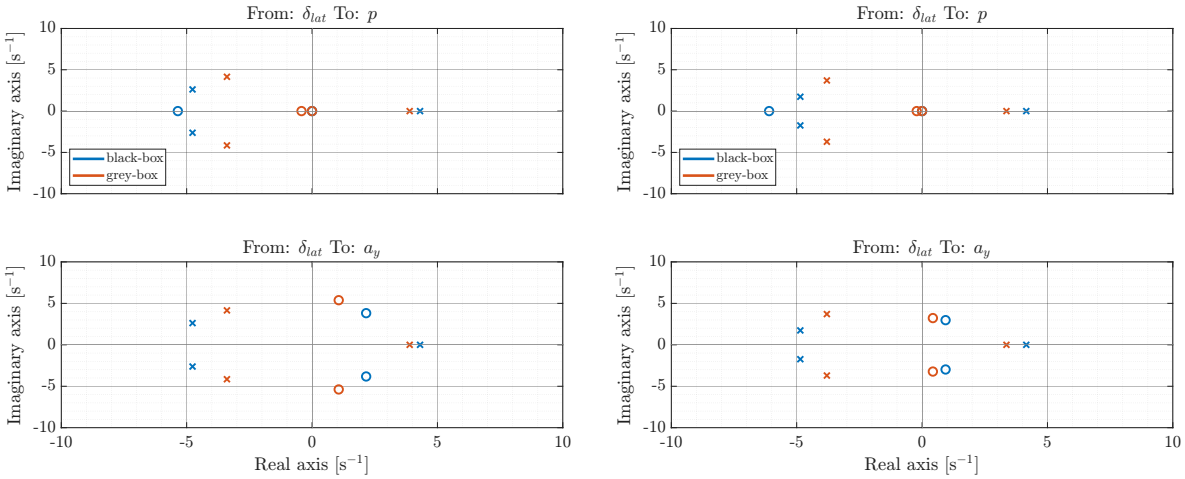
(a) 10% power. (b) 15% power.

Figure 5.35: Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 10% and 15% power.



(a) 20% power. (b) 25% power.

Figure 5.36: Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 20% and 25% power.



(a) 30% power. (b) 35% power.

Figure 5.37: Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 30% and 35% power.

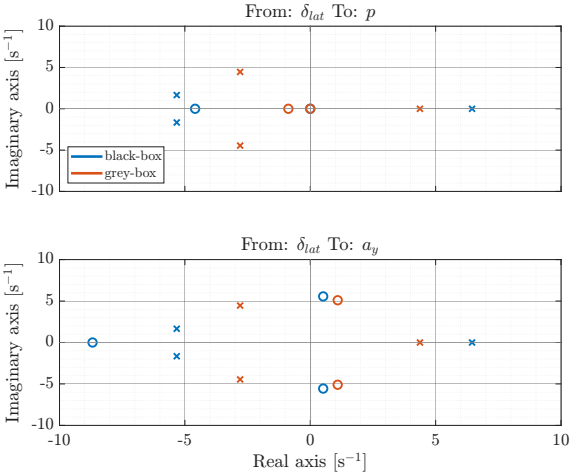


Figure 5.38: Pole-zero map comparison between the black-box and frequency-domain OE grey-box model at 40% power.

5.4. Results

This section presents the results obtained from the structured identification procedures introduced in the previous sections. The analysis is focused on the estimation of the six physical parameters defining the lateral grey-box model, namely the stability and control derivatives collected in Equation (5.1). The purpose is twofold: first, to compare the nominal parameter values obtained with the $\mathcal{H}_\infty$ model-matching approach and with the

frequency-domain Output-Error minimization procedure; second, to assess the reliability of the estimated parameters through a Monte Carlo analysis.

5.4.1. Estimated parameters

The nominal values of the physical parameters estimated by the two structured identification procedures are reported in the following tables.

Table 5.2 shows the results obtained with the $\mathcal{H}_\infty$ model-matching algorithm. In agreement with what was observed in Section 5.3.1, the estimate at 5% WindShaper power appears to be poorly conditioned, as indicated by the unusually large values of Y_p, L_p, Y_δ, and L_δ. Because of this, the parameters obtained at 0% power were used as initial guesses for the structuring procedure at 10% power. The other estimated parameters show an overall reasonable trend across the tested operating conditions: the derivatives Y_v, L_v, L_p, and L_δ retain a consistent sign over most of the power levels; furthermore, their magnitudes remain generally compatible with the expected order of the lateral dynamics parameters. Conversely, Y_p and Y_δ exhibit less regular trends, including some sign changes, and should therefore be interpreted with more caution. This suggests that the nominal $\mathcal{H}_\infty$ estimates are broadly coherent, although the identifiability of some parameters may still be limited.

Power [%]	Y_v **[1/s]**	Y_p **[m/(s rad)]**	L_v **[rad/(s m)]**	L_p **[1/s]**	Y_δ **[m/s^2]**	L_δ **[rad/s^2]**
0	-0.5566	-0.0391	15.6797	5.2184	-19.0071	1497.2508
5	-0.0005	-1.9596	0.0463	-66.8866	-596.8098	-17283.8229
10	-0.0095	0.2157	0.0783	0.4069	-105.5888	307.0371
15	-0.5165	-0.1455	1.1460	0.3597	9.4734	581.8825
20	-0.1830	0.0573	0.4885	0.8811	5.1408	429.1132
25	-1.0256	-0.4025	3.0498	1.3430	1.4202	533.6211
30	-0.5464	-0.2772	3.1112	2.4605	-24.2046	485.5912
35	-0.3049	-0.2806	1.6227	1.9350	-19.4139	386.2742
40	-1.5360	-0.6900	9.8218	5.0196	-14.2390	499.1871

Table 5.2: Nominal structured parameter estimates for different WindShaper power levels, considering the SIMO PBSID$_{opt}$-$\mathcal{H}_\infty$ grey-box model with input δ_{lat} and outputs p and a_y.

Table 5.3 shows the parameter estimates obtained using the frequency-domain OE approach: some derivatives, such as Y_v, Y_δ, and L_δ, exhibit a more consistent sign across the different WindShaper power levels, while their magnitudes remain generally bounded.

On the other hand, Y_p and L_p still present sign changes and non-monotonic variations, suggesting that these parameters are more sensitive to the chosen optimization criterion and to the information content of the selected black-box models.

Power [%]	Y_v [1/s]	Y_p [m/(s rad)]	L_v [rad/(s m)]	L_p [1/s]	Y_δ [m/s^2]	L_δ [rad/s^2]
0	-0.5113	-0.0094	13.6150	4.3256	-18.4321	1419.5823
5	-0.0559	0.0479	-2.7426	-0.7491	-28.8352	1357.7203
10	0.0353	0.2775	0.0472	1.1632	-31.6779	897.4638
15	0.4426	-0.4787	-0.5124	8.7950	-114.0371	1193.7200
20	-0.3651	-0.0398	2.4221	-0.9312	-11.0863	1083.5023
25	-0.8592	-0.1729	4.5844	-0.5423	-8.4807	895.9575
30	-0.5317	-0.0916	11.6781	-2.3631	-11.9401	1259.6838
35	-0.3700	-0.1185	9.9680	-3.8590	-18.2250	1166.1508
40	-1.1876	-0.2487	12.8702	-0.0323	-16.4233	673.7800

Table 5.3: Nominal structured parameter estimates for different WindShaper power levels, considering the SIMO PBSID$_{opt}$-OE grey-box model with input δ_{lat} and outputs p and a_y.

Overall, the OE estimates do not show the same isolated anomaly observed in the $\mathcal{H}_\infty$ result at 5% power. Nevertheless, the nominal comparison alone is not sufficient to conclude that one method provides more reliable physical parameters. For this reason, the Monte Carlo analysis presented in the following section is used to assess the dispersion and robustness of the estimates obtained with both structured identification procedures.

5.4.2. Monte Carlo analysis

The uncertainty of the structured estimates is evaluated through a Monte Carlo analysis, following the procedure adopted in [13]. The method exploits the asymptotic covariance of the selected PBSID$_{opt}$ black-box model. In particular, the covariance matrix associated with the estimated state-space matrices is computed using the PBSID Toolbox function `dvar4abck`, as discussed in [52]. Starting from this covariance matrix, a set of randomly perturbed black-box models is generated by assuming a multivariate normal distribution of the estimated matrices.

Each randomly generated model is then used as a new unstructured reference model for the structured identification procedure. Therefore, for each Monte Carlo sample, the grey-box model-matching problem is solved again and a new value of the parameter vector ϑ_{lat} is obtained. In this way, an empirical distribution is built for each stability and control

derivative. The same settings used for the nominal identification are kept fixed during the Monte Carlo analysis, including the selected black-box model, the frequency range used for model matching, the optimization options, and the initial parameter values. This ensures that the resulting dispersion is associated with the uncertainty of the underlying PBSID$_{opt}$ model rather than with a different tuning of the structured optimization problem.

Possible outliers, which are defined as elements that are more than 1.5 interquartile ranges above the upper quartile (75%) or below the lower quartile (25%), are removed using the InterQuartile Range (IQR) criterion: a Monte Carlo realization is rejected if at least one component of the corresponding parameter vector is classified as an outlier. After outlier removal, the mean value and standard deviation of each Monte Carlo distribution are computed. For a generic parameter vector sample $\hat{\vartheta}_{lat}^{(k)}$ with $k = 1, \ldots, N_{MC}$, the sample mean is evaluated as

$$\bar{\vartheta}_{lat} = \frac{1}{N_{MC}} \sum_{k=1}^{N_{MC}} \hat{\vartheta}_{lat}^{(k)}, \tag{5.3}$$

while the sample variance is

$$\sigma_{\vartheta}^2 = \frac{1}{N_{MC} - 1} \sum_{k=1}^{N_{MC}} \left(\hat{\vartheta}_{lat}^{(k)} - \bar{\vartheta}_{lat} \right)^2. \tag{5.4}$$

These quantities are reported together with the nominal parameter estimates obtained in the previous subsection. The comparison between the nominal values, the Monte Carlo mean values, and the corresponding standard deviations provides an indication of the reliability of the identified physical parameters. Figures 5.39–5.44 show the results obtained with the $\mathcal{H}_\infty$ model matching, while the results for the frequency-domain OE method are shown in Figures 5.45–5.50.

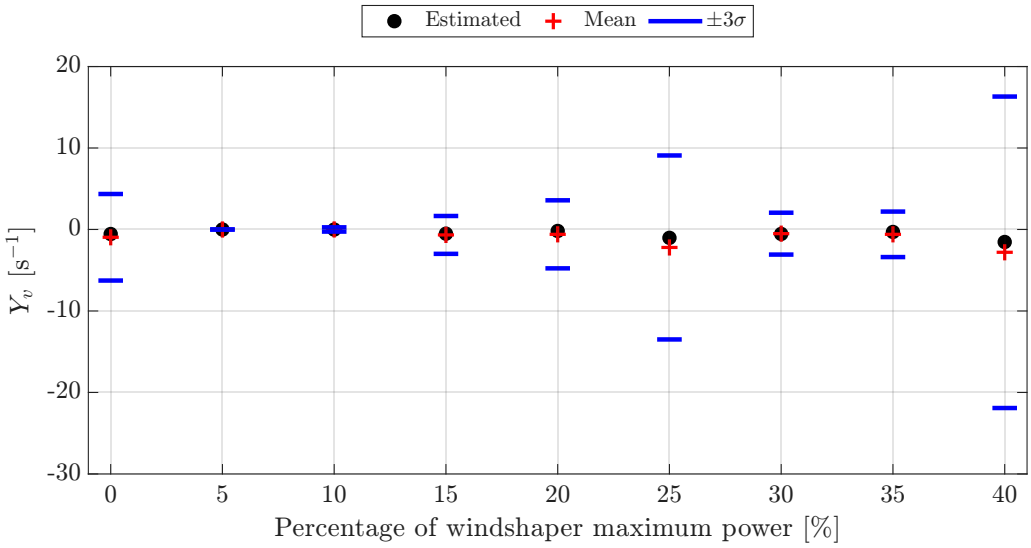


Figure 5.39: Monte Carlo distribution of the parameter Y_v estimated with the $\mathcal{H}_\infty$ model-matching approach.

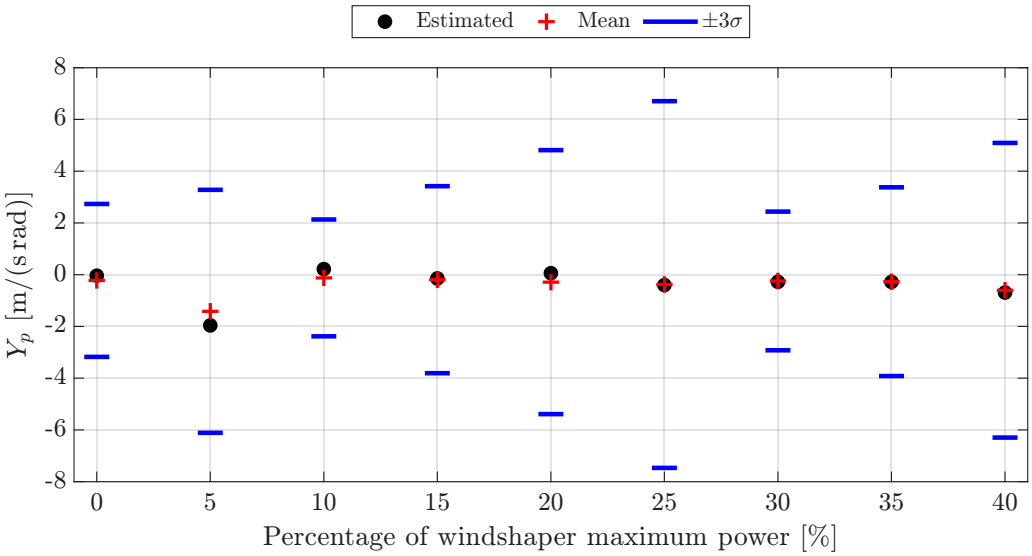


Figure 5.40: Monte Carlo distribution of the parameter Y_p estimated with the $\mathcal{H}_\infty$ model-matching approach.

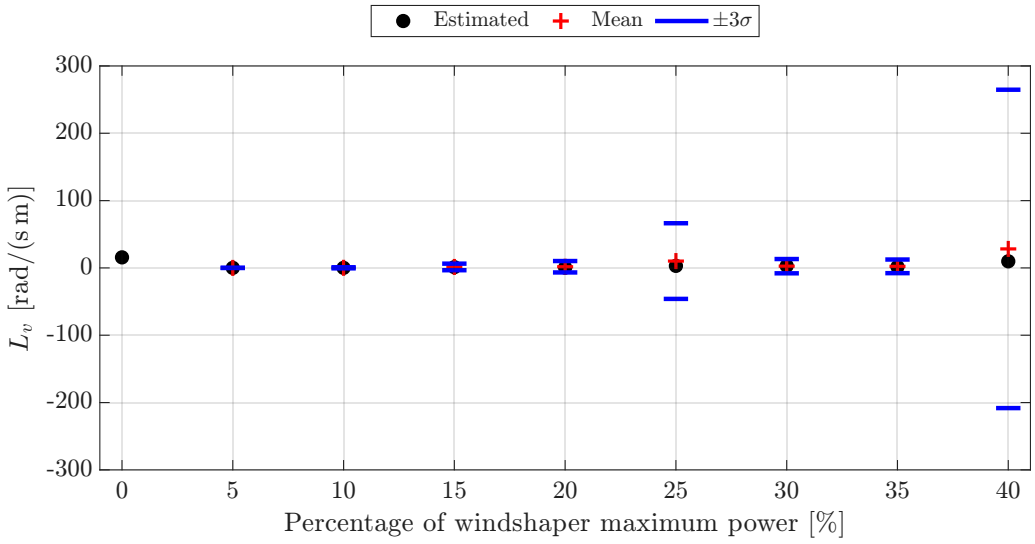


Figure 5.41: Monte Carlo distribution of the parameter L_v estimated with the $\mathcal{H}_\infty$ model-matching approach.

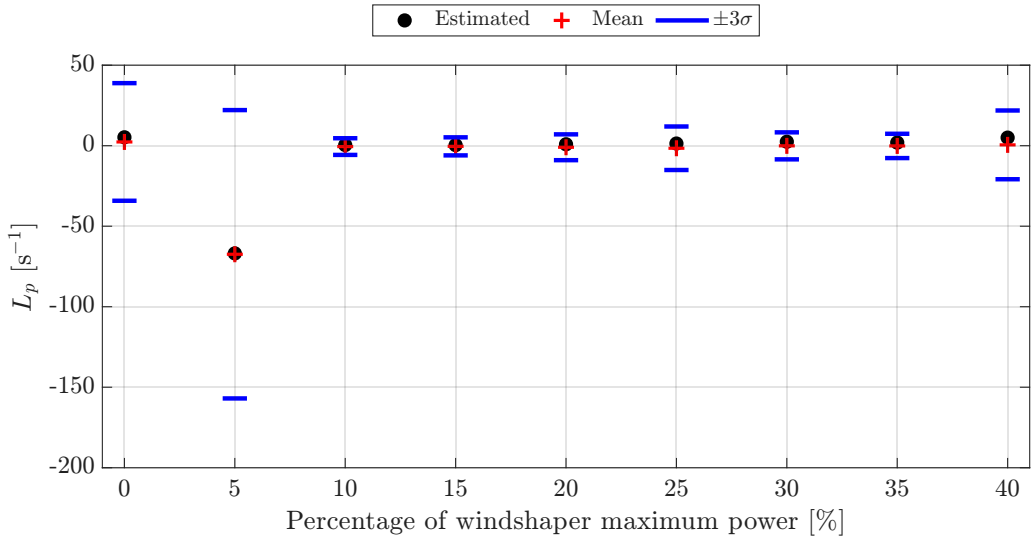


Figure 5.42: Monte Carlo distribution of the parameter L_p estimated with the $\mathcal{H}_\infty$ model-matching approach.

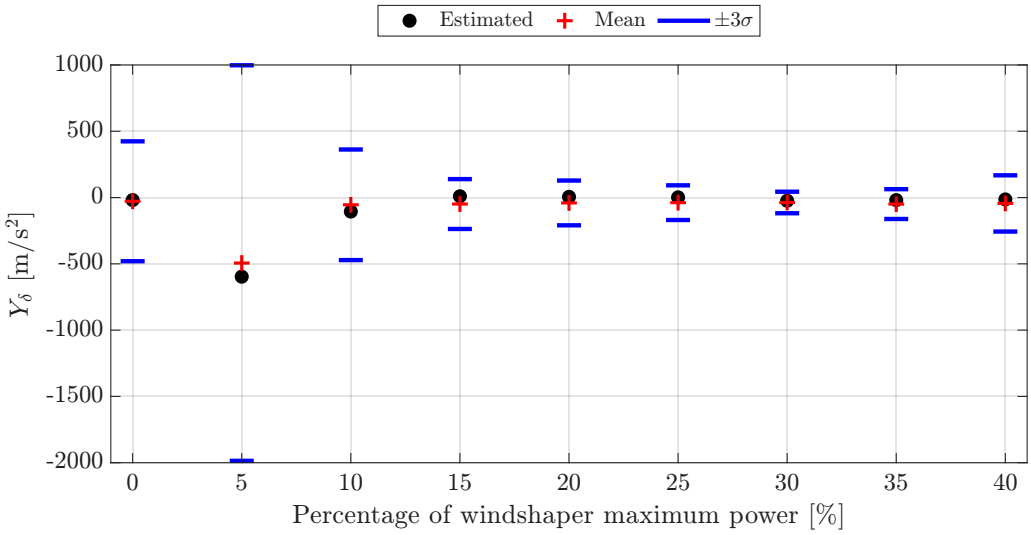


Figure 5.43: Monte Carlo distribution of the parameter Y_δ estimated with the $\mathcal{H}_\infty$ model-matching approach.

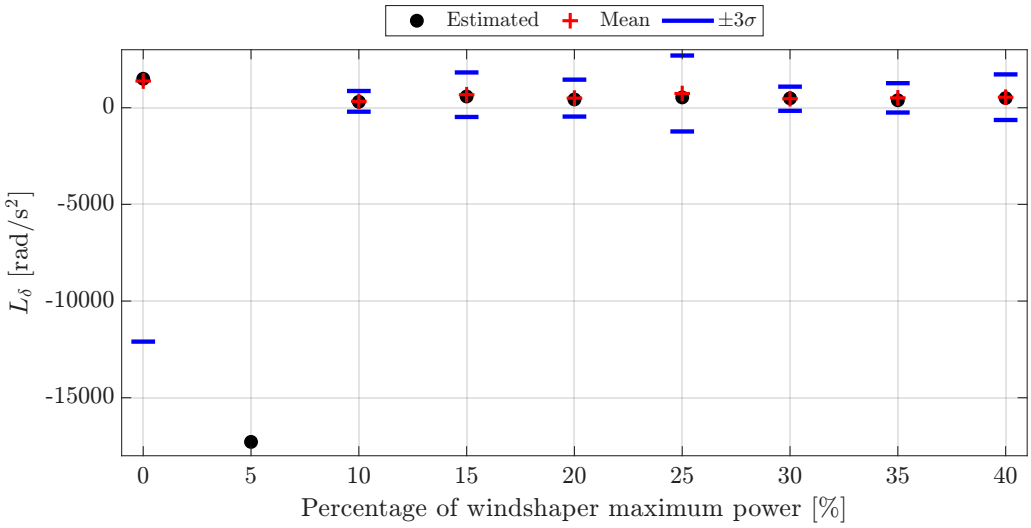


Figure 5.44: Monte Carlo distribution of the parameter L_δ estimated with the $\mathcal{H}_\infty$ model-matching approach.

Estimates obtained with $\mathcal{H}_\infty$ model matching show that, especially for some parameters, the results obtained are not robust. In particular, the estimate of L_v at 0% (Figure 5.41), and those of L_δ at 0% and 5% WindShaper power (Figure 5.44) have large 3σ confidence intervals, which are not shown in the figure in order to make the rest of the nominal estimates more readable.

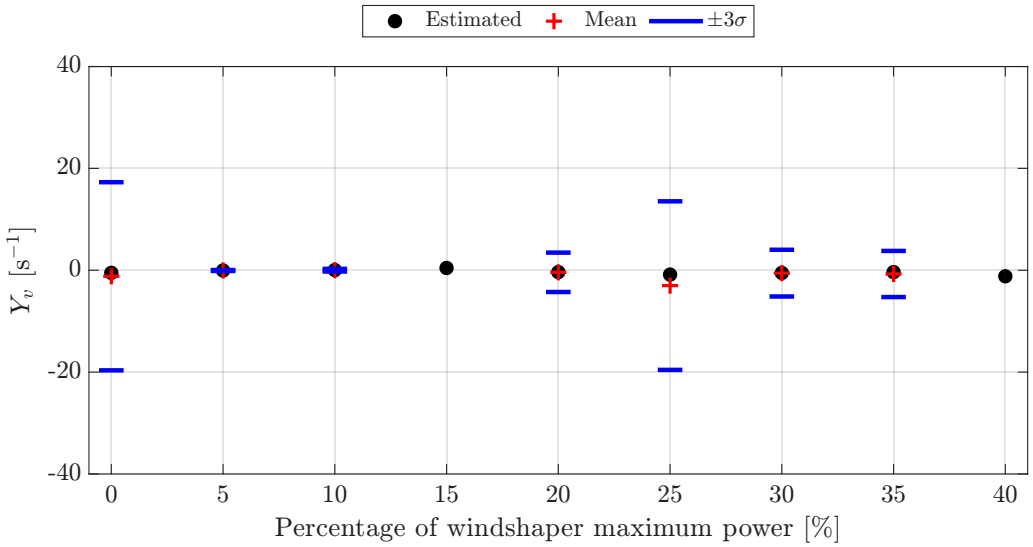


Figure 5.45: Monte Carlo distribution of the parameter Y_v estimated with the frequency-domain Output-Error minimization procedure.

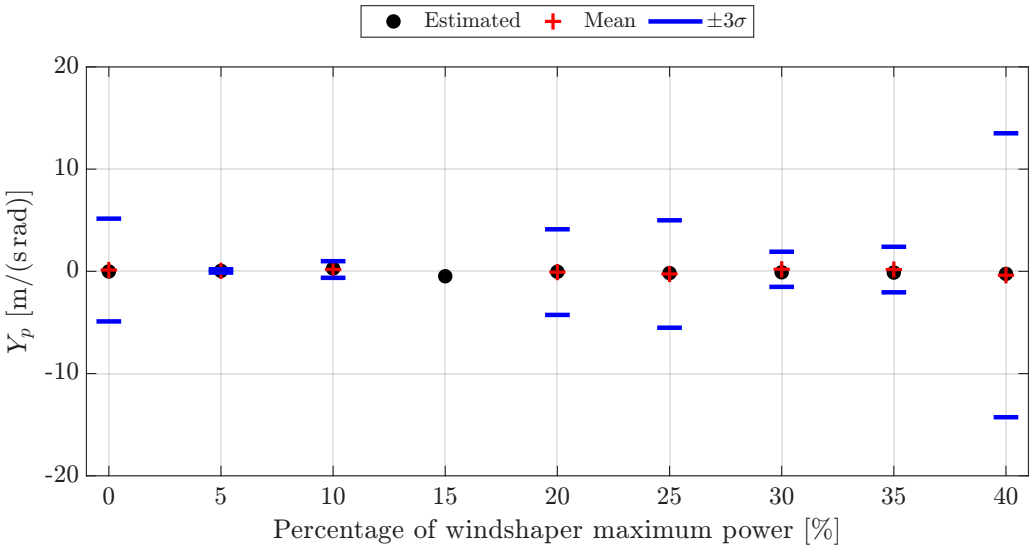


Figure 5.46: Monte Carlo distribution of the parameter Y_p estimated with the frequency-domain Output-Error minimization procedure.

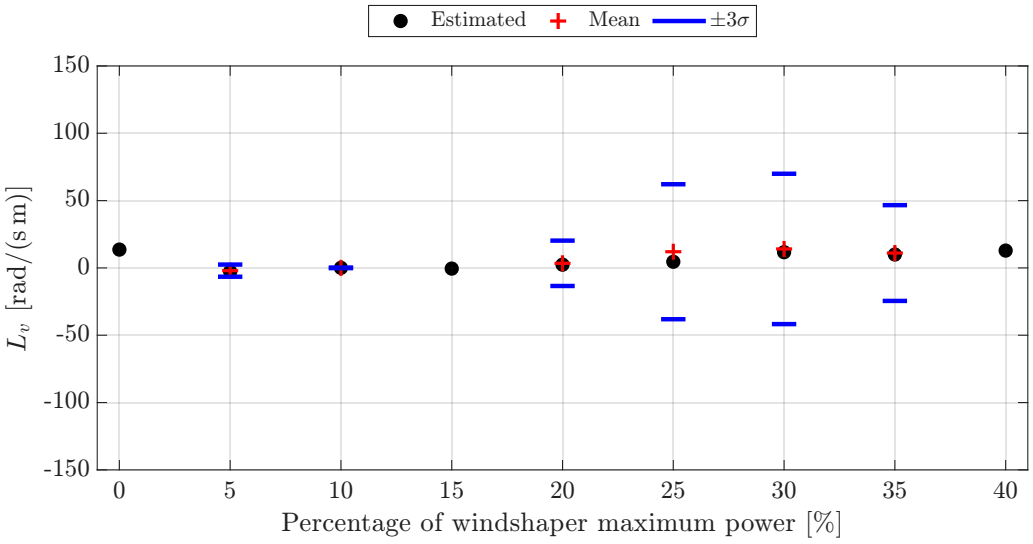


Figure 5.47: Monte Carlo distribution of the parameter L_v estimated with the frequency-domain Output-Error minimization procedure.

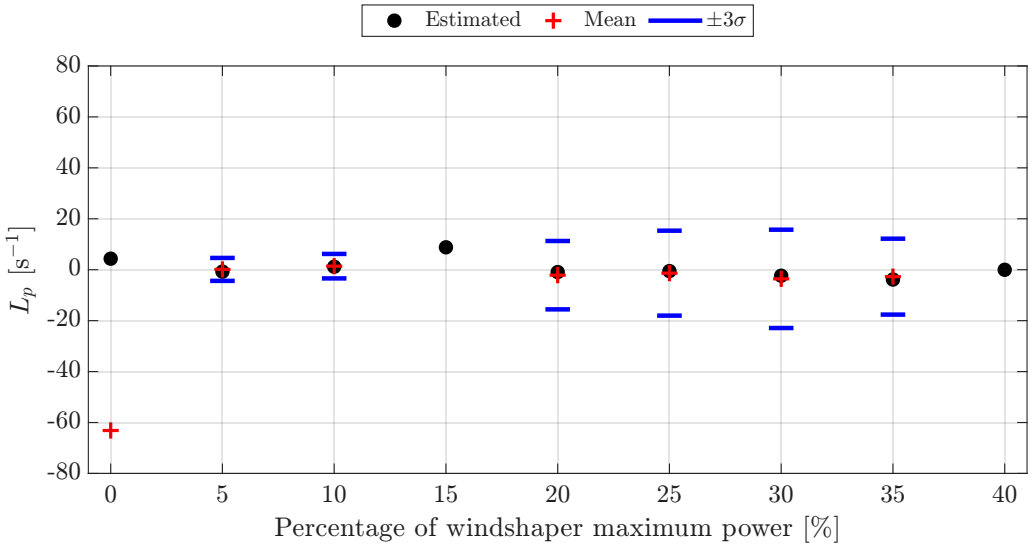


Figure 5.48: Monte Carlo distribution of the parameter L_p estimated with the frequency-domain Output-Error minimization procedure.

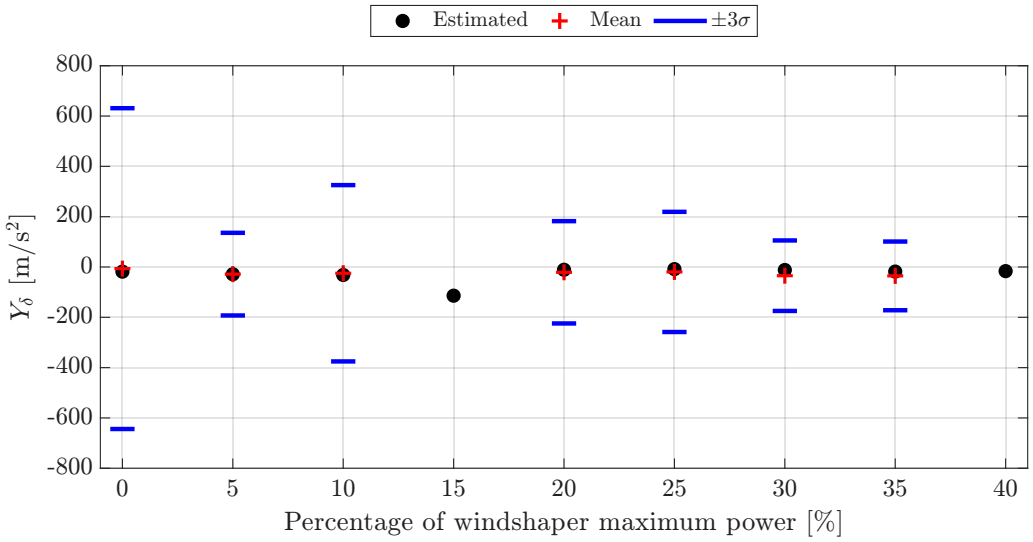


Figure 5.49: Monte Carlo distribution of the parameter Y_δ estimated with the frequency-domain Output-Error minimization procedure.

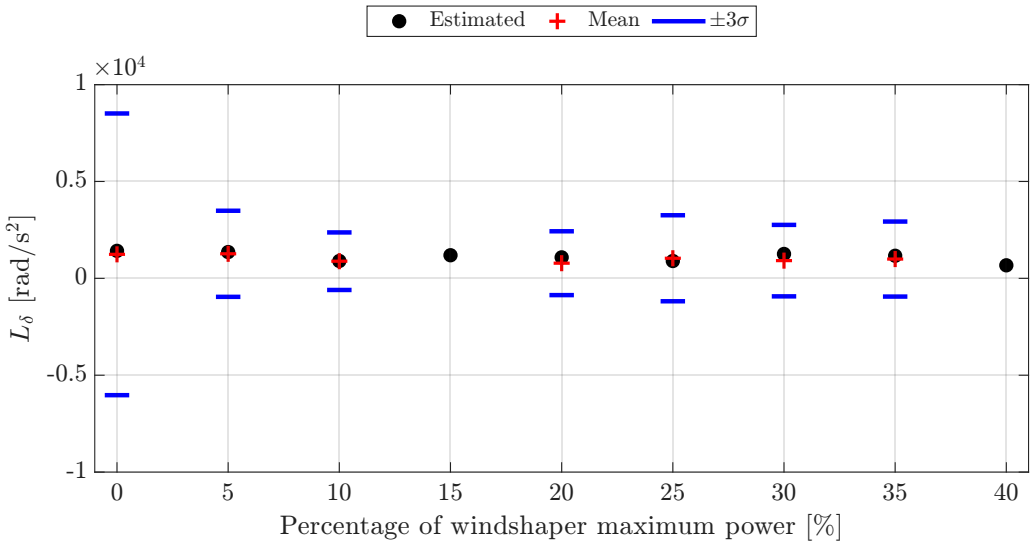


Figure 5.50: Monte Carlo distribution of the parameter L_δ estimated with the frequency-domain Output-Error minimization procedure.

In the case of the results obtained with the frequency-domain OE minimization, the estimates at 15% and 40% power show the worst performance in terms of statistical robustness, once again with 3σ confidence intervals which were too large to be properly displayed in Figures 5.45–5.50.

The Monte Carlo analysis shows that neither structured identification procedure provides uniformly robust parameter estimates over the whole range of WindShaper power levels.

In several cases, the 3σ confidence intervals are large when compared with the corresponding nominal values, and in some plots the vertical scale had to be enlarged in order to make the nominal estimates readable.
Several factors may explain the limited robustness observed in the Monte Carlo distributions: first of all, given the intrinsically non-convex nature of the model-matching algorithms, small perturbations of the black-box PBSID_{opt} model may lead the optimization toward different local minima. The uncertainty of the underlying black-box model also plays an important role: since the Monte Carlo samples are generated by perturbing the selected PBSID_{opt} model according to its estimated covariance, if the black-box frequency response is uncertain in some frequency regions, this uncertainty is propagated and possibly amplified by the structured optimization step. Large confidence intervals in the physical parameters therefore indicate that the grey-box parametrization is sensitive to the uncertainty of the unstructured estimate.
This result does not necessarily imply that the corresponding grey-box model is unable to reproduce the input-output behavior, but rather that different combinations of physical parameters may provide similar model-matching performance. Hence, the Monte Carlo analysis highlights a lack of robustness in the recovery of the physical parametrization, especially for the parameters that are weakly observable from the available input-output data.

5.5. Comparison between $\mathcal{H}_\infty$ model matching and frequency-domain Output-Error minimization

The results presented in the previous sections show that the two structured model-matching algorithms lead to different outcomes: although both methods use the same black-box PBSID_{opt} models and the same physical parametrization, the different optimization criteria lead to different compromises between the two output channels and, consequently, to different nominal parameter estimates.

From the cross-validation results reported in Table 5.1, the frequency-domain OE approach provides, on average, better time-domain validation performance than the $\mathcal{H}_\infty$ model matching. In particular, the $\mathcal{H}_\infty$ estimates show a strong degradation at low and intermediate WindShaper power levels: the model at 5% power fails to reproduce both output channels, while the models at 10–25% power are unable to reproduce the lateral acceleration a_y. Conversely, the OE method gives a more regular behavior on the roll-rate channel p and generally improves the a_y validation with respect to the $\mathcal{H}_\infty$ approach, especially from 20% power onward. The only relevant exceptions are the 10% and 15%

models, for which the OE model also fails to capture the lateral acceleration response.

The Monte Carlo analysis on the parameter estimates, however, shows that neither method is robust in the estimate of $\hat{\vartheta}^{\star}$: therefore, the OE method improves the nominal validation performance, but it does not remove the uncertainty affecting the recovery of the physical parameters. This behavior suggests that the limiting factor is not only the numerical optimization algorithm, but also the information content of the black-box models with respect to the chosen grey-box parametrization.

For these reasons, the frequency-domain OE method can be considered the preferable approach in this work, but only in a practical validation sense: it provides better time-domain cross-validation results, while giving more regular nominal parameter trends for some of the derivatives. However, this does not imply that the OE estimates are fully reliable as physical parameter estimates; the most reasonable conclusion is therefore that frequency-domain OE is the better-performing model-matching strategy for reproducing the input-output behavior of the selected PBSID_{opt} models, whereas neither method provides a completely robust recovery of the lateral-dynamics physical parametrization.

Conclusions and further developments

The aim of this thesis was to investigate different procedures for the structured identification of the lateral dynamics of a multirotor UAV, while at the same time refining the experimental procedure defined in the previous works [13] to obtain reliable black-box models under different operating conditions. A physically parametrized grey-box model, suitable for the identification task, was first developed starting from the linearized dynamics equations and then decoupling the resulting model in order to study the SIMO lateral dynamics. An experimental campaign was carried out at different flight speeds in order to gather the necessary input–output data to be used in the PBSID$_{opt}$ algorithm, which provided accurate black-box models to be used during the structuring procedures. Two structuring procedures were applied to the estimated black-box models: $\mathcal{H}_\infty$ model matching, which formulates the problem as the minimization of the worst-case frequency-domain discrepancy between the unstructured and the structured model, and frequency-domain Output-Error minimization, which minimizes a sampled frequency-domain discrepancy over a finite grid of frequencies.

The models identified via PBSID$_{opt}$ achieved high cross-validation scores at all WindShaper powers, with the exception of the 5% power case. The lower performance at this power level is likely related to the flow generated by the WindShaper, which may act more as an external aerodynamic disturbance than as a sufficiently coherent excitation of the lateral dynamics. The comparison between the two structuring procedures showed that the Output-Error minimization generally produced more promising results in terms of matching with the black-box frequency response. Therefore, within the present implementation and for the available datasets, the frequency-domain Output-Error approach appears to be the more effective of the two structuring procedures. However, the Monte Carlo analysis showed that the structured estimates are not yet sufficiently robust. A plausible explanation is that the structured models possess a strong sensitivity to the uncertainty inherited from the PBSID$_{opt}$ step, especially in frequency ranges where the excitation level and the input–output coherence are lower. This result is important be-

cause it prevents the structured models from being interpreted as final, reliable estimates of the true physical derivatives.

Overall, the thesis confirms that the adopted identification pipeline is technically feasible; at the same time, the results indicate that the present experimental and numerical setup is not yet sufficient to obtain robust parameter estimates across all WindShaper power levels. The main conclusion is therefore twofold. On one hand, the proposed procedure is a valid basis for the structured identification of multirotor UAV dynamics. On the other hand, the robustness of the grey-box estimates is strongly dependent on the quality and uncertainty of the preliminary black-box identification step, which must become a central criterion in future model selection. This could be done by finding additional metrics to quantify the uncertainty of the unstructured estimate, to be used together with the VAF index when evaluating and selecting the black-box models to be used in the structured identification.

Furthermore, the quality of the structured identification may ultimately be limited by the information content of the data. A wider and more structured set of experiments should be performed, considering different excitation signals and amplitudes, with the explicit objective of improving the input–output coherence over the frequency range relevant for lateral rigid-body dynamics. This is particularly important at low frequencies, where the effect of aerodynamic disturbances and position-control action can mask the response of the lateral dynamics. In this respect, the low-power results, especially the 5% WindShaper case, should be interpreted with caution. Available flow-field characterization data at the UAV location could be used to better relate the imposed WindShaper power to the actual wind velocity experienced by the vehicle. This could help determine whether, at 5% power, the aerodynamic effects induced by the WindShaper are indeed comparable to other disturbances and to the action of the feedback controller, which could explain the reduced coherence and the lower identification performance observed at this operating point.

Bibliography

[1] P. van Overschee and B. De Moor. *Subspace identification for linear systems: theory, implementation, applications*. Kluwer Academic Publishers, Boston, MA, 1996.

[2] B. L. Ho and R. E. Kalman. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 14(12):545–548, 1966.

[3] P. van Overschee and B. De Moor. N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[4] M. Verhaegen. Identification of the deterministic part of MIMO state-space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

[5] G. J. van der Veen, J. W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen. Closed-loop subspace identification methods: an overview. *IET Control Theory & Applications*, 7(10):1339–1358, 2013.

[6] A. Chiuso and G. Picci. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.

[7] A. Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034–1048, 2007.

[8] A. Chiuso. On the relation between CCA and predictor-based subspace identification. *IEEE Transactions on Automatic Control*, 52(10):1795–1812, 2007.

[9] J. Wartmann and S. Seher-Weiss. Application of the predictor-based subspace identification method to rotorcraft system identification. In *Proceedings of the 39th European Rotorcraft Forum*, September 2013.

[10] M. Bergamasco and M. Lovera. Identification of linear models for the dynamics of a hovering quadrotor. *IEEE Transactions on Control Systems Technology*, 22(5): 1696–1707, 2014.

[11] M. Wu, M. Lovera, and M. Z. Q. Chen. Application of PBSID and structured H_∞ methods in unmanned helicopter system identification. In *2019 IEEE International*

Conference on Unmanned Systems and Artificial Intelligence (ICUSAI), pages 136–141, 2019.

[12] G. Fumai. Accuracy analysis of PBSID$_{\text{opt}}$–H_∞ for multirotor UAVs model identification. Master's thesis, Politecnico di Milano, Milan, Italy, 2020.

[13] N. Dell'Aera. Multirotor UAV model identification under varying flight conditions. Master's thesis, Politecnico di Milano, Milan, Italy, 2025. Master of Science in Aeronautical Engineering.

[14] I. Houtzager. *Towards data-driven control for modern wind turbines*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2011.

[15] M. Bergamasco and M. Lovera. State-space model identification: from unstructured to structured models with an H_∞ approach. In *Proceedings of the 5th IFAC Symposium on System Structure and Control*, volume 46 of *IFAC Proceedings Volumes*, pages 202–207, 2013.

[16] M. Bergamasco, A. Ragazzi, and M. Lovera. Rotorcraft system identification: a time/frequency-domain approach. In *Proceedings of the 19th IFAC World Congress*, pages 8861–8866, Cape Town, South Africa, 2014.

[17] M. Bergamasco and M. Lovera. Continuous-time predictor-based subspace identification using Laguerre filters. *IET Control Theory & Applications*, 5(7):856–867, 2011.

[18] M. Bergamasco, M. Sguanci, and M. Lovera. Continuous-time model identification for rotorcraft dynamics. *IFAC Proceedings Volumes*, 45(16):816–821, 2012.

[19] D. Vizer, G. Mercère, O. Prot, and E. Laroche. H_∞-norm-based optimization for the identification of gray-box LTI state-space model parameters. *Systems & Control Letters*, 92:34–41, 2016.

[20] J. Schoukens, R. Pintelon, and Y. Rolain. Time-domain identification, frequency-domain identification: equivalencies! differences? In *Proceedings of the 2004 American Control Conference*, Boston, MA, USA, 2004. IEEE.

[21] M. B. Tischler and R. K. Remple. *Aircraft and rotorcraft system identification: engineering methods with flight-test examples*. American Institute of Aeronautics and Astronautics, 2006.

[22] E. A. Morelli. System identification programs for aircraft (SIDPAC). In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Monterey, CA, 2002.

[23] E. A. Morelli and J. A. Grauer. Practical aspects of frequency-domain approaches for aircraft system identification. *Journal of Aircraft*, 57(2):268–291, 2020.

[24] W. Yuan and J. Katupitiya. A grey-box system identification procedure for scale model helicopters based on frequency-domain estimation methods. In *Proceedings of the 28th International Congress of the Aeronautical Sciences*, Brisbane, Australia, September 2012.

[25] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, New York, 3 edition, 2015.

[26] A. Barjau Condomines and J. A. Batlle. *Rigid body kinematics*. Cambridge University Press, Cambridge, 2020.

[27] M. Lovera. *Subspace identification methods: theory and applications*. Ph.d. thesis, Politecnico di Milano, Milan, Italy, 1997.

[28] P. van Overschee and B. De Moor. Closed-loop subspace system identification. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 1848–1853, San Diego, CA, USA, 1997. IEEE.

[29] M. Verhaegen and V. Verdult. *Filtering and system identification: a least squares approach*. Cambridge University Press, 2007.

[30] J.-W. van Wingerden, I. Houtzager, P. M. O. Gebraad, and M. Verhaegen. Predictor-based subspace identification toolbox, version 0.6. Delft University of Technology, 2015. Released 18-Feb-2015.

[31] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston and Sons, Washington, DC, 1977.

[32] P. C. Hansen. Regularization tools: a MATLAB package for analysis and solution of discrete ill-posed problems, version 4.1. Technical report, Technical University of Denmark, 2008.

[33] L. Ljung. *System identification: theory for the user*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1999.

[34] C. Yu, L. Ljung, and M. Verhaegen. Identification of structured state-space models. *Automatica*, 90:54–61, 2018.

[35] P. A. Parrilo and L. Ljung. Initialization of physical parameter estimates. In *Pro-*

ceedings of the 13th IFAC Symposium on System Identification, volume 36 of *IFAC Proceedings Volumes*, pages 1483–1488, Rotterdam, The Netherlands, 2003. Elsevier.

[36] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

[37] P. Gahinet and P. Apkarian. Structured H_∞ synthesis in MATLAB. In *Proceedings of the 18th IFAC World Congress*, volume 44 of *IFAC Proceedings Volumes*, pages 1435–1440, 2011.

[38] L. Ljung. State of the art in linear system identification: time and frequency domain methods. In *Proceedings of the American Control Conference*, 2004.

[39] R. Pintelon and J. Schoukens. *System identification: a frequency domain approach*. Wiley-IEEE Press, 2 edition, 2012.

[40] The MathWorks, Inc. Estimating models using frequency-domain data. MATLAB documentation, 2026.

[41] The MathWorks, Inc. greyest: estimate ODE parameters of linear grey-box model. MATLAB documentation, 2026.

[42] The MathWorks, Inc. greyestOptions: option set for greyest. MATLAB documentation, 2026.

[43] L. Ljung. Estimation focus in system identification: prefiltering, noise models, and prediction. In *Proceedings of the 38th IEEE Conference on Decision and Control*, 1999.

[44] J. G. Proakis and D. G. Manolakis. *Digital signal processing: principles, algorithms, and applications*. Pearson/Prentice Hall, Upper Saddle River, NJ, 4 edition, 2006.

[45] WindShape SA. WindShape SA. Company website, 2025. Ch. de l'Émeraude 10, 1214 Vernier, Switzerland.

[46] F. Noca, G. Catry, N. Bosson, L. J. Bardazzi, S. Márquez, and A. Gros. Wind and weather facility for testing free-flying drones. In *AIAA Aviation Forum*, Dallas, TX, 2019. American Institute of Aeronautics and Astronautics. WindShape, Geneva, Switzerland.

[47] F. Noca, T. Bujard, G. Visvaratnam, G. Catry, and N. Bosson. Flow profiling in a WindShaper for testing free-flying drones in adverse winds. In *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, 2021. HES-SO University of Applied Sciences and Arts Western Switzerland and WindShape.

[48] F. Noca, T. Bujard, G. Visvaratnam, G. Catry, and N. Bosson. Free-flight testing of UAVs in a WindShape facility with GNSS-based tracking. In *AIAA SciTech Forum*. American Institute of Aeronautics and Astronautics, 2021.

[49] T. Fiorucci, A. Walpen, N. Bosson, G. Catry, and F. Noca. UAV takeoff, hover and landing test in real-world flight conditions within a laboratory setting. In *AIAA SciTech Forum*, Orlando, FL, January 2025. American Institute of Aeronautics and Astronautics.

[50] Y. Wi and M. Cescon. Modeling and identification of quadrotor dynamics affected by wind stress. *IFAC-PapersOnLine*, 58(28):7–12, 2024.

[51] P. Welch. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.

[52] J. W. van Wingerden. The asymptotic variance of the $\text{PBSID}_{\text{opt}}$ algorithm. *IFAC Proceedings Volumes*, 45(16):1167–1172, 2012.

List of Figures

1.1 Quadrotor kinematics [12]. 6

2.1 Standard form for structured $\mathcal{H}_\infty$ synthesis. 25
2.2 Model matching structure and equivalent standard form. 26

3.1 Drone in free flight in the WindShaper wind field, from [49]. 32
3.2 Flowfield distribution downstream of the WindShaper, from [49]. 32
3.3 Turbulence distribution downstream of the WindShaper, from [49]. 32
3.4 ANT-X drone. 33
3.5 Cascade control structure. 36
3.6 Lateral dynamics control system. 37
3.7 Dataset δ_{lat}, p, a_y and excitation signal δ_{sweep} used for the identification of lateral dynamics at 20% of WindShaper power. 37
3.8 Dataset δ_{lat}, p, a_y and excitation signal δ_{sweep} used for the validation of lateral dynamics at 20% of WindShaper power. 38

4.1 Power spectral density of the outputs for varying WindShaper power. . . . 40
4.2 Input-output coherence between δ_{sweep} and y for varying WindShaper power. 40
4.3 Comparison of input-output coherence between δ_{sweep} and y at 5% and 10% of the previous experiments [13] and the current ones. 41
4.4 Power-Spectral-Density and coherence analysis for the lateral acceleration signal at 20% WindShaper power. 42
4.5 Raw and filtered a_y, with $\omega_c = 100$ rad/s. 43
4.6 Singular value decomposition, $ABCK$ model, $p = f = 50$; a gap can be spotted after the first five singular values. 44
4.7 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 0% WindShaper power. 47
4.8 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 5% WindShaper power. 47
4.9 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 10% WindShaper power. 48

4.10 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 15% WindShaper power. 48
4.11 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 20% WindShaper power. 49
4.12 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 25% WindShaper power. 49
4.13 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 30% WindShaper power. 50
4.14 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 35% WindShaper power. 50
4.15 Time-history of the measured signal y_{val} and the simulated signal $\hat{y}$ at 40% WindShaper power. 51
4.16 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 0% WindShaper power. 52
4.17 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 5% WindShaper power. 52
4.18 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 10% WindShaper power. 52
4.19 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 15% WindShaper power. 53
4.20 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 20% WindShaper power. 53
4.21 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 25% WindShaper power. 53
4.22 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 30% WindShaper power. 54
4.23 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 35% WindShaper power. 54
4.24 Comparison between the non-parametric FRF and the identified model via PBSID$_{opt}$ at 40% WindShaper power. 54

5.1 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 0% power. . . 60
5.2 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 5% power. . . 60
5.3 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 10% power. . . 60
5.4 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 15% power. . . 61
5.5 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 20% power. . . 61
5.6 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 25% power. . . 61

5.7 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 30% power. . . 62
5.8 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 35% power. . . 62
5.9 Time-domain cross-validation of the $\mathcal{H}_\infty$ grey-box model at 40% power. . . 62
5.10 Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 0% and 5% power. 63
5.11 Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 10% and 15% power. 64
5.12 Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 20% and 25% power. 64
5.13 Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box models at 30% and 35% power. 65
5.14 Frequency response function of the black-box and $\mathcal{H}_\infty$ grey-box model at 40% power. 65
5.15 Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 0% and 5% power. 66
5.16 Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 10% and 15% power. 66
5.17 Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 20% and 25% power. 67
5.18 Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box models at 30% and 35% power. 67
5.19 Pole-zero map comparison between the black-box and $\mathcal{H}_\infty$ grey-box model at 40% power. 68
5.20 Time-domain cross-validation of the OE grey-box model at 0% power. . . . 68
5.21 Time-domain cross-validation of the OE grey-box model at 5% power. . . . 69
5.22 Time-domain cross-validation of the OE grey-box model at 10% power. . . 69
5.23 Time-domain cross-validation of the OE grey-box model at 15% power. . . 69
5.24 Time-domain cross-validation of the OE grey-box model at 20% power. . . 70
5.25 Time-domain cross-validation of the OE grey-box model at 25% power. . . 70
5.26 Time-domain cross-validation of the OE grey-box model at 30% power. . . 70
5.27 Time-domain cross-validation of the OE grey-box model at 35% power. . . 71
5.28 Time-domain cross-validation of the OE grey-box model at 40% power. . . 71
5.29 Frequency response function of the black-box and frequency-domain OE grey-box models at 0% and 5% power. 72
5.30 Frequency response function of the black-box and frequency-domain OE grey-box models at 10% and 15% power. 72

5.31 Frequency response function of the black-box and frequency-domain OE grey-box models at 20% and 25% power. 73
5.32 Frequency response function of the black-box and frequency-domain OE grey-box models at 30% and 35% power. 73
5.33 Frequency response function of the black-box and frequency-domain OE grey-box model at 40% power. 74
5.34 Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 0% and 5% power. 74
5.35 Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 10% and 15% power. 75
5.36 Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 20% and 25% power. 75
5.37 Pole-zero map comparison between the black-box and frequency-domain OE grey-box models at 30% and 35% power. 76
5.38 Pole-zero map comparison between the black-box and frequency-domain OE grey-box model at 40% power. 76
5.39 Monte Carlo distribution of the parameter Y_v estimated with the $\mathcal{H}_\infty$ model-matching approach. 80
5.40 Monte Carlo distribution of the parameter Y_p estimated with the $\mathcal{H}_\infty$ model-matching approach. 80
5.41 Monte Carlo distribution of the parameter L_v estimated with the $\mathcal{H}_\infty$ model-matching approach. 81
5.42 Monte Carlo distribution of the parameter L_p estimated with the $\mathcal{H}_\infty$ model-matching approach. 81
5.43 Monte Carlo distribution of the parameter Y_δ estimated with the $\mathcal{H}_\infty$ model-matching approach. 82
5.44 Monte Carlo distribution of the parameter L_δ estimated with the $\mathcal{H}_\infty$ model-matching approach. 82
5.45 Monte Carlo distribution of the parameter Y_v estimated with the frequency-domain Output-Error minimization procedure. 83
5.46 Monte Carlo distribution of the parameter Y_p estimated with the frequency-domain Output-Error minimization procedure. 83
5.47 Monte Carlo distribution of the parameter L_v estimated with the frequency-domain Output-Error minimization procedure. 84
5.48 Monte Carlo distribution of the parameter L_p estimated with the frequency-domain Output-Error minimization procedure. 84

5.49 Monte Carlo distribution of the parameter Y_δ estimated with the frequency-domain Output-Error minimization procedure. 85
5.50 Monte Carlo distribution of the parameter L_δ estimated with the frequency-domain Output-Error minimization procedure. 85

List of Tables

3.1 Black-box cross-validation results for different WindShaper power levels and PBSID$_{opt}$ optimal *hyperparameters*, considering the SISO system with input δ_{lat} and output p [13]. 34
3.2 Nominal structured parameter estimates for different WindShaper power levels, considering the SISO grey-box model with input δ_{lat} and output p [13]. 34

4.1 PBSID$_{opt}$ *hyperparameters* and cross-validation performance of the unstructured models. 45

5.1 Cross-validation performance of the grey-box models obtained with $\mathcal{H}_\infty$ and frequency-domain OE methods. 59
5.2 Nominal structured parameter estimates for different WindShaper power levels, considering the SIMO PBSID$_{opt}$-$\mathcal{H}_\infty$ grey-box model with input δ_{lat} and outputs p and a_y. 77
5.3 Nominal structured parameter estimates for different WindShaper power levels, considering the SIMO PBSID$_{opt}$-OE grey-box model with input δ_{lat} and outputs p and a_y. 78