

POLITECNICO DI MILANO
School of Industrial and Information Engineering
Master of Science in Aeronautical Engineering



Model identification and inversion-based control for
multi-rotor UAVs

Advisor: Prof. Marco LOVERA
Co-Advisor: Dr. Simone PANZA
Eng. Mattia GIURATO

Thesis by:
Daniele MIGLIORE Matr. 883572

Academic Year 2018–2019

A mia sorella

Acknowledgments

Durante il mio percorso di studi, ho incontrato molte persone che mi hanno sostenuto e permesso di crescere umanamente e professionalmente.

Per prima cosa vorrei ringraziare il professor Marco Lovera, che mi ha permesso di sviluppare una tesi interessante e innovativa, per la quale ho potuto applicare i miei studi in un ambito nuovo e attuale, rendendosi sempre disponibile per qualsiasi tipo di chiarimento. Non può mancare un grazie a Simone che, con le sue competenze tecniche, la sua passione e la sua costante presenza, mi ha appoggiato nello svolgimento delle prove sperimentali. Ringrazio inoltre Mattia per i suoi suggerimenti e la sua disponibilità.

Ringrazio Giovanni e gli altri tesisti del laboratorio per la compagnia durante questi mesi. Un grazie enorme va ad Alessandro, Gianluca, Luca, Ludovica, Marco e Salvatore, amici più che compagni di università. Grazie anche a Marc con cui ho trascorso gli ultimi anni.

Ringrazio anche gli amici che ci sono sempre stati, in particolare Mike e Adre.

Grazie di cuore ai miei genitori che mi hanno permesso di affrontare questi studi senza farmi mancare nulla e dandomi tutto l'affetto possibile. Grazie a Ilaria, che mi ha insegnato a non mollare mai. Grazie a mia nonna che mi ha sempre trattato come un figlio. Grazie a mio nonno che è sempre stato nei miei pensieri.

Infine, grazie a Carmen, la persona che mi è sempre stata accanto, sostenendomi ogni giorno.

Abstract

Nowadays the interest in unmanned aerial vehicles (UAVs) is constantly increasing for different military and civil applications; these activities call for high-level requirements and therefore high-performance control laws. Mathematical models of the UAV dynamics can be used to synthesize the controllers. Classically, the problem is solved with a SISO approach, i.e., for each axis a model of the attitude dynamics is identified and then used to tune the parameters of a cascade control system.

The aim of this thesis is to identify structured models for the angular and linear dynamics of a multi-rotor UAV, considering single-input-multiple-output (SIMO) problems. These structured models can be exploited to design control laws based on the UAV equations of motion.

Black-box and grey-box model identification techniques are studied and applied to a quadrotor UAV. Then a new approach to obtain a structured model starting from an unstructured one has been proposed. Estimation data has been collected in flight, in closed-loop and in laboratory conditions at the Aerospace Systems and Control Laboratory of Politecnico di Milano. After analyzing and comparing the results, it was possible to design and simulate model inversion control laws such as the Explicit Model Following (EMF) and the Dynamic Inversion (DI). The designed controllers have been deployed to the quadrotor flight control unit and compared sending repeatable command inputs from a ground station. Experimental results have verified the achieved performance and have validated the accuracy of the identified models by comparing measured data with simulated data.

Sommario

Al giorno d'oggi l'interesse per aeromobili a pilotaggio remoto (APR) è in continuo aumento in diverse applicazioni militari e civili, le quali prevedono requisiti sempre più stringenti e, pertanto, leggi di controllo ad alta prestazione. Per la sintesi dei controllori si possono usare modelli matematici rappresentanti la dinamica di questi velivoli. L'approccio comunemente utilizzato prevede l'identificazione di modelli della dinamica di assetto per il progetto di regolatori in cascata, considerando problemi di tipo singolo-ingresso-singola-uscita (SISO).

Lo scopo di questa tesi è quello di ottenere modelli strutturati di tipo singolo-ingresso-multi-uscita (SIMO) della dinamica angolare e della dinamica lineare per un multirobot APR. Questo può permettere di progettare leggi di controllo basate sulle equazioni della dinamica del volo del velivolo.

Tecniche di identificazione di modelli a scatola nera e a scatola grigia sono studiate e applicate a un quadricottero. Viene proposto inoltre un nuovo approccio per ottenere un modello strutturato partendo da uno non strutturato. I dati per la stima dei modelli sono stati raccolti in volo in anello chiuso presso il laboratorio di controllo e sistemi aerospaziali (ASCL) del Politecnico di Milano. Dopo aver analizzato e confrontato i risultati ottenuti, è stato possibile progettare e simulare le leggi di controllo quali la Dynamic Inversion (DI) e l'Explicit Model Following (EMF). I controllori progettati sono stati importati sull'unità di controllo di volo del quadricottero e confrontati comandando il drone da una stazione di terra. I risultati degli esperimenti hanno verificato le prestazioni raggiunte e validato l'affidabilità dei modelli identificati confrontando i dati misurati con quelli simulati.

Contents

Acknowledgments	I
Abstract	III
Sommario	V
List of figures	XI
List of tables	XV
Introduction	1
1 Multi-rotor dynamics and control review	5
1.1 Reference frames	5
1.2 Euler angles and three dimensional rotations	6
1.3 Flight dynamics equations	9
1.4 Actuator model	11
2 Drone platform	13
2.1 ANT-R quadrotor characteristics	13
2.2 Hardware components	14
2.3 Software components	15
2.4 Aerospace systems and control laboratory	16
I Quadrotor dynamics identification	17
3 Model identification of multi-rotor UAVs	19
3.1 Identification experiments	20
3.2 Black-box model identification	22
3.2.1 Predictor-Based System IDentification	23
3.2.2 Predictor-Based System IDentification parameters	24
3.3 Grey-box model identification	24
3.3.1 Equations of motion	25

3.3.2	Accelerometer measurements	26
3.3.3	Linear grey-box model estimation	26
3.3.4	Lateral dynamics	27
3.3.5	Longitudinal dynamics	28
3.3.6	Directional dynamics	29
3.3.7	Vertical dynamics	29
3.4	Model matching	30
3.4.1	The H_∞ approach	31
3.5	Validation phase	32
3.6	Previous works	33
3.7	Conclusions	33
4	Validation phase and results	35
4.1	ANT-R identification experiments	35
4.2	Black-box models	37
4.2.1	Lateral dynamics	37
4.2.2	Longitudinal dynamics	37
4.2.3	Directional dynamics	39
4.2.4	Vertical dynamics	41
4.3	Grey-box models	41
4.3.1	Lateral dynamics	44
4.3.2	Longitudinal dynamics	45
4.3.3	Directional dynamics	47
4.3.4	Vertical dynamics	49
4.4	Black-box and grey-box model identification results comparison .	51
4.5	Model matching	55
4.5.1	Directional dynamics	55
4.5.2	Vertical dynamics	55
4.5.3	Lateral and longitudinal dynamics	56
4.6	Uncertainty analysis	57
4.7	Conclusions	58
II	Quadrotor dynamics control	63
5	Dynamic inversion and explicit model following control laws	65
5.1	Complete plant model	65
5.2	ANT-R stock flight controller	66
5.3	Dynamic Inversion control law design	68
5.3.1	Stability issues	69
5.3.2	Inner loop: attitude	70
5.3.3	Outer loop: velocity	72
5.3.4	Error dynamics	73

5.4	Explicit Model Following control law design	74
5.4.1	Inner loop: attitude	75
5.4.2	Outer loop: velocity	75
5.4.3	Disturbance rejection	77
5.5	Conclusions	78
6	Simulator results	81
6.1	Dynamic Inversion simulator	81
6.2	Explicit Model Following simulator	84
6.3	Simulator results comparison	84
6.4	Robustness analysis	85
6.5	Conclusions	85
7	Experimental results	91
7.1	Flight controllers design	91
7.2	In-flight tests results	92
7.3	Simulations and in-flight tests comparison	99
7.4	Dynamic Inversion and stock ANT-R autopilot comparison	99
7.5	Directional dynamics	99
7.6	Conclusions	104
	Conclusions	105

List of Figures

1.1	X rotors configuration [1].	11
2.1	ANT-R quadcopter.	14
2.2	3DR Pixhawk Mini autopilot [2].	14
2.3	NanoPi computer companion [3].	15
2.4	Infrared OptiTrack camera [4].	16
2.5	Infrared sensitive markers [4].	16
3.1	Example of a frequency sweep injection.	20
3.2	Example of a PRBS injection.	21
3.3	Example of a 3211 signal.	22
3.4	Choice of n from inspection of the singular values in equation (3.15).	25
3.5	Lateral estimation data set.	28
3.6	Longitudinal estimation data set.	29
3.7	Directional estimation data set.	30
3.8	Vertical estimation data set.	31
4.1	Bode diagram of lateral black-box model: roll rate (top) and lateral acceleration (bottom).	38
4.2	Black-box closed-loop validation: roll angle ϕ	38
4.3	Black-box closed-loop validation of the lateral model: roll rate (top) and lateral acceleration (bottom).	39
4.4	Bode diagram of longitudinal black-box model: pitch rate (top) and longitudinal acceleration (bottom).	40
4.5	Black-box closed-loop validation: pitch angle θ	40
4.6	Black-box closed-loop validation of the longitudinal model: pitch rate (top) and longitudinal acceleration (bottom).	41
4.7	Bode diagram of directional black-box model: yaw rate.	42
4.8	Black-box directional model validation: yaw rate.	42
4.9	Bode diagram of vertical black-box model: vertical velocity (top) and vertical velocity derivative (bottom).	43
4.10	Black-box vertical model validation: vertical velocity (top) and vertical velocity derivative (bottom).	43

4.11 Bode diagram of lateral grey-box model: roll rate (top) and lateral acceleration (bottom).	44
4.12 Grey-box closed-loop validation: roll angle ϕ	45
4.13 Grey-box closed-loop validation of the lateral model: roll rate (top) and lateral acceleration (bottom).	46
4.14 Bode diagram of longitudinal grey-box model: pitch rate (top) and longitudinal acceleration (bottom).	46
4.15 Grey-box closed-loop validation: pitch angle θ	47
4.16 Grey-box closed-loop validation of the longitudinal model: pitch rate (top) and longitudinal acceleration (bottom).	48
4.17 Bode diagram of directional grey-box model: yaw rate.	48
4.18 Grey-box directional model validation: yaw rate.	49
4.19 Bode diagram of vertical grey-box model: vertical velocity (top) and vertical velocity derivative (bottom).	50
4.20 Grey-box vertical model validation: vertical velocity (top) and vertical velocity derivative (bottom).	51
4.21 Directional model Bode diagram comparison between SMI and output-error approaches: yaw rate.	52
4.22 Coherence function of the directional non-parametric frequency response function.	53
4.23 Lateral model Bode diagram comparison between SMI and output-error approaches: roll rate p (top), lateral acceleration a_y (bottom).	54
4.24 Coherence function of the lateral non-parametric frequency response functions: roll rate p (top), lateral acceleration a_y (bottom).	54
4.25 Directional model Bode diagram comparison between output-error and model-matching approaches: yaw rate.	55
4.26 Vertical model Bode diagram comparison between output-error and model-matching approaches: vertical velocity (top) and vertical velocity derivative (bottom).	56
4.27 Lateral model Bode diagram comparison between output-error and model-matching approaches: roll rate p (top) and lateral acceleration a_y (bottom).	58
4.28 Grey-box uncertainty analysis of lateral dynamics: Bode diagram.	59
4.29 Grey-box uncertainty analysis of lateral dynamics: pole-zero map.	59
4.30 Grey-box uncertainty analysis of longitudinal dynamics: Bode diagram.	60
4.31 Grey-box uncertainty analysis of longitudinal dynamics: pole-zero map.	60
4.32 Grey-box uncertainty analysis of directional dynamics: Bode diagram.	61
4.33 Grey-box uncertainty analysis of directional dynamics: pole-zero map.	61
4.34 Grey-box uncertainty analysis of vertical dynamics: Bode diagram.	62

4.35	Grey-box uncertainty analysis of vertical dynamics: pole-zero map.	62
5.1	ANT-R stock flight controller: attitude dynamics.	68
5.2	Dynamic Inversion block diagram.	69
5.3	Dynamic Inversion block diagram: inner loop.	71
5.4	Dynamic Inversion block diagram: outer loop, lateral axis.	73
5.5	Explicit Model Following block diagram.	75
5.6	Explicit Model Following block diagram: inner loop.	76
5.7	Explicit Model Following block diagram: outer loop.	77
6.1	Dynamic Inversion simulator: Simulink architecture.	83
6.2	Simulation comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , roll rate p , roll angle ϕ and lateral actuators control input δ_{lat}).	86
6.3	Simulation comparison: longitudinal speed command doublet response, from top to bottom (longitudinal velocity V_x , pitch rate q , pitch angle θ and longitudinal actuators control input δ_{long}).	87
6.4	Simulation comparison: roll angle command doublet response (from top to bottom: roll angle ϕ , roll rate p and lateral actuators control input δ_{lat}).	88
6.5	Dynamic Inversion robustness analysis: lateral velocity command doublet response (from top to bottom: lateral velocity V_y , roll rate p , roll angle ϕ and lateral actuators control input δ_{lat}).	89
7.1	Simulink flight controller architecture example.	93
7.2	DI and EMF comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , lateral position y , roll rate p , roll angle ϕ , lateral control actuator input δ_{lat}).	95
7.3	DI (a) and EMF (b) outer loop control action and model inversion terms comparison.	96
7.4	DI and EMF comparison: outer loop control action term.	96
7.5	DI and EMF comparison: outer loop model inversion term.	97
7.6	DI and EMF comparison: inner loop control action term.	97
7.7	DI and EMF comparison: inner loop model inversion term.	98
7.8	DI and EMF comparison: vertical axis (from top to bottom: vertical velocity V_z , vertical position z , vertical actuators control input δ_{vert}).	98
7.9	DI measured and simulated data comparison (from top to bottom: roll angle ϕ , roll rate p , lateral acceleration a_y , lateral control input actuator δ_{lat}).	100
7.10	EMF measured and simulated data comparison (from top to bottom: roll angle ϕ , roll rate p , lateral acceleration a_y , lateral control input actuator δ_{lat}).	101

7.11	DI and stock control laws comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , lateral position y , roll rate p , roll angle ϕ , lateral control actuator input δ_{lat}).	102
7.12	DI and stock control laws comparison: vertical axis (from top to bottom: vertical velocity V_z , vertical position z , vertical actuators control input δ_{vert}).	103

List of Tables

2.1	3DR Pixhawk Mini features [2].	14
2.2	NanoPi computer companion features [3].	15
4.1	PRBS parameters.	36
4.2	Grey-box models eigenvalues.	44
4.3	Lateral identified parameters.	45
4.4	Longitudinal identified parameters.	47
4.5	Directional identified parameters.	49
4.6	Vertical identified parameters.	50
4.7	Validation parameters comparison between SMI and output-error approaches: vertical dynamics.	52
4.8	Validation parameters comparison: directional dynamics.	52
4.9	Validation parameters comparison between SMI and output-error approaches: lateral dynamics.	53
4.10	Validation parameters comparison between SMI and output-error approaches: longitudinal dynamics.	53
4.11	Comparison of directional identified parameters.	56
4.12	Comparison of vertical identified parameters.	57
4.13	Comparison of lateral identified parameters.	57
4.14	Comparison of longitudinal identified parameters.	57
5.1	Complete plant identified parameters.	67
5.2	ANT-R stock flight controller: proportional gains.	68
5.3	ANT-R stock flight controller: PID parameters.	68
5.4	Dynamic Inversion: natural frequencies ω_n and damping ratios ξ for inner loop.	74
5.5	Dynamic Inversion: natural frequencies ω_n and damping ratios ξ for outer loop.	74
5.6	Linear Quadratic Regulator: inner loop state penalties.	79
5.7	Linear Quadratic Regulator: inner loop input penalties.	79
5.8	Linear Quadratic Regulator: outer loop state penalties.	79
5.9	Linear Quadratic Regulator: outer loop input penalties.	79
6.1	Dynamic Inversion: PID gains.	82

6.2 Dynamic Inversion: PI gains. 82
6.3 Dynamic Inversion: P gains. 82
6.4 Signal saturation values. 82
6.5 First order command filter values. 82
6.6 Second order command filter values. 82

Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aircraft characterized by the absence of a pilot aboard. For this thesis, multi-rotor UAVs will be considered. The interest in this type of small aircraft is steadily increasing over the years in numerous military and civilian applications such as disaster relief, search and rescue, cargo transport, industrial plant monitoring, motion-picture industry and many others. These activities impose high-level requirements.

To meet this growing demand, high-performance control laws are required. In fact, despite the advantages of weight, size and low costs, multi-rotor UAVs are affected by non-linearity and instability. Typically, flight controllers are designed simulating their closed-loop architecture with a model of the aircraft to be controlled, which is the result of an experimental identification campaign. System identification is the art of building mathematical models for a real system based on experimental data. Finding a model which best fits measured data is relatively simple, on the contrary, estimating a model possessing a structure which reflects the physical behaviour of the system and which can predict the behavior of the aircraft is far from trivial.

Nowadays, having an aircraft simulator is essential because it allows to save a lot of time (and resources) avoiding time-consuming and often risky experimental tests. Moreover, if a grey-box approach is followed, the structured model can be used to design model inversion-based control laws which can replace the stock autopilot deployed in the quadcopter, usually consisting in a PID cascade control system.

For the purpose of this thesis, both the angular and linear dynamics of a quadrotor UAV are identified considering single-input-multiple-output (SIMO) models instead of the classical single-input-single-output (SISO) ones for the attitude dynamics only. Two different approaches are studied: black-box and grey-box model identification. The former is performed using the subspace model identification (SMI) method, a robust and efficient technique which can deal with multiple-input-multiple-output (MIMO) problems, starting from measured data only, both in open and closed-loop conditions. The only disadvantage is the lack of a physical insight of the system which means that no information about the state space representation is given. For this reason, a grey-box approach is also used in order to estimate the parameters of a model of the aircraft with

physically-motivated structure (i.e., based on the vehicle equations of motion) using the output-error method, an iterative technique which requires the knowledge of the system in terms of dynamics equations. A new structured model identification technique is proposed which can overcome the disadvantages of the previous methods: unstructured black-box models can be used to obtain structured models by performing a model matching with an H_∞ approach.

With the hypothesis of decoupled dynamics at low speed, the quadrotor can be identified exciting each body axis from the hovering condition separately. The obtained models must be validated using different data from those used for the identification phase and simulating the closed-loop architecture for unstable modes. Moreover, an uncertainty analysis of the results is performed to study their reliability. The complete structured plant model of the aircraft and therefore the physical parameters (stability and control derivatives with respect to states and inputs) are so obtained. The stock autopilot implemented on the quadcopter flight control unit can be replaced with Dynamic Inversion (DI) and Explicit Model Following (EMF) control laws, designed inverting the equations of motion which are used to construct the structured model. Before testing in flight the quadcopter, simulations are executed together with a robustness analysis of the flight controllers.

State of the art

System identification has been much discussed in the last two decades in the literature. First of all an interesting overview of the subject can be found in [5], in which Ljung examines the art and technique of building mathematical model in different research areas.

Examples of black-box model identification for multi-rotor UAVs can be found in [6], [7], [8] and [9]. In [6] (2014), Bergamasco and Lovera provided the first contribution considering the closed-loop identification for the dynamics of a hovering quadrotor using the subspace model identification technique, together with a detailed uncertainty analysis of the identified models.

More recently, in [9], Wu and Lovera investigated the problem of model identification of the attitude dynamics of a small-scale helicopter, providing a detailed comparison of time-domain and frequency-domain techniques. The former is performed with the SMI method while the software CIPHER (Comprehensive Identification from Frequency Responses) is used for frequency-domain model identification.

Panizza, Riccardi and Lovera in [7] provided a comparison of the black-box and grey-box model identification for the attitude dynamics of a variable-pitch quadrotor, demonstrating the limits of the output-error method in comparison to the higher performance obtained with SMI approach. More details about this study can be found in [8]. Then the novel model-matching technique which is

proposed and described in [10] by Bergamasco and Lovera is performed for the longitudinal axis only (SISO problem).

Structured identification for the dynamics of multi-rotor UAVs can be also found in [11], [12], [13], [14] and [15] (lateral axis only), where CIPHER software is used to obtain the stability and control parameters of the aircraft. In [11] (Saetti et al.), once the multi-rotor physical parameters are obtained, the Dynamic Inversion (DI) and the Explicit Model Following (EMF) control laws are designed for the identified quadcopter. DI and EMF were also applied in [15] (Tischler et al.) for the inner attitude loop (lateral case only).

Thesis structure

To facilitate the reading, the organization of the thesis structure is provided:

- Chapter 1: an overview of the flight dynamics and control theory for multi-rotor UAVs is provided;
- Chapter 2: the quadrotor UAV used for the purpose of this thesis is described in detail. The hardware and software components are provided with also a description of the indoor facility used for the experimental tests;
- Chapter 3: different model identification techniques are described, together with a selection of possible identification inputs. An overview of the previous works carried out at the ASCL laboratory is also proposed;
- Chapter 4: the quadrotor identified models are analyzed in detail, together with a description of the validation phase. Black-box, grey-box and model matching identification results are compared both in the time and in the frequency domain. An uncertainty analysis of the final results is performed;
- Chapter 5: a procedure for the design of the Explicit Model Following and the Dynamic Inversion control laws is proposed in detail together with a description of the stock quadrotor autopilot;
- Chapter 6: the complete identified model is used to simulate the closed-loop architecture for a preliminary tuning of the controllers and to perform a robustness analysis;
- Chapter 7: the two different controllers are implemented on the quadcopter flight control unit and compared in the time domain. The accuracy of the simulators is tested comparing measured data with simulated data. Finally, a comparison of the stock autopilot control system and the inversion-based flight controller is proposed.

Chapter 1

Multi-rotor dynamics and control review

In this chapter, an overview of the flight dynamics and control of a multi-rotor UAV will be provided in order to understand all the formalism and conventions which can be found in this work. For a more detailed study of the subject, the reader can refer to [16], [17] and [18].

1.1 Reference frames

The motion of an aircraft can be described using different reference frames. Usually, multi-rotor UAVs are controlled in an Earth fixed frame while the equations of motion, that will be used in this thesis, are written in the body axes. These reference systems will be therefore described.

Earth fixed frame

The hypotheses of flat and still Earth surface are made since the experimental tests are performed in an indoor cage as will be shown in Chapter 2. For these reasons, a fixed frame $\mathcal{F}_E = \{O_E, N, E, D\}$ attached to the Earth can be considered an inertial reference system. The origin can be an arbitrary fixed point on the Earth, the standard convention provides the N axis pointing North, the E axis East and the D axis aligned with the direction of gravity, pointing downward. This reference system is also known as the NED (meaning North-East-Down) frame.

Body frame

Typically is preferred to express the equations of motion in a moving reference system instead of an inertial frame. The body frame $\mathcal{F}_B = \{O_B, X_B, Y_B, Z_B\}$ is used for this purpose. This frame, in fact, has the origin in the center of gravity of the aircraft and changes the orientation with it. The X_B axis is parallel to

the longitudinal axis of the aircraft, the Z_B axis lays in the plane of symmetry pointing downward and the Y_B axis is found according to the right-handed rule.

1.2 Euler angles and three dimensional rotations

In order to switch from one reference system to another, rotation matrices must be introduced. In this section, a procedure of how these rotation matrices can be obtained will be provided.

Euler angles

The Euler angles (Φ, Θ, Ψ) are three angular quantities which can be used to describe the position of a generic reference frame $\mathcal{F}_1 = \{x_1, y_1, z_1\}$, such as the fixed-body frame, with respect to an inertial reference one $\mathcal{F}_2 = \{x_2, y_2, z_2\}$. In order to make the axes of the two frames coincide, a sequence of three rotations is required. Considering the components of a generic vector defined in the system $\mathcal{F}_1 = \{x_1, y_1, z_1\}$ which have to be rotated to the system $\mathcal{F}_2 = \{x_2, y_2, z_2\}$, the passages to obtain the rotation matrix T_{21} from system 1 to system 2 will be provided, with T_{21} such that:

$$T_{21} \in \mathbb{R}^{3 \times 3}, T_{21}^{-1} = T_{21}^\top, \det(T_{21}) = 1. \quad (1.1)$$

The equation (1.1) means respectively that the matrix T_{21} represents a three dimensional rotation, is orthogonal and does not change the magnitude of the vector to be rotated.

To understand how rotation matrices work, one axis at a time is considered associating a rotation matrix R_k around the k axis for each Euler angle (Φ, Θ, Ψ) . For example, the rotation about the x axis of an angle Φ is:

$$R_x(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi & \cos \Phi \end{bmatrix}. \quad (1.2)$$

This transformation does not change the component of the vector aligned with the x axis and the sign of the rotation angle follows the right-handed rule. Other rotation matrices are obtained in a similar fashion:

$$R_y(\Theta) = \begin{bmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ \sin \Theta & 0 & \cos \Theta \end{bmatrix}, \quad (1.3)$$

$$R_z(\psi) = \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.4)$$

Rotations can be considered in cascade and the generic vector components expressed in the new reference system are so obtained:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = T_{21} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = R_z(\Psi)R_y(\Theta)R_x(\Phi) \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (1.5)$$

where T_{21} is the rotation matrix from the reference system \mathcal{F}_1 to the reference system \mathcal{F}_2 :

$$T_{21} = \begin{bmatrix} C_\Psi C_\Theta & C_\Psi S_\Phi S_\Theta - S_\Psi C_\Phi & C_\Psi C_\Phi S_\Theta + S_\Psi S_\Phi \\ S_\Psi C_\Theta & S_\Psi S_\Theta S_\Phi + C_\Psi C_\Phi & S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi \\ -S_\Theta & S_\Phi C_\Theta & C_\Phi C_\Theta \end{bmatrix}. \quad (1.6)$$

with $C_\alpha = \cos \alpha$ and $S_\alpha = \sin \alpha$. Since the matrix T_{21} is orthogonal, the inverse coincides with its transpose: $T_{21}^{-1} = T_{21}^T = T_{12}$. An interesting application for this thesis is the conversion of the velocity vector V_E expressed in the NED frame \mathcal{F}_E to the body frame \mathcal{F}_B :

$$V_E = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}, \quad (1.7)$$

$$v_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (1.8)$$

$$v_B = T_{BE}(\phi, \theta, \psi) V_E \quad (1.9)$$

$$T_{BE} = \begin{bmatrix} C_\psi C_\theta & S_\psi C_\theta & -S_\theta \\ C_\psi S_\phi S_\theta - S_\psi C_\phi & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\phi C_\theta \\ C_\psi C_\phi S_\theta + S_\psi S_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi & C_\phi C_\theta \end{bmatrix}. \quad (1.10)$$

where v_B is the velocity of the aircraft resolved to body axes, $T_{EB}(\phi, \theta, \psi)$ is the rotation matrix from the Earth fixed frame to the body frame, ϕ , θ , ψ are the roll angle, pitch angle and yaw angle respectively.

Time derivatives of Euler angles

The Euler rates are defined as:

$$\omega_e = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (1.11)$$

while the body angular velocity vector as:

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (1.12)$$

The relationship between Euler rates and body rates is useful from a practical point of view as body rates can be measured with sensors installed on-board in an easier way.

To get the body-axis rates from the Earth-axis rates, consider the Euler rates individually, resolve them individually to intermediate axes and then finally to body axes. Define the Euler rate elemental vectors:

$$\omega_{\dot{\phi}} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}, \omega_{\dot{\theta}} = \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix}, \omega_{\dot{\psi}} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \quad (1.13)$$

Rotate $\omega_{\dot{\psi}}$ through the angle θ about the y axis and add the result to the $\omega_{\dot{\theta}}$ vector. Rotate the sum about the x axis through the angle ϕ , and add the result to the $\omega_{\dot{\phi}}$ vector. The result is the vector of body-axis angular rates:

$$\omega_b = \omega_{\dot{\phi}} + R_x(\phi)[\omega_{\dot{\theta}} + R_y(\theta)\omega_{\dot{\psi}}]. \quad (1.14)$$

After some rearrangement:

$$\omega_b = E(\phi, \theta)\omega_e \quad (1.15)$$

with:

$$E = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (1.16)$$

To get the Earth-axis rates in terms of the body-axis rates, the transformation matrix E is inverted:

$$E^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}. \quad (1.17)$$

Unlike the T_{21} matrix in equation (1.6), the inverse of E is not the transpose and furthermore E^{-1} is singular for $\theta = \pm 90$ deg. This singularity is known as the gimbal lock and can be solved using quaternions. Finally, for the small perturbation theory, an approximation can be applied:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = E^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (1.18)$$

Quaternions

A possible solution for the problem of the gimbal lock is the use of quaternions. A quaternion is a four-dimensional representation of a sphere that can be used to

represent the orientation of the airplane body-fixed frame with respect to a fixed inertial one. Quaternions are generally represented in the form:

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (1.19)$$

where q_0, q_1, q_2, q_3 are real numbers such that $\|q\| = 1$. A relation between the Euler angles (ϕ, θ, ψ) and the quaternions q is provided:

$$\phi = \tan^{-1} \left[\frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right], \quad (1.20)$$

$$\theta = \sin^{-1} [-2(q_1q_3 - q_0q_2)], \quad (1.21)$$

$$\psi = \tan^{-1} \left[\frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right]. \quad (1.22)$$

Finally, the quaternion derivative as functions of the three body axes rotational rates (p, q, r) are given by:

$$\frac{d}{dt} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 & p & q & r \\ -p & 0 & -r & q \\ -q & r & 0 & -p \\ -r & -q & p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (1.23)$$

1.3 Flight dynamics equations

The dynamic equilibrium of an aircraft can be expressed by two vectorial equations:

$$F_a + F_v + F_i = 0, \quad (1.24)$$

$$M_a + M_v + M_i = 0 \quad (1.25)$$

where a, v and i indexes refer to *applied, reaction* and *inertial* respectively. For an aircraft in flight, the reaction forces and moments are null. The inertial forces and moments can be defined in an inertial reference frame as:

$$F_i = -\frac{dQ}{dt}, \quad (1.26)$$

$$M_i = -\frac{dK}{dt} - v_p \wedge Q \quad (1.27)$$

where Q is the momentum, K is the moment associated with the momentum and P is the reference point. Then, considering the reference point P coinciding with

the center of gravity CG, the applied forces and moments are:

$$F_a = \frac{dQ}{dt}, \quad (1.28)$$

$$M_a = \frac{dK}{dt}. \quad (1.29)$$

Linear equations of motion

Starting from equation (1.28) and by using Poisson's formulas, it is possible to write the linear motion equation:

$$F = \frac{d(Q)}{dt} = \frac{d(mv_b)}{dt} = \left(\frac{dm}{dt}\right)v_b + m\left(\frac{\partial v_b}{\partial t} + \omega_b \times v_b\right). \quad (1.30)$$

In the case of a UAV, the mass m does not change and therefore equation (1.30) can be re-written as:

$$mv_b + \omega_b \times (mv_b) = F_{ext} \quad (1.31)$$

where the vector F_{ext} represents the sum of the gravity force F_g and the forces generated by the UAV propellers F_{prop} :

$$F_{ext} = F_g + F_{prop}. \quad (1.32)$$

Angular equations of motion

The definition of the inertia matrix is introduced:

$$J_n = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{zy} & J_{zz} \end{bmatrix}. \quad (1.33)$$

where:

$$J_{xx} = \int (y^2 + z^2)dm, \quad J_{yy} = \int (x^2 + z^2)dm, \quad J_{zz} = \int (x^2 + y^2)dm, \quad (1.34)$$

$$J_{xy} = \int (xy)dm, \quad J_{xz} = \int (xz)dm, \quad J_{yz} = \int (yz)dm. \quad (1.35)$$

If the quadrotor frame can be considered symmetric, it follows that the equation (1.33) will be diagonal because $J_{xy} = J_{xz} = J_{yz} = 0$.

Considering now the equation (1.29), the angular motion equation can be obtained:

$$J_n \dot{\omega}_b + \omega_b \times J_n \omega_b = M_{ext} \quad (1.36)$$

where M_{ext} is the sum of the moments generated by the UAV propellers M_{prop} and the aerodynamic damping M_{damp} caused by the rotating propellers moving through the air:

$$M_{ext} = M_{prop} + M_{damp}. \quad (1.37)$$

These linear and angular equations of motion will be the starting point for the structured model identification performed in Part I.

1.4 Actuator model

Multi-rotor aircraft are controlled by regulating the thrust generated by each propeller. For the purpose of this thesis, quadcopter UAVs with an X rotors configuration will be analyzed. Figure 1.1 is reported as an example to show motor numbering and senses of rotation.

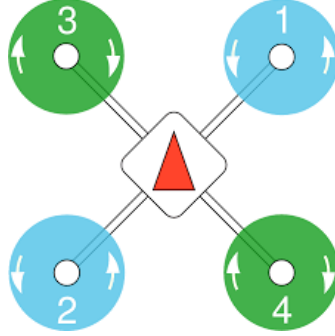


Figure 1.1: X rotors configuration [1].

Once the airframe of the multi-rotor aircraft is defined, forces and moments can be described more in detail. Thrust T and torque Q are quadratic functions of the rotational speed Ω of the i -th electric motors:

$$T_i = C_T \rho A R^2 = K_T \Omega_i^2, \quad (1.38)$$

$$Q_i = C_Q \rho A R^3 = K_Q \Omega_i^2 \quad (1.39)$$

where C_T and C_Q are the dimensionless thrust and torque coefficients, ρ is the air density, A and R are the area of the propeller disk and its radius respectively. Introducing also the parameter b as the distance between the center of gravity of the quadcopter and the j -th rotor, is now possible to define in the body axes the vectors F_{prop} and M_{prop} introduced respectively in equation (1.32) and equation (1.37):

$$F_{prop} = - \begin{bmatrix} 0 \\ 0 \\ K_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix}, \quad (1.40)$$

$$M_{prop} = \begin{bmatrix} K_T \frac{b}{\sqrt{2}}(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ K_T \frac{b}{\sqrt{2}}(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ K_Q(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \quad (1.41)$$

These vectors can be rearranged in order to construct the so called "allocation matrix" which describes the generated thrust and moments as functions of the

propeller speeds:

$$\begin{bmatrix} T \\ L \\ M \\ N \end{bmatrix} = M_{Allocation} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (1.42)$$

$$M_{Allocation} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -K_T b/2^{1/2} & K_T b/2^{1/2} & K_T b/2^{1/2} & -K_T b/2^{1/2} \\ K_T b/2^{1/2} & -K_T b/2^{1/2} & K_T b/2^{1/2} & -K_T b/2^{1/2} \\ K_Q & K_Q & -K_Q & -K_Q \end{bmatrix}. \quad (1.43)$$

The equation (1.42) is inverted to control the rotational motor speeds for the required thrust and moments:

$$M_{Mixer} = M_{Allocation}^{-1} \quad (1.44)$$

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = M_{Mixer} \begin{bmatrix} T \\ L \\ M \\ N \end{bmatrix} \quad (1.45)$$

$$M_{Mixer} = \begin{bmatrix} 1/(4K_T) & -2^{1/2}/(4K_T b) & 2^{1/2}/(4K_T b) & 1/(4K_Q) \\ 1/(4K_T) & 2^{1/2}/(4K_T b) & -2^{1/2}/(4K_T b) & 1/(4K_Q) \\ 1/(4K_T) & 2^{1/2}/(4K_T b) & 2^{1/2}/(4K_T b) & -1/(4K_Q) \\ 1/(4K_T) & -2^{1/2}/(4K_T b) & -2^{1/2}/(4K_T b) & -1/(4K_Q) \end{bmatrix}. \quad (1.46)$$

Finally, angular speed is transformed in voltage and applied to the motor.

Chapter 2

Drone platform

The ANT-R quadrotor is one of the latest multirotor UAV built at the Aerospace Systems and Control Laboratory (ASCL) at Politecnico di Milano, now used for research purposes. In this chapter, the quadrotor main characteristics will be provided, together with an overview of the ANT-R hardware and software components and of the test facility.

2.1 ANT-R quadrotor characteristics

The quadcopter was assembled with commercial off-the-shelf components, with the aim of obtaining a light racer quadrotor with high performance and optimized for forward flight. The ANT-R main features are:

- Take Off Weight (TOW): 800 *g*;
- Frame size: 250 *mm* carbon fiber frame;
- Motors: 4*x* EMAX brushless motors Race Spec;
- Propellers: 4*x* three-bladed propellers 5045;
- Rotor configuration: *H* configuration;
- Battery: 4 *s* Li-Po 2650 *mAh*;
- Flight time (hovering): 800 *s*.

A detailed view of the quadcopter is shown in Figure 2.1. As can be seen from the figure, the *H* rotors configuration, which can be considered equal to an *X* configuration in terms of rotors numbering, makes the platform asymmetric, determining different inertia moments about the *x* and *y* body axes.



Figure 2.1: ANT-R quadcopter.

Main Processor	STM32F427 Rev 3
Max current sensing	90 A
Max input voltage	45 V (10 S LiPo)
Dimensions	38 x 43 x 12 mm
Weight	15.8 g

Table 2.1: 3DR Pixhawk Mini features [2].

2.2 Hardware components

In this section the hardware components of the quadrotor will be described:

- Flight Control Unit (FCU): an electronic board which runs the flight control laws and the Inertial Measurement Unit. The Pixhawk Mini [2], a next-generation evolution of the Pixhawk, is used as FCU for the ANT-R quadrotor, see Figure 2.2 for a detailed view of the board and Table 2.1 for the main features;



Figure 2.2: 3DR Pixhawk Mini autopilot [2].

CPU	Quad-core Cortex-A7 Up to 1.2 GHz
RAM	512 MB DDR3
Storage	8 GB
PCB Size	$40\text{ x }40\text{ mm}$
Power Supply	DC $5\text{ V} / 2\text{ A}$
OS/Software	UbuntuCore
Weight	7.5 g

Table 2.2: NanoPi computer companion features [3].

- Flight companion computer: the NanoPi [3] is integrated onboard (see Figure 2.3), main features are reported in Table 2.2. The FCU and the NanoPi communicate with the MAVLink [19] messaging protocol;



Figure 2.3: NanoPi computer companion [3].

- Magnetometer: it measures the Earth's magnetic field and is used as further sensor for heading reference;
- Electronic Speed Controller (ESC): an electronic circuit used to limit and control the rotational speed of each DC motor;
- Power Distribution Board (PDB): it connects the ESCs to the drone battery;
- Radio receiver module: it is used to control the quadcopter with a remote-controller (RC); the RC used is the FrSky Taranis X9D Plus;
- GPS receiver: it can be used as additional source for position and linear velocity measurements;
- Safety switch: this component allows the user to arm the quadcopter.

2.3 Software components

As for the hardware, the software components are listed:

- PX4 firmware: an open source autopilot flight control system [20] for drones and other unmanned vehicles, which is run on the Pixhawk Mini;
- Robot Operating System (ROS): a software installed in the companion computer of the quadcopter, used to communicate with the ground station;
- MAVproxy: a fully-functioning Ground Control Station (GCS) for UAVs. Thanks to MAVproxy it is possible to control the Pixhawk Mini from GCS by the MAVLink protocol.

2.4 Aerospace systems and control laboratory

The laboratory used for all the experiments described in this thesis is the Aerospace Systems and Control Laboratory (ASCL) at Politecnico di Milano. Inside the laboratory there is the Flying Arena for Rotorcraft Technologies (FlyART), a dedicated flying cage for drones with dimensions $12 \times 6 \times 3 \text{ m}$. Since the indoor GPS signal is poor, this cage is equipped with the OptiTrack motion capture system (Mo-Cap) [4], which is used to determine the position (and also the attitude) of the quadcopter. The system is composed by 12 fixed Infra-Red (IR) sensitive cameras (see Figure 2.4) which track the quadcopter thanks to infrared sensitive markers (see Figure 2.5) mounted with a unique layout on the top of the multi-rotor airframe, so as to distinguish each drone. The software used to control the Mo-Cap system is the Motive software (see [4] for more details). Data are sent wirelessly to the NanoPi companion computer, which is connected to the Pixhawk FCU.



Figure 2.4: Infrared OptiTrack camera [4].

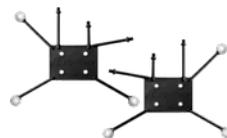


Figure 2.5: Infrared sensitive markers [4].

Finally, a Ground Station (GS) is equipped with two Operating Systems (OS): Windows OS (Windows 10 pro [21]) in which Motive is installed and Linux OS (Ubuntu [22]) in which MATLAB [23] is run.

Part I

Quadrotor dynamics identification

Chapter 3

Model identification of multi-rotor UAVs

This part of the thesis concerns the problem of system identification. System identification is the science of building mathematical models starting from input-output data. Given the growing need to have multi-rotor aircraft with high performance, it is necessary to have accurate models which better describe the real system, in order to simulate and implement high-performance control laws. For this reason, the field of system identification has been much discussed in last decades in the literature, see most recent studies [7], [8], [11], [6], [9], [13], [14], [15] and [24]. Model identification techniques can be characterized in terms of model structure (black-box, grey-box or white-box models), data domain (time domain or frequency domain) and type of the model (state space model or transfer function).

In this chapter, an overview of the input data selection for the identification phase will be provided. Then, two types of techniques will be analyzed: black-box model identification studying the subspace model identification (SMI) method, and grey-box model identification performed with the output-error method. As is known, the former is extremely robust and efficient, non iterative and gives the possibility to solve MIMO problems, both in open and closed-loop conditions. The only disadvantage is the lack of a physical meaning of the state space representation. This method indeed allows the user to derive a model starting from input-output data only. Output-error instead requires iterations and the knowledge of the equations which describe the real system, which can be derived, in this case, from Newton's second law. Moreover, closed-loop data have an output-input correlation which makes the estimates of this method biased. However, output-error allows to obtain the physical parameters of the aircraft, which are needed for the design of the model inversion control laws described in Part II of the thesis. Finally, a new approach will be shown (see [10]): the idea is to obtain a structured model starting from the identified black-box one by solving a model matching problem.

The UAV to be identified is the ANT-R quadrotor, the main characteristics of which are reported in Chapter 2. An overview of the previous system identification activities performed on this quadcopter will be provided.

3.1 Identification experiments

Different types of inputs can be studied for system identification. Usually, for this type of small aircraft, off-board automated inputs from the ground station are preferred rather than manually piloted with the remote-controller. MATLAB functions were implemented in order to have fast, precise and repeatable inputs, injecting the excitation before the quadcopter mixer (see Chapter 1.4 for more details about the mixer).

The decision of which class of input to use depends on the domain in which the identification phase will be performed. For the frequency-domain approaches, periodic excitations are used: a frequency sweep excitation is shown in Figure 3.1 as an example. The long duration of the input is needed in order to minimize data leakages in the computation of the frequency spectra and to excite the system at low frequencies. Other characteristic parameters of the sweep are the minimum and maximum frequency and the amplitude, which can be expressed as percentage of the maximum applicable input.

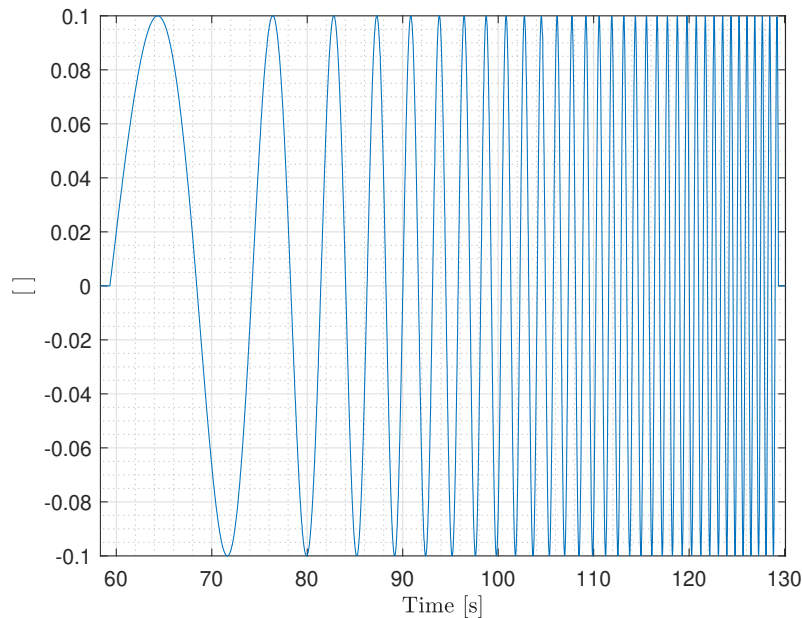


Figure 3.1: Example of a frequency sweep injection.

For the time-domain identification instead, it is possible to use shorter experiments to excite a large range of frequencies. An example is the Pseudo Random

Binary Sequence (PRBS); this signal consists in steps commands of constant amplitude and as the name suggests, the main property is the random choice of the step sign (see Figure 3.2 for an example). PRBS sequence is characterized by its duration, the bandwidth of excitation and the maximum (and minimum) normalized amplitude of the input.

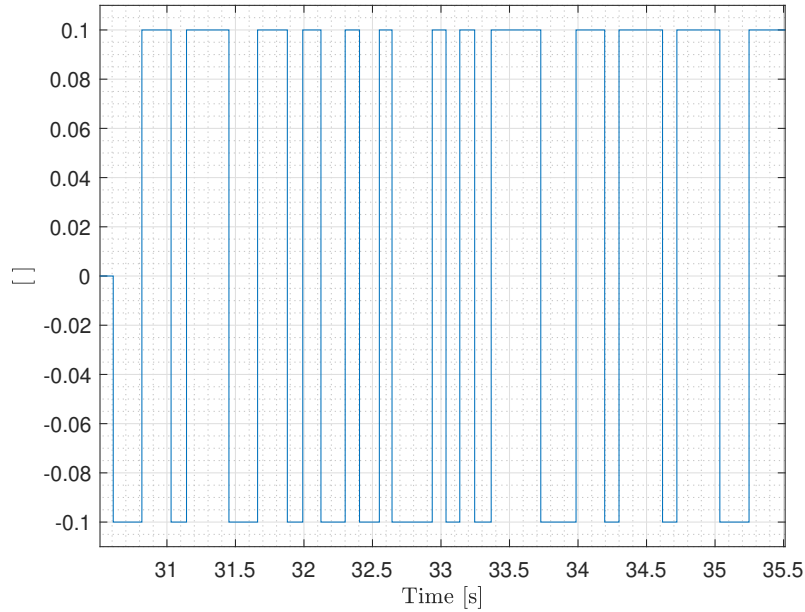


Figure 3.2: Example of a PRBS injection.

Another interesting signal for the time-domain identification is the so-called 3211 maneuver. The designation refers to the particular duration of each control reversal. Usually at least two different pulse amplitudes are used to verify the linearity of the system and the duration of the second pulse is chosen half the period of the dominant mode of the response to be identified. Guidelines and details can be found in [25] and an example of the sequence is shown in Figure 3.3.

Given the instability of a multicopter aircraft around the body x and y axes, frequency sweep excitation is preferred rather than PRBS and 3211 sequences, in order to use frequency-domain identification methods. Yaw and heave are typically stable, PRBS sequences and time-domain estimation can be used.

During the identification phase, delays must be taken into account: if present, they can lead to the estimation of a non-minimum phase system. For this reason, time delays must be estimated by studying the cross-correlation between signals, using the MATLAB function *delayest* or graphically, comparing inputs and outputs. Then, delays can be removed with a backward shift of the outputs. Finally, in the validation phase, they must be re-introduced to simulate the model and

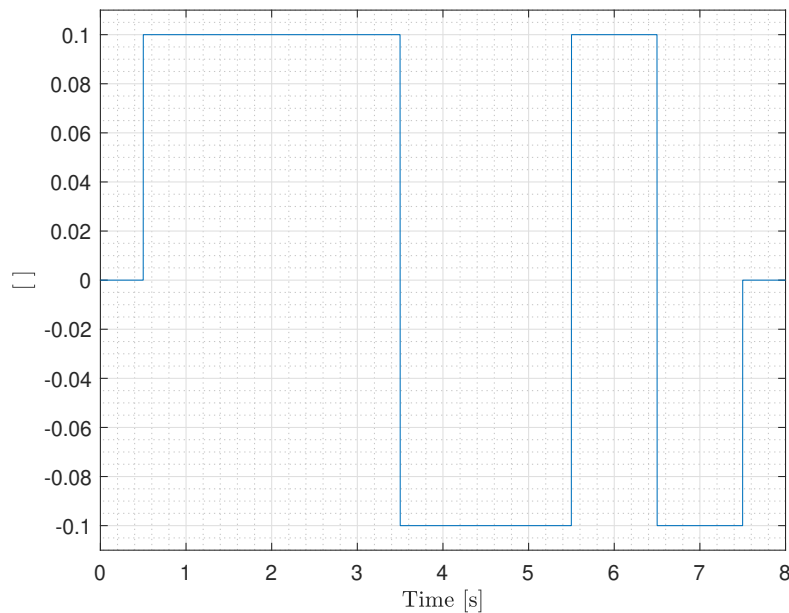


Figure 3.3: Example of a 3211 signal.

compare the simulated output with the measured output.

3.2 Black-box model identification

Black-box identification allows to obtain an unstructured model of a system, which means that the state space representation has no physical meanings, using only input-output data. Subspace Model Identification (SMI) belongs to this class of methods and has been much discussed in the last few decades in the literature (see [7], [8], [6], [9]). This technique in fact allows to obtain an unstructured model, starting from collected inputs and outputs data generated both in open and closed-loop conditions. Moreover, it is attractive since the algorithm is non iterative, it deals with SISO and MIMO problems, it is robust, efficient and very reliable from a numerical point of view, even considering short experiments; on the contrary, the frequency-domain approach requires collecting data for longer time in order to perform good frequency sweeps.

The procedure of the model identification through black-box modeling is now proposed in detail. To do this, the recent Predictor-Based System IDentification (PBSID) (see [26]) will be used and described.

3.2.1 Predictor-Based System Identification

In this section the PBSID algorithm is analyzed in detail. Considering the state space model representation:

$$x(t+1) = Ax(t) + Bu(t) + w(t) \quad (3.1)$$

$$y(t) = Cx(t) + Du(t) + v(t) \quad (3.2)$$

and assuming $w(t)$ and $v(t)$ are ergodic sequences of finite variance, the innovation form is obtained:

$$x(t+1) = Ax(t) + Bu(t) + Ke(t) \quad (3.3)$$

$$y(t) = Cx(t) + Du(t) + e(t). \quad (3.4)$$

Substituting e in (3.3) and introducing $Z(t)$ as the input-output vector:

$$x(t+1) = (A - KC)x(t) + (B - KD)u(t) + Ky(t) \quad (3.5)$$

$$x(t+1) = \bar{A}x(t) + [B - KD \quad K] \begin{bmatrix} u(t) \\ y(t) \end{bmatrix} \quad (3.6)$$

$$x(t+1) = \bar{A}x(t) + \bar{B}Z(t) \quad (3.7)$$

the following system of equations is obtained through forward propagation:

$$x(t+2) = \bar{A}^2x(t) + [\bar{A}\bar{B} \quad \bar{B}] \begin{bmatrix} z(t) \\ z(t+1) \end{bmatrix} \quad (3.8)$$

$$\vdots$$

$$x(t+p) = \bar{A}^p x(t) + K^p Z^{(t,t+p-1)} \quad (3.9)$$

with $K^p = [\bar{A}^{p-1}\bar{B} \quad \bar{B}]$ where p and f are called "past" and "future" windows sizes. Since \bar{A} is asymptotically stable: $\lim_{p \rightarrow \infty} \bar{A}^p = 0$ and if p is considered large enough, the output becomes:

$$y(t+p) = CK^p Z^{(t,t+p-1)} + Du(t+p) + e(t+p) \quad (3.10)$$

$$y(t+p+f) = CK^p Z^{(t+f,t+f+p-1)} + Du(t+f+p) + e(t+p+f). \quad (3.11)$$

Considering $p = f$, the CK^p and D matrices can be estimated solving a least square problem:

$$\min_{CK^p, D} \|y(t+p) - CK^p Z^{(t,t+p-1)} - Du(t+p)\|. \quad (3.12)$$

Since CK^p and K^p are now known, multiplying equation (3.9) by Γ^p defined as the extended observability matrix (3.13), with $\bar{A}^p = 0$:

$$\Gamma^p = \begin{bmatrix} C \\ C\bar{A} \\ \vdots \\ C\bar{A}^{p-1} \end{bmatrix} \quad (3.13)$$

$$\Gamma^p x(t+p) = \Gamma^p K^p Z^{(t,t+p-1)} \quad (3.14)$$

the second member of the equation is all known. Using the Singular Value Decomposition (SVD), an estimate of the state sequence X is obtained:

$$\Gamma^p K^p Z^{(t,t+p-1)} = U \Sigma V^T \quad (3.15)$$

$$X = \Sigma^{1/2} V^T. \quad (3.16)$$

The C matrix is obtained solving another least squares problem:

$$\min_C \|y(t) - CX(t) - Du(t)\|. \quad (3.17)$$

Finally, it is now possible to compute e and estimate also A, B, K :

$$\min_{A,B,K} \|X(t+1) - AX(t) - Bu(t) - Ke(t)\|. \quad (3.18)$$

3.2.2 Predictor-Based System Identification parameters

Once the estimation input-output data are collected in flight, black-box model identification is done choosing the model order n , the past and future windows sizes p and f . The order of the identifiable model can be guessed from sudden "drops" in the plot of the singular values computed in equation (3.15), see Figure 3.4 for an example.

Choosing systems with unnecessarily high order can make the model less reliable, usually a simple model which provides the best fit to data is preferred. The model order and parameters p and f that best meet this trade-off on a different input-output dataset, the so-called cross-validation dataset, are chosen as final values.

The obtained unstructured models will be used to perform the new model matching approach proposed in [10] and described in Section 3.4.

3.3 Grey-box model identification

Unlike the previous method, linear grey-box model identification is based on an iterative approach: the algorithm needs the definition of a parametric model class with unknown parameters, besides the estimation data. In order to define the model class, dynamics equations were derived from Newton's second law (see [18] and [16]). This physical insight is the advantage of this type of identification method: thanks to the obtained parameters it was possible to implement the model inversion control laws described in the Part II of the thesis.

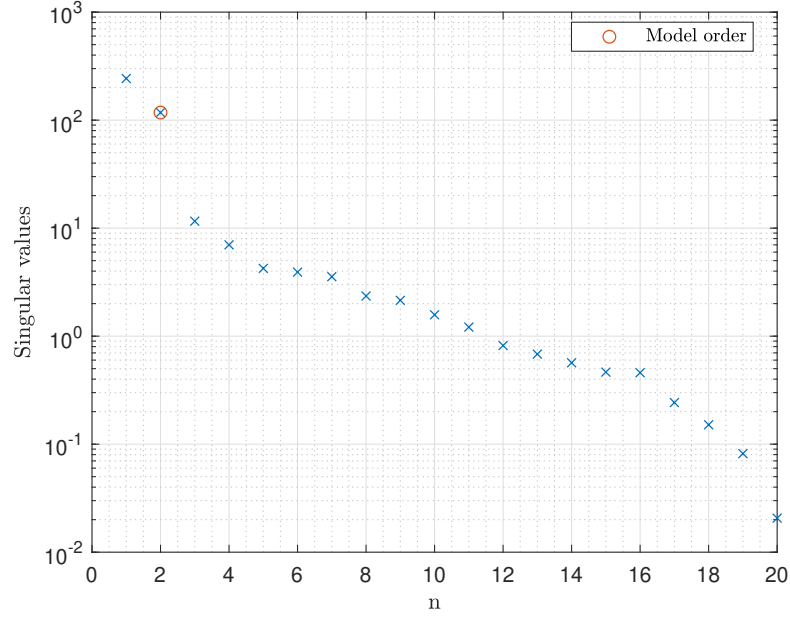


Figure 3.4: Choice of n from inspection of the singular values in equation (3.15).

3.3.1 Equations of motion

Starting from the Newton's second law, the force and moment equations expressed in the inertial reference system are obtained:

$$F = \frac{dQ}{dt} \quad (3.19)$$

$$M = \frac{dK}{dt} + v_p \wedge Q \quad (3.20)$$

where F and M are the external applied forces and moments, Q is the momentum, K is the moment associated with the momentum and P is the reference point. Then, with the hypothesis of small perturbations about the trim condition and considering the reference point P coinciding with the center of gravity, the equations of motion expressed in the quadrotor body-fixed frame (the origin is the center of gravity, with the x axis aligned with the longitudinal axis of the quadcopter, z axis is perpendicular to x and it lays in the longitudinal plane pointing downward, y axis is perpendicular to the other axes according to the right handed rule) are derived:

$$\dot{u} = X_u u + X_q q - g \sin \theta + X_\delta \delta_{long} \quad (3.21)$$

$$\dot{v} = Y_v v + Y_p p + g \sin \phi + Y_\delta \delta_{lat} \quad (3.22)$$

$$\dot{w} = Z_w w + Z_\delta \delta_{vert} \quad (3.23)$$

$$\dot{p} = L_v v + L_p p + L_\delta \delta_{lat} \quad (3.24)$$

$$\dot{q} = M_u u + M_q q + M_\delta \delta_{long} \quad (3.25)$$

$$\dot{r} = N_r r + N_\delta \delta_{dir} \quad (3.26)$$

where u , v and w are the perturbation speeds along the three body axes (x , y and z), p , q and r are the roll, pitch and yaw rates respectively, ϕ , θ and ψ are the roll, pitch and yaw angles. The parameters X_u , X_q , M_u , M_q , Y_v , Y_p , L_v , L_p , N_r , Z_w are the dimensional stability derivatives with respect to velocities and rates, δ_{lat} , δ_{long} , δ_{dir} , δ_{vert} are the control inputs, X_δ , Y_δ , Z_δ , L_δ , M_δ , N_δ are the dimensional control derivatives with respect to the relative control input and finally g is the gravitational acceleration. It must be observed that these derivatives parameters contain mass and moments of inertia of the quadcopter, for example:

$$X_u = \frac{1}{m} \left(\frac{dX}{du} \right), M_u = \frac{1}{J_{yy}} \left(\frac{dM}{du} \right). \quad (3.27)$$

3.3.2 Accelerometer measurements

Accelerometers measurements specific external forces excluding gravity. For this reason, these measurements were used as outputs for the identification phase:

$$a_x = \dot{u} + g \sin \theta = X_u u + X_q q + X_\delta \delta_{long} \quad (3.28)$$

$$a_y = \dot{v} - g \sin \phi = Y_v v + Y_p p + Y_\delta \delta_{lat} \quad (3.29)$$

$$a_z = \dot{w} - g = Z_w w - g + Z_\delta \delta_{vert}. \quad (3.30)$$

3.3.3 Linear grey-box model estimation

Grey-box model identification has been performed using the output-error method. Consider the data set of measured input-output $\{u(t_i), y(t_i)\}$ with $i \in [1, N]$ and the state space representation:

$$x(t+1) = A(\Theta)x(t) + B(\Theta)u(t) \quad (3.31)$$

$$y(t) = C(\Theta)x(t) + D(\Theta)u(t) + v(t) \quad (3.32)$$

with v a Gaussian process with zero mean defined as the measurement noise and Θ the vector of the unknown parameters. Disturbance acting on the plant is not included in this model class. Maximum Likelihood (ML) principle consists in choosing as estimate of the parameter the value of Θ which maximizes the likelihood function, which is defined as the joint probability of the observed data. Defining the prediction error e as the difference between the measured output y_m and the output of the model \tilde{y} :

$$e(k, \Theta) = y_m(k) - \tilde{y}(k, \Theta) \quad (3.33)$$

maximize the likelihood function is equal to minimize the cost function:

$$J(\Theta) = \frac{1}{N} \sum_{k=1}^N e(k, \Theta)^2 \quad (3.34)$$

which is equal to the sum of the squares of the deviations between the measured and the model output.

To do this, the MATLAB function *greyest* can be used. The initial system, which is described by the equations in Section 3.3.1 and Section 3.3.2, must have the same input and outputs dimensions of the estimations data. Data can be expressed both in time and frequency-domain. Since the equations of motion are written in continuous time, it is necessary to take this into account during the procedure. Then, the unknown parameters must be initialized, and they must constitute a stable system if the identification is performed in time-domain. For unstable modes in fact, *greyest* is used in frequency-domain. The numerical search method used for iterative parameter estimation was set to 'auto': Subspace Gauss-Newton, Adaptive subspace Gauss-Newton, Levenberg-Marquardt, and Steepest descent least squares search algorithms are tried in sequence at each iteration and the one which minimizes the cost function is chosen (see [27] for more details).

To distinguish between local and global minima of the cost function, different initial conditions are chosen for each unknown parameter of the considered systems. Once the model is found, the standard deviations of these parameters are also known. A good criterion to exclude parameters in the model structure is the standard deviation $\sigma_{\%}$ expressed as percentage of each parameter Θ_i : if $\sigma_{\%i} > 20\%$, Θ_i can be neglected, since it would not affect the cost function $J(\Theta)$ (see also [11] and [12]).

In the next subsections, the linearized systems used for grey-box model identification will be provided separately for each axis, considering decoupled dynamics at low speed.

3.3.4 Lateral dynamics

The linearized system which describes the lateral dynamics is:

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & g \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ L_\delta \\ 0 \end{bmatrix} \delta_{lat} \quad (3.35)$$

with outputs:

$$\begin{bmatrix} p \\ a_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ Y_v & Y_p & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} 0 \\ Y_\delta \end{bmatrix} \delta_{lat} \quad (3.36)$$

where the unknown parameters are the stability derivatives (Y_v , Y_p , L_v , L_p) and the control derivatives (Y_δ and L_δ). The input is the lateral actuators control

input, which is the sum of the injected excitation and the feedback action term. Roll rate and acceleration along y axis are considered as outputs as shown in Figure 3.5.

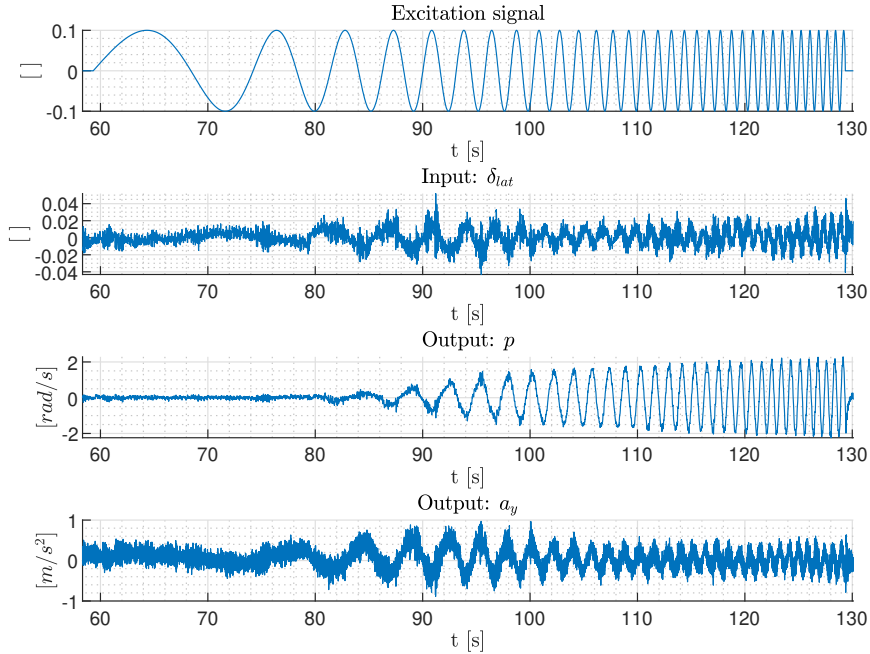


Figure 3.5: Lateral estimation data set.

3.3.5 Longitudinal dynamics

The linearized system which describes the longitudinal dynamics is:

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_q & -g \\ M_u & M_q & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_\delta \\ M_\delta \\ 0 \end{bmatrix} \delta_{long} \quad (3.37)$$

with outputs:

$$\begin{bmatrix} q \\ a_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ X_u & X_q & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ X_\delta \end{bmatrix} \delta_{long} \quad (3.38)$$

where the unknown parameters are the stability derivatives (X_u , X_q , M_u , M_q) and the control derivatives (X_δ and M_δ). The input is the longitudinal actuators control input and the considered outputs are pitch rate and acceleration along x axis (see Figure 3.6).

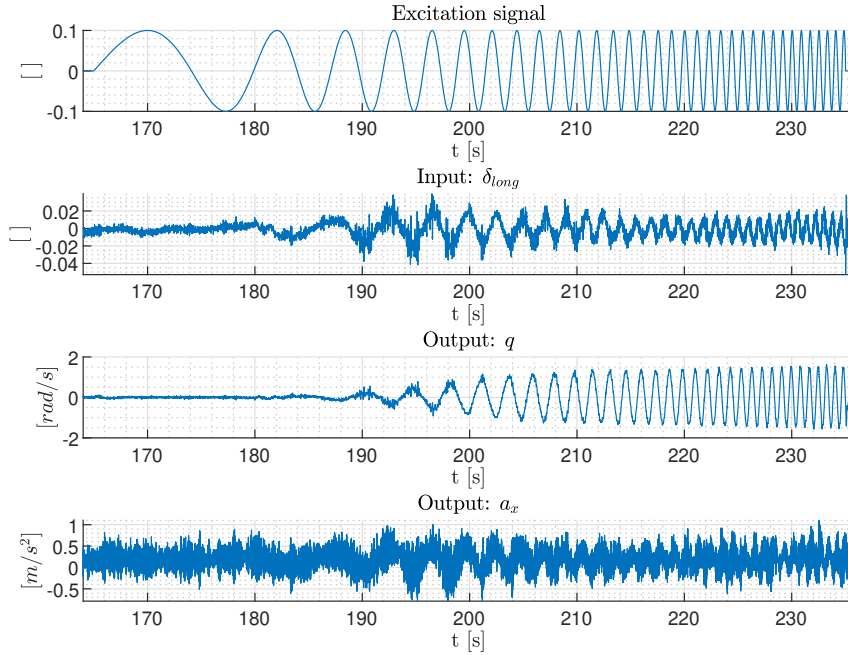


Figure 3.6: Longitudinal estimation data set.

3.3.6 Directional dynamics

In this case the linearized system is derived only from the moment equation:

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} N_r & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} N_\delta \\ 0 \end{bmatrix} \delta_{dir} \quad (3.39)$$

with r as output.

The unknown parameters are the stability and control derivatives N_r and N_δ . The input is the directional actuators control input, which is the sum of the injected excitation and the feedback action term. Yaw rate is the only output as can be seen in Figure 3.7.

3.3.7 Vertical dynamics

Finally, the z axis is considered:

$$\dot{w} = Z_w w + Z_\delta \delta_{vert} \quad (3.40)$$

with outputs:

$$\begin{bmatrix} w \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 1 \\ Z_w \end{bmatrix} w + \begin{bmatrix} 0 \\ Z_\delta \end{bmatrix} \delta_{vert}. \quad (3.41)$$

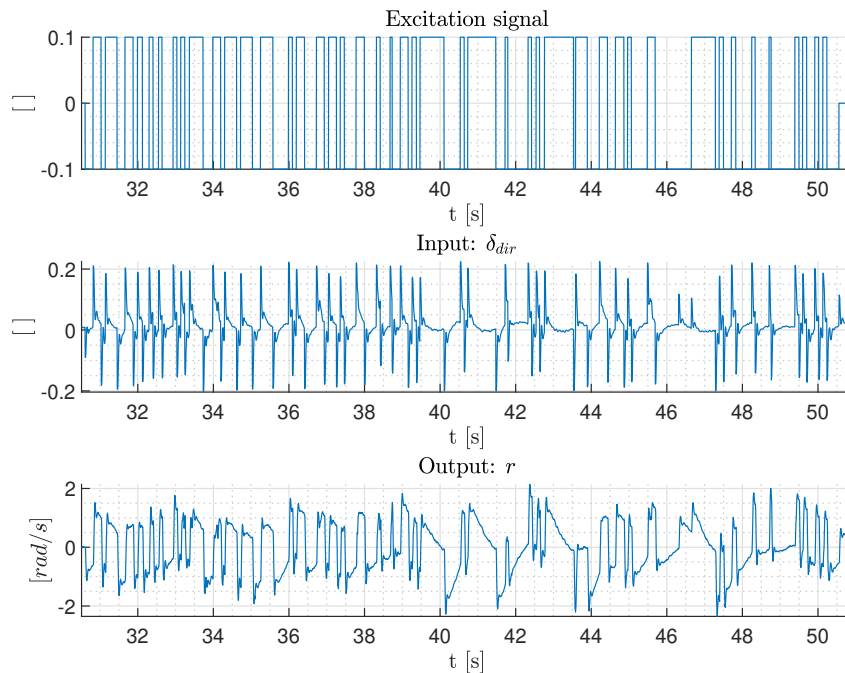


Figure 3.7: Directional estimation data set.

Parameters Z_w and Z_δ are the unknown of the system identification, with the vertical actuators control as input, the velocity along z axis and its derivative as outputs (see Figure 3.8).

3.4 Model matching

This section analyzes a more recent approach to perform state space model identification, following the method described in [10]. As seen in previous sections, there is no physical meaning in the state space representations obtained using SMI methods, but they are famous for their robustness, efficiency and non-iterative nature and they only require estimation input-output data. On the contrary, grey-box identification needs an initial system of the equations of motion, it is iterative, but it provides the physical parameters of the aircraft. Therefore, the idea is to obtain a structured model starting from an unstructured one, using structured H_∞ optimization. The complexity of the method lies in the number of uncertain parameters. This problem is mathematically equivalent to the one solved by P. Apkarian in 2006 (refer to [28] and [29] for details).

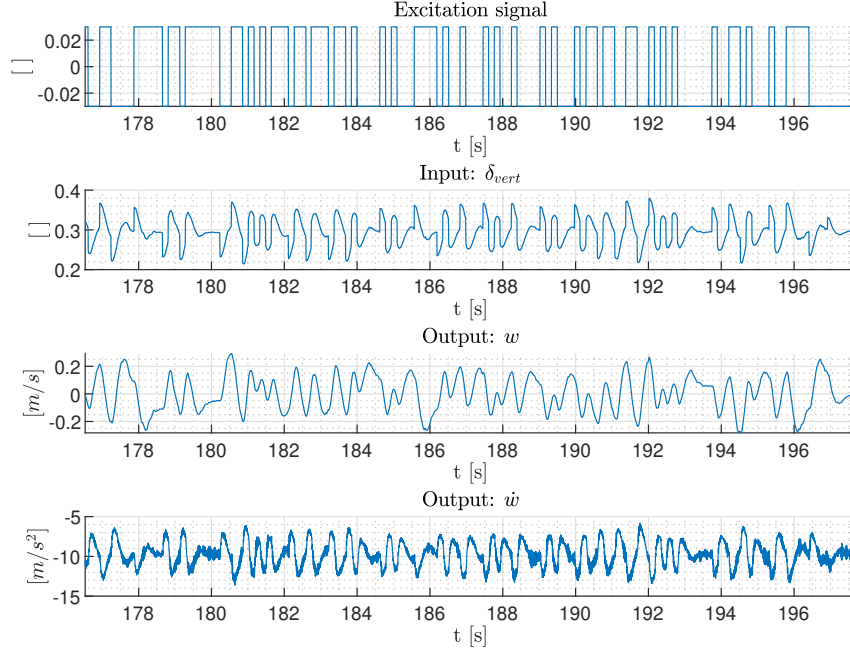


Figure 3.8: Vertical estimation data set.

3.4.1 The H_∞ approach

Given the system representing the unstructured identified model \hat{M}_{id} from observed input-output data:

$$\dot{x}(t) = \hat{A}x(t) + \hat{B}u(t) \quad (3.42)$$

$$y(t) = \hat{C}x(t) + \hat{D}u(t), \quad (3.43)$$

introducing Θ as the vector of the unknown physical parameters of the model, the structured state-space model $M_s(\Theta)$ is:

$$\dot{x}(t) = A(\Theta)x(t) + B(\Theta)u(t) \quad (3.44)$$

$$y(t) = C(\Theta)x(t) + D(\Theta)u(t). \quad (3.45)$$

The vector Θ is found by solving the optimization problem:

$$\Theta^* = \arg \min \|\hat{M}_{id} - M_s(\Theta)\|, \quad (3.46)$$

or using transfer functions:

$$\Theta^* = \arg \min \|\hat{G}_{id}(s) - G_s(s, \Theta)\|_\infty \quad (3.47)$$

where $\hat{G}_{id}(s)$ and $G_s(s, \theta)$ are the unstructured and structured transfer functions respectively.

The idea is then to minimize the difference between the two models (possibly of the same order) in terms of the H_∞ norm. This is done using the MATLAB function *systune* (see [30] and [31] for details): tunable elements of the system are initialized and the system of equations described in Section 3.3 is defined, then design requirements are specified focusing on a frequency range of interest (where the identified model captures the real system with high accuracy), paying attention to the default option of the function which imposes stability of the closed-loop system; actually since this is an identification problem there is no feedback, so the stability property is referred to the (open-loop) identified model. Finally *systune* is used to find the unknown free parameters.

Before matching the two systems, since the equations of motion are expressed in continuous-time, the identified model with PBSID algorithm must be converted from discrete to continuous-time (*systune* works with block elements with the same sampling time only). This has been done using the Tustin discretization method, which approximates the integral of the differential equation with trapezoidal areas.

Results can be compared with the ones obtained with the output-error model identification.

3.5 Validation phase

Once the identified model is found, it must be validated using additional input-output datasets; if the obtained model is unstable, a closed-loop validation must be performed. To compare the measured output with the model output and therefore to verify the fidelity of the identified model, validation parameters are used such as the Variance Accounted For (VAF), defined as:

$$\text{VAF} = \max \left(1 - \frac{\text{Var}(y_m - y_{est})}{\text{Var}(y)}, 0 \right) 100, \quad (3.48)$$

the percentage of fitting (FIT):

$$\text{FIT} = \max \left(1 - \frac{\|y_m - y_{est}\|^2}{\|y_m - \mathbb{E}(y_m)\|^2}, 0 \right) 100 \quad (3.49)$$

and the Prediction Error Cost (PEC):

$$\text{PEC} = \frac{1}{\sqrt{N}} \|y_m - y_{est}\|^2. \quad (3.50)$$

The two signals are the same if $\text{VAF} = \text{FIT} = 100\%$ and $\text{PEC} = 0$. Another validation method consists in comparing the estimate of the non-parametric frequency response function, obtained from input/output data, to the frequency response of the transfer function of the identified model.

3.6 Previous works

The multi-rotor aircraft chosen for system identification in this thesis is the ANT-R quadrotor, a racer platform used for research studies at the ASCL laboratory at Politecnico di Milano. A detailed description of the quadrotor can be found in Chapter 2. Until now, only the attitude dynamics of the quadcopter has been identified considering SISO problems and using subspace model identification with PRBS injections. These models in fact were needed to tune the gains of the cascade PID controller of the attitude loops (see Chapter 5.2 for more details about the ANT-R flight controller). In one of the most recent applications in fact, the platform was used as a follower drone to study UAV autonomous landing on moving aerial vehicle (see [32]) and therefore a high-performance controller was extremely necessary, without needing to have a physical insight of the state space representation of the system.

In this case, to obtain the physical parameters of the quadrotor which are needed for the model inversions control laws proposed in Part II, SIMO (single-input, multiple-output) structured problems must be studied, considering both the force and moment equations and considering all the axes.

3.7 Conclusions

In this chapter, different methods for both structured and unstructured system identification have been described. For grey-box model identification, the equations of motion along each axis were derived, considering decoupled dynamics at low speed. Additionally, an overview of the previous works performed on the identified quadcopter was proposed. The obtained models will be shown in detail in the next chapter and an uncertainty analysis of the results will be performed.

Chapter 4

Validation phase and results

In this chapter the identified models of the ANT-R quadrotor will be presented, validated and discussed using the subspace model identification, the output-error model identification and a new model matching technique, as seen in the previous chapter. Validation phase is performed differently on each axis, depending on the stability of the particular system. In fact, since this type of aircraft is open-loop unstable around the x and y axes, once the model is found, it must be validated considering the complete closed-loop architecture. On the contrary, when a stable system is validated, it is sufficient to simulate the response from input to outputs of the open-loop identified model. Results obtained with different techniques will be compared both in the time domain and in the frequency domain. Finally, an uncertainty analysis of the chosen models will be performed.

4.1 ANT-R identification experiments

All the experimental data have been obtained in flight in laboratory conditions: the ANT-R quadrotor was controlled off-board from the ground station, with position control activated for safety reasons. Automated inputs were preferred over those commanded by a pilot, in order to have precise, fast and repeatable signals. Considering dynamic decouplings at low speeds, the quadcopter has been perturbed from the hovering condition on each axis separately. The excitation has been injected directly on the actuators.

Frequency sweeps were considered to study lateral and longitudinal dynamics only, generating rolling and pitching moments. This choice comes from the frequency-domain method which has been used to obtain the grey-box model, given the unstable nature of the quadrotor dynamics. Sweeps were performed with a normalized amplitude of 0.1 (which means 10% of the maximum applicable moment), a range of frequency between 0.3 rad/s and 3 rad/s and a duration of 70 s. As seen in the previous chapter, the long duration is required in order to excite the system at low frequencies. Amplitude and frequency range parameters

	A []	ω [rad/s]	T [s]
Longitudinal dynamics	0.1, 0.2	[0-50]	20
Lateral dynamics	0.1, 0.2	[0-50]	20
Directional dynamics	0.1	[0-30]	20
Vertical dynamics	0.3	[0-20]	20

Table 4.1: PRBS parameters.

were the results of a build-up approach, starting from small values and increasing them gradually to excite correctly the dominant dynamics without any risk.

Pseudo random binary sequences were primarily chosen to identify directional and vertical dynamics, producing yawing moments and thrust on z axis. PRBS sequences have been also used for the lateral and longitudinal planes in order to compare results from different input excitations. Parameters for the sequences are reported in Table 4.1. As for the frequency sweeps, PRBS maximum frequencies and amplitudes were the results of several tests following a build-up approach. Tests with higher amplitudes have been terminated with the radio controller for safety reasons while higher frequencies would not have been consistent with the expected dominant attitude dynamics. The duration for all the tests was set to 20 s, a trade-off choice in order to have a short experiment reaching also low frequencies.

The following variables have been used as outputs for system identification: rates and accelerations for lateral and longitudinal planes, yaw-rate only for directional plane. These measurements come from the Inertial Measurements Unit (IMU) with a sampling time of $1/250$ s, which corresponds to 250 Hz. Finally, vertical velocity and its derivative were used for the z axis, where the first one was obtained by the OptiTrack, a motion capture system, with a sampling time of 0.01 s (100 Hz). The derivative of velocity has instead been calculated using a finite difference method. These physical quantities are fundamental for the aerodynamic derivatives of forces and moments (see Ref [18] and Chapter 3.3 for details). Actuator controls, which are the sum of injected signal and feedback term, were logged as inputs instead, expressed as normalized amplitudes (maximum value of 1).

For the validation phase of unstable dynamics, piloted maneuvers with velocity and angle set-point were executed and used to simulate the model in closed-loop conditions. Directional and vertical models were validated (and cross-validated for subspace models) on different PRBS sequences. Injections on position were also performed to simulate the complete closed-loop architecture (attitude and position) with amplitudes equal to 1 m and 2 m and a maximum frequency of 15 rad/s given the slower position dynamics (for the attitude maximum value was 50 rad/s).

It must be remembered that, when comparing the collected and simulated outputs, delays must be taken into account since they are removed during the

identification phase. Estimated delays range from 0.008 s and 0.032 s, depending on the signals.

4.2 Black-box models

Following the procedure explained in Chapter 3.2, subspace model identification has been performed to obtain unstructured models which best represent the dynamics of the quadcopter.

4.2.1 Lateral dynamics

The first considered problem is the lateral plane: 2 PRBS sequences and the best frequency sweep flight data were chosen for the model identification. Another sweep motion and 2 more PRBS sequences were used for the cross-validation. Then, 2 datasets for the closed-loop validation have been obtained piloting manually the ANT-R. Additionally, an input-output data set obtained with PRBS excitation injected on position y was also used for the validation phase to simulate the complete (attitude and position) closed-loop architecture of the quadrotor.

From singular values plot, the order of the model was found to be $n = 3$, which is in agreement with the system of equations described in grey-box model identification in Chapter 3, with past and future windows sizes $p = f = 12$. The Bode diagram of the model is shown in Figure 4.1. Best results were obtained with the frequency sweep excitations.

For the validation phase, since the ANT-R proved to be unstable around x axis, closed-loop simulation has been performed. The architecture from angle set-point to the quadrotor outputs (see Chapter 5 for more details) was simulated in closed-loop in the time-domain: roll rate, acceleration along y axis and the roll angle are shown in Figure 4.2 and Figure 4.3 as an example, comparing simulated and measured signals together with the Variance Accounted For (VAF) of the model. Closed-loop validation based on data collected in manually commanded flight has been shown since it represents a more realistic situation than a PRBS sequence response. Results indicate that the identified model is accurate considering the attitude, worse performance are obtained simulating the acceleration in closed-loop.

4.2.2 Longitudinal dynamics

Since the considered quadrotor is asymmetrical, the longitudinal plane must be studied with no symmetry assumption. As for the lateral case, 2 PRBS sequences and the best frequency sweep flight data were chosen for the model identification, another sweep motion and 2 more PRBS sequences were used for the cross-validation. Finally 2 data set have been obtained piloted manually the quadrotor

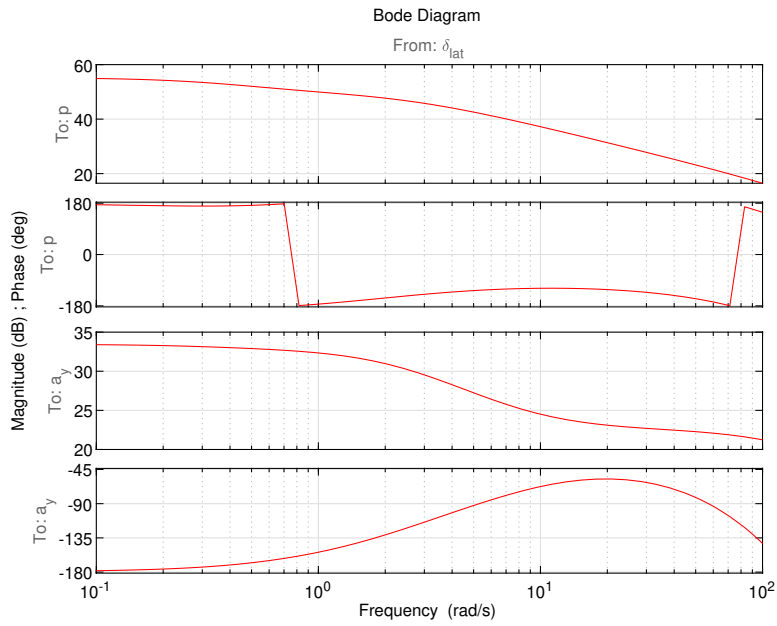


Figure 4.1: Bode diagram of lateral black-box model: roll rate (top) and lateral acceleration (bottom).

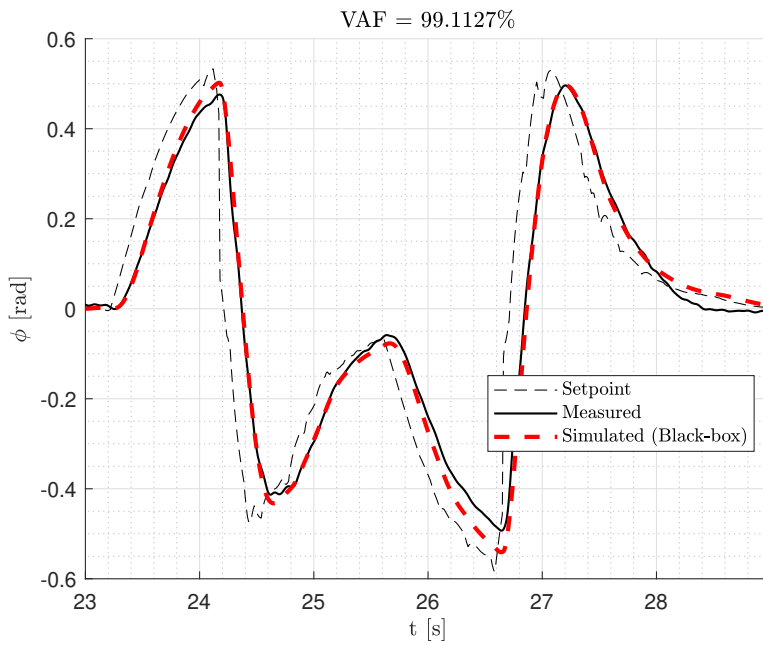


Figure 4.2: Black-box closed-loop validation: roll angle ϕ .

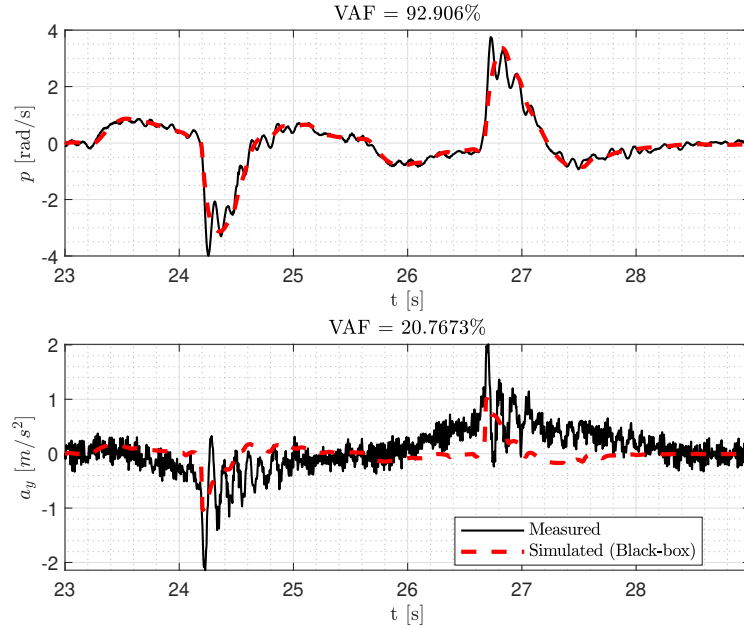


Figure 4.3: Black-box closed-loop validation of the lateral model: roll rate (top) and lateral acceleration (bottom).

for the closed-loop validation and 1 data set injecting a PRBS sequence on the position with the same parameter values as the lateral case.

The model order was found to be $n = 3$, with $p = f = 12$ as for the lateral case. The plot of the frequency response function is shown in Figure 4.4. Validation in the time-domain with measured data obtained piloting manually the ANT-R shows results similar to the lateral dynamics: the unstructured model can predict well only the attitude of the quadrotor (see Figure 4.5 for the pitch angle and Figure 4.6 for the complete model outputs simulation).

4.2.3 Directional dynamics

The first analyzed open-loop stable system is the directional dynamics: 5 data set were obtained with different PRBS sequences. 2 data set were used for the model identification, 1 for the cross-validation and the last 2 for the validation of the identified model.

Even if the system of equations described in Chapter 3 equation (3.39) is of the first order, considering $n = 3$ from singular values plot with $p = f = 21$, the model (Figure 4.7) has been found to be much more accurate. As can be seen from Figure 4.8, the VAF metric reaches 90% (if a first order model is considered, the VAF drops to 73%). Validation and cross-validation were done simulating directly the response of the model to a directional actuator input, given the stability of the quadcopter around z axis. This stable model gave very good results in terms

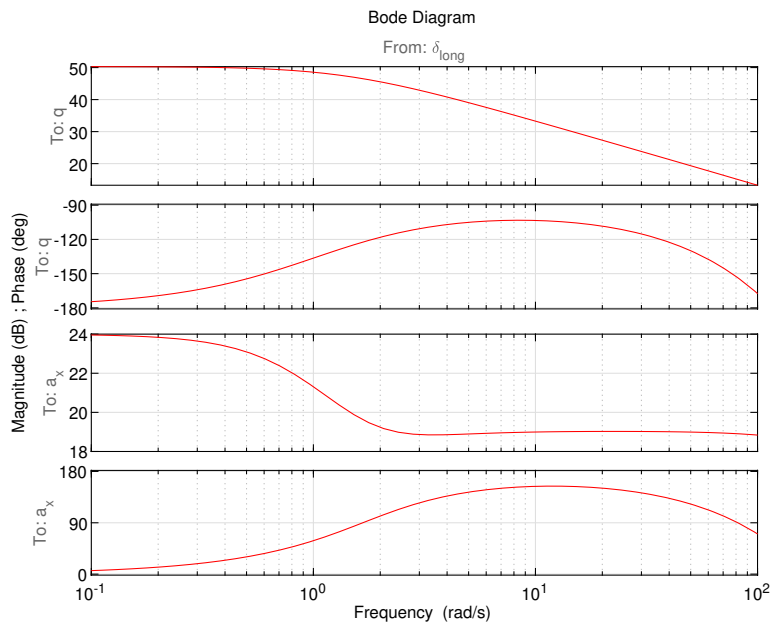


Figure 4.4: Bode diagram of longitudinal black-box model: pitch rate (top) and longitudinal acceleration (bottom).

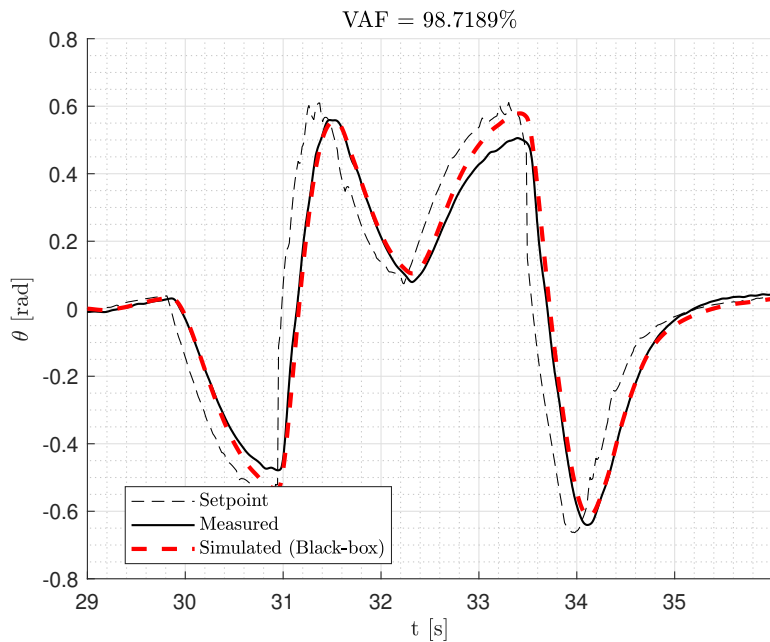


Figure 4.5: Black-box closed-loop validation: pitch angle θ .

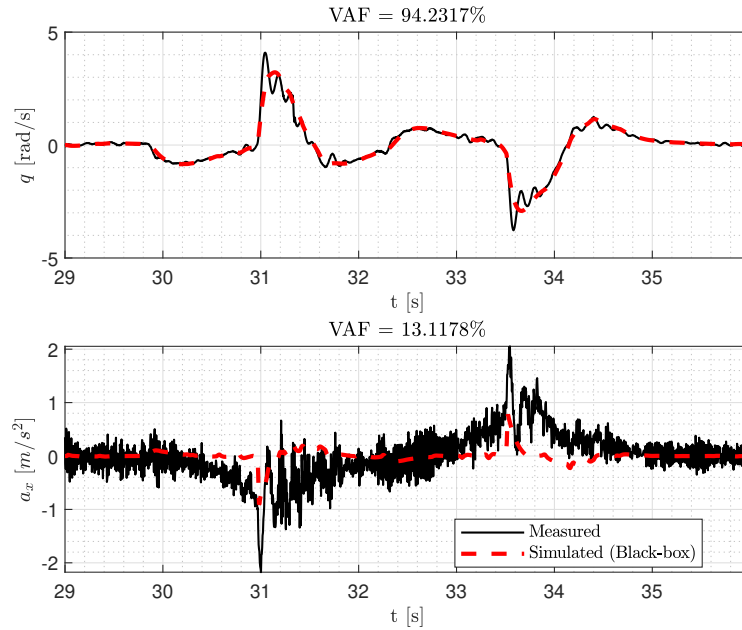


Figure 4.6: Black-box closed-loop validation of the longitudinal model: pitch rate (top) and longitudinal acceleration (bottom).

of VAF on all the considered datasets.

4.2.4 Vertical dynamics

Finally, the vertical axis has been considered. 3 best datasets obtained with 3 different PRBS injections were used to identify, cross-validate and validate the model.

The quadrotor proved to be stable along the z axis, with an identified black-box model which is very accurate for both the considered outputs (vertical velocity and its derivative), see Figure 4.10. A second order model with $p = f = 9$ has been chosen both by looking at the singular values and by cross-validating and comparing the simulated outputs with those measured (as for the directional case, a model order of 1 would have dropped the performance). The Bode diagram is illustrated in Figure 4.9.

4.3 Grey-box models

In this section structured models are derived, including their physical parameters (dimensional stability and control derivatives) and their accuracies. In order to compare the same dynamic model obtained with different methods, validation results are reported on the same data set.

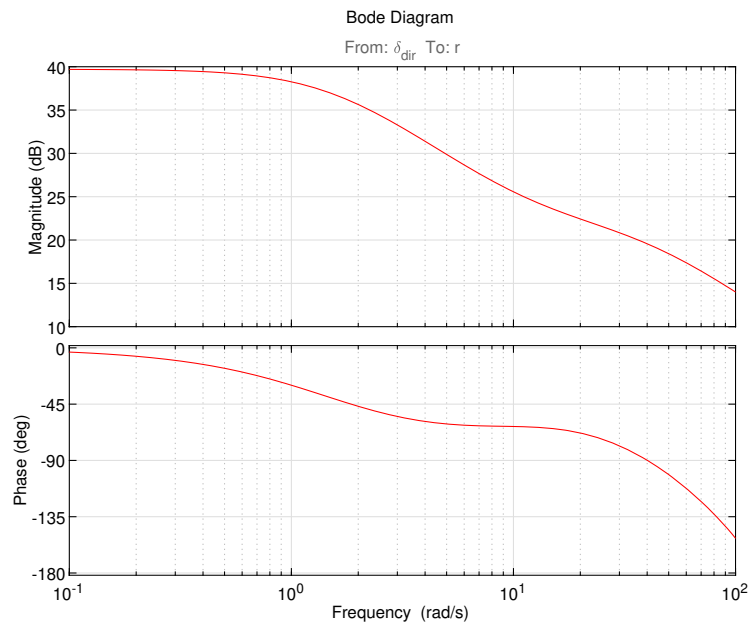


Figure 4.7: Bode diagram of directional black-box model: yaw rate.

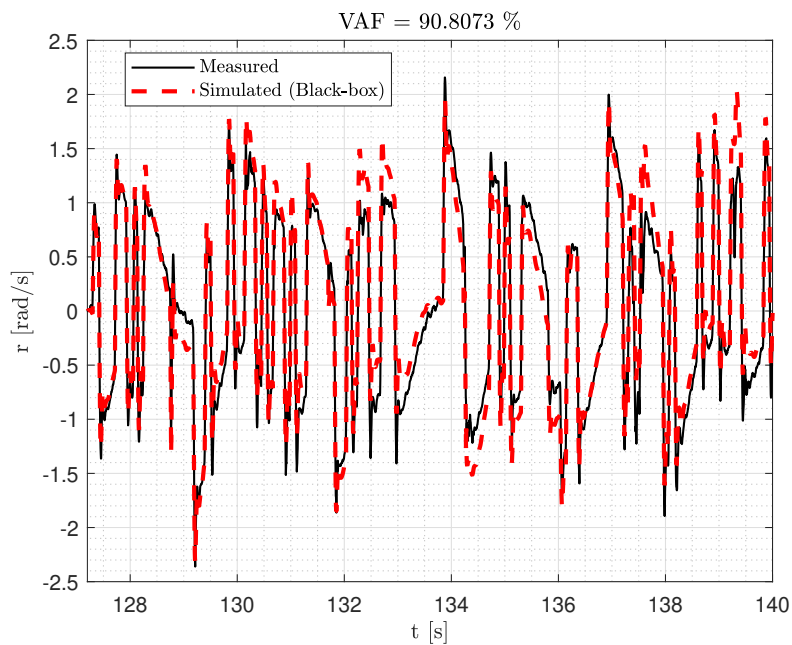


Figure 4.8: Black-box directional model validation: yaw rate.

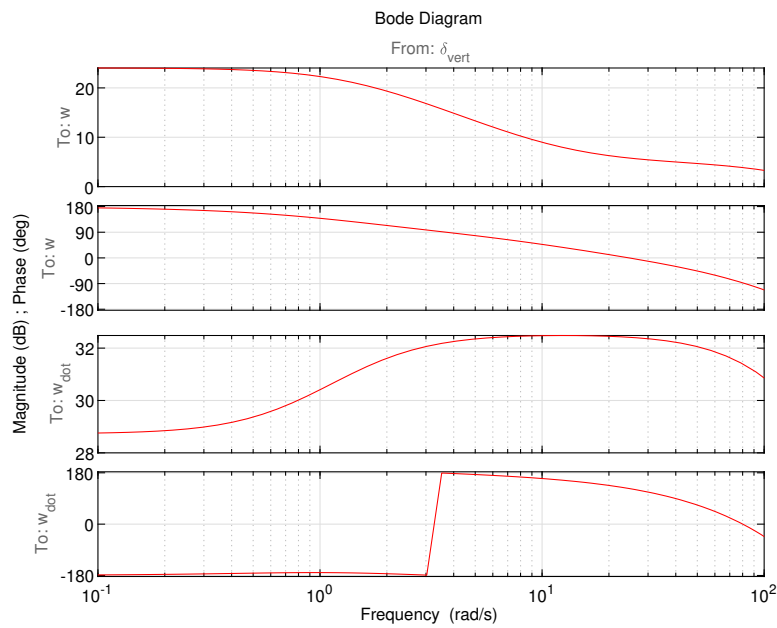


Figure 4.9: Bode diagram of vertical black-box model: vertical velocity (top) and vertical velocity derivative (bottom).

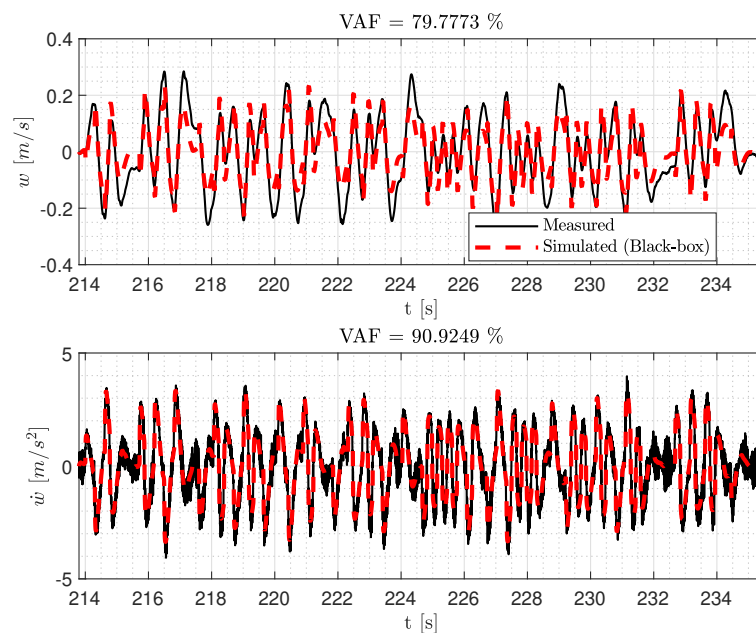


Figure 4.10: Black-box vertical model validation: vertical velocity (top) and vertical velocity derivative (bottom).

	Pole	Frequency [rad/s]	Damping ratio []
First order roll mode	-4.252	4.252	-
Roll oscillating mode	$1.994 \pm 3.603i$	4.118	-0.484
First order pitch mode	-4.250	4.250	-
Pitch oscillating mode	$2.008 \pm 3.611i$	4.132	-0.486
Yaw mode	-8.178	8.178	-
Vertical mode	-0.731	0.731	-

Table 4.2: Grey-box models eigenvalues.

4.3.1 Lateral dynamics

Grey-box model identification in frequency-domain is applied and the obtained lateral model is shown in Figure 4.11. The eigenvalues of the system are shown in

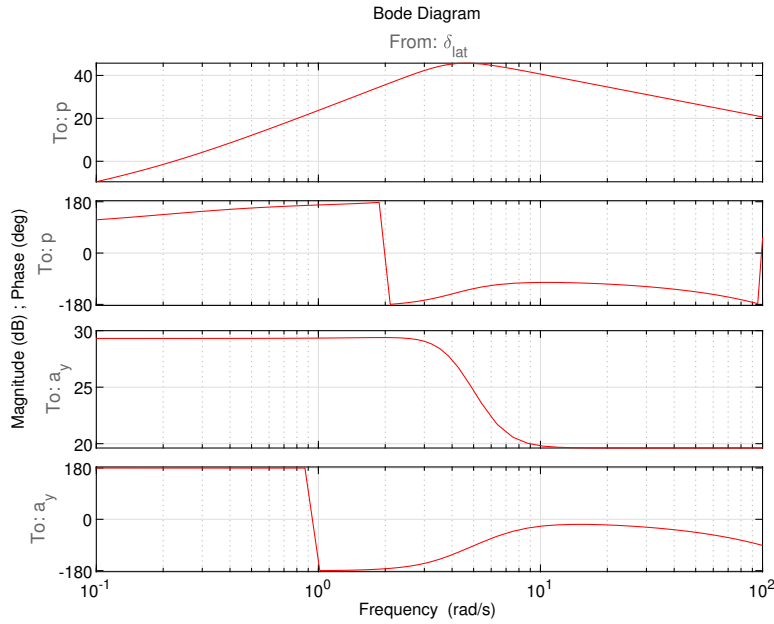


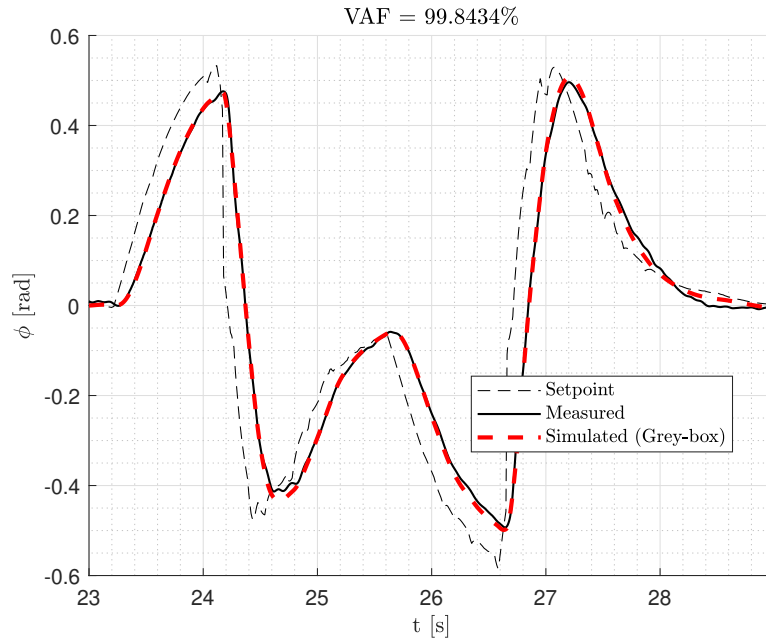
Figure 4.11: Bode diagram of lateral grey-box model: roll rate (top) and lateral acceleration (bottom).

Table 4.2. As can be observed, one of them has negative real part, and a pair of complex conjugate eigenvalues is present in the right-half plane, which represent an unstable mode for the quadrotor in open-loop conditions. The identified parameters and their standard deviation are reported in Table 4.3. The derivatives of force and moment with respect to roll rate were set to zero since $\sigma_{\%}$ was found to be 17.09% and 58.589% respectively (other values are lower than 5%). The same data set used for black-box validation are used here to simulate the ANT-R in closed-loop conditions. Results are shown in Figure 4.12 and 4.13 with the

Parameter	Value	$\sigma\%$
Y_v	-0.264 1/s	4.837
Y_p	0 m/(s rad)	-
L_v	-7.349 rad s/m	4.927
L_p	0 1/s	-
Y_δ	9.568 m/s ²	4.647
L_δ	1079.339 rad/s ²	2.762

Table 4.3: Lateral identified parameters.

Variance Accounted For of the model. As can be observed from Figures 4.3 and 4.13, the structured model follows the measured outputs even better than the black-box one, especially the acceleration.

Figure 4.12: Grey-box closed-loop validation: roll angle ϕ .

4.3.2 Longitudinal dynamics

As for the lateral case, the frequency-domain identification is performed on the longitudinal axis. The eigenvalues of the system are similar to the lateral ones, the small differences are due to the asymmetry of the quadrotor. Results are shown in Figure 4.14 and Table 4.2. Also in this case, the derivatives of force and moment with respect to pitch rate were set to zero since their standard deviation $\sigma\%$ was too large (96.3069% for the force and 111.6025% for the moment); the

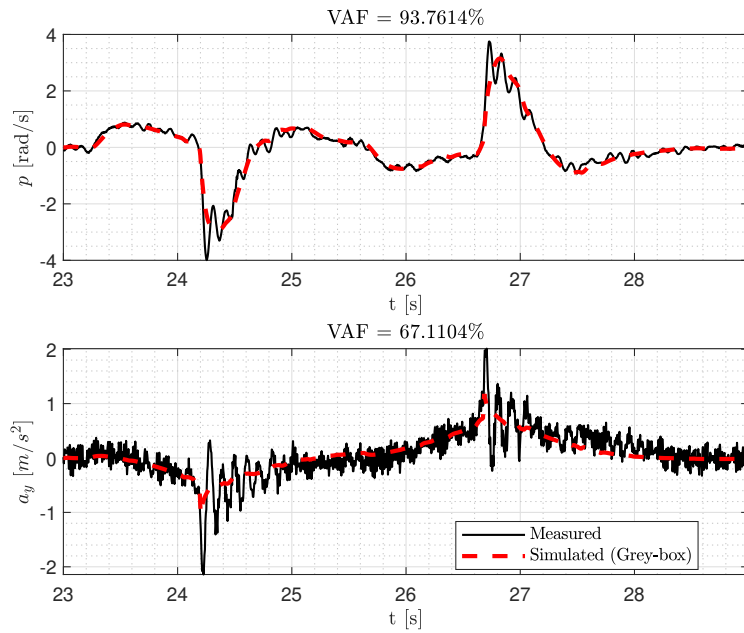


Figure 4.13: Grey-box closed-loop validation of the lateral model: roll rate (top) and lateral acceleration (bottom).

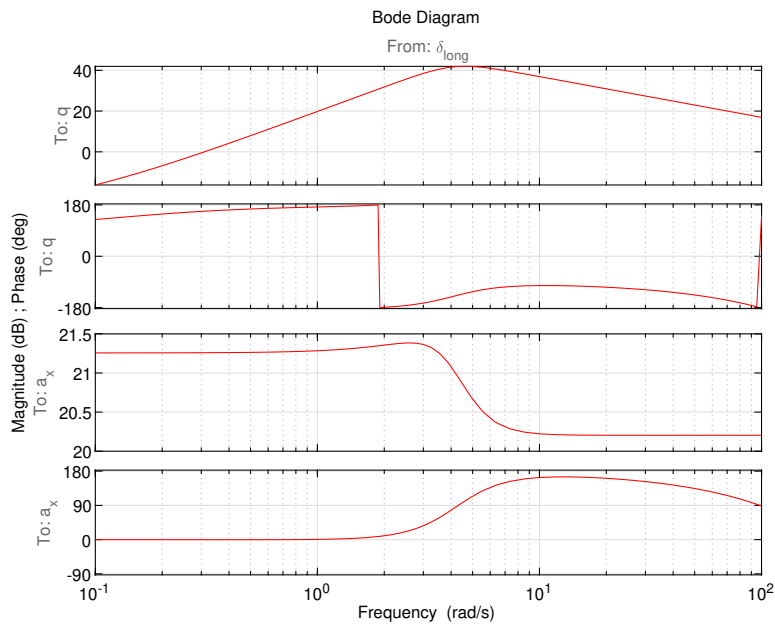
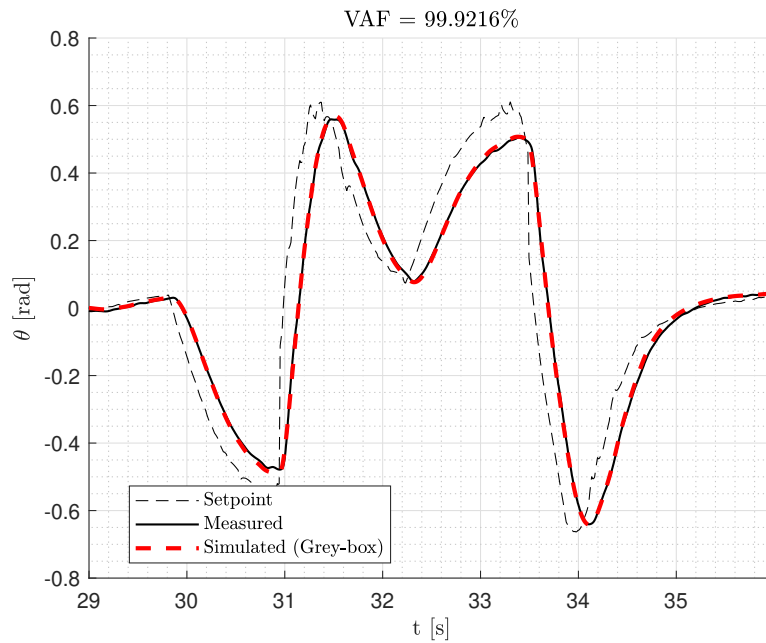


Figure 4.14: Bode diagram of longitudinal grey-box model: pitch rate (top) and longitudinal acceleration (bottom).

Parameter	Value	$\sigma\%$
X_u	-0.234 1/s	4.651
X_q	0 m/(s rad)	-
M_u	7.525 rad s/m	4.353
M_q	0 1/s	-
X_δ	-10.237 m/s ²	3.204
M_δ	701.578 rad/s ²	2.573

Table 4.4: Longitudinal identified parameters.

other values are reported in Table 4.4. The pitch angle setpoint has been used to simulate the closed-loop configuration: a comparison between simulated and measured outputs is shown in Figures 4.15 and Figure 4.16. The model showed very accurate results on both the considered outputs.

Figure 4.15: Grey-box closed-loop validation: pitch angle θ .

4.3.3 Directional dynamics

The quadrotor model identified by the *greyest* function has proved to be stable about the z axis. Figure 4.17 and Table 4.2 show the Bode diagram and the real negative pole of the model. Even if the percentage standard deviation is high (see Table 4.5) with respect to other identified parameters in the complete model, N_r was taken into account for the validation phase, which was done simulating

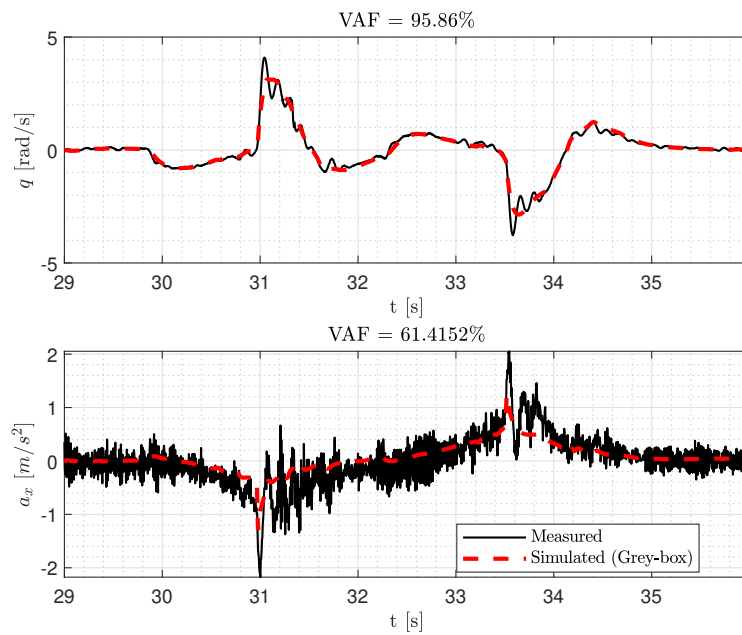


Figure 4.16: Grey-box closed-loop validation of the longitudinal model: pitch rate (top) and longitudinal acceleration (bottom).

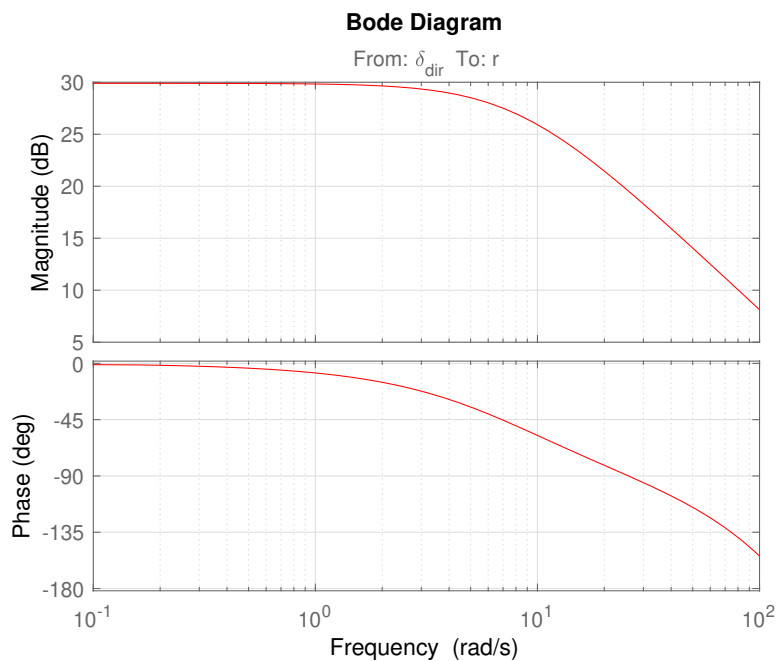


Figure 4.17: Bode diagram of directional grey-box model: yaw rate.

Parameter	Value	$\sigma\%$
N_r	-8.178 1/s	8.289
N_δ	255.590 rad/s ²	3.679

Table 4.5: Directional identified parameters.

the directional model with a different PRBS signal as excitation. Results of the validation in the time domain are reported in Figure 4.18. The model simulated well the measured outputs on all the validation data set.

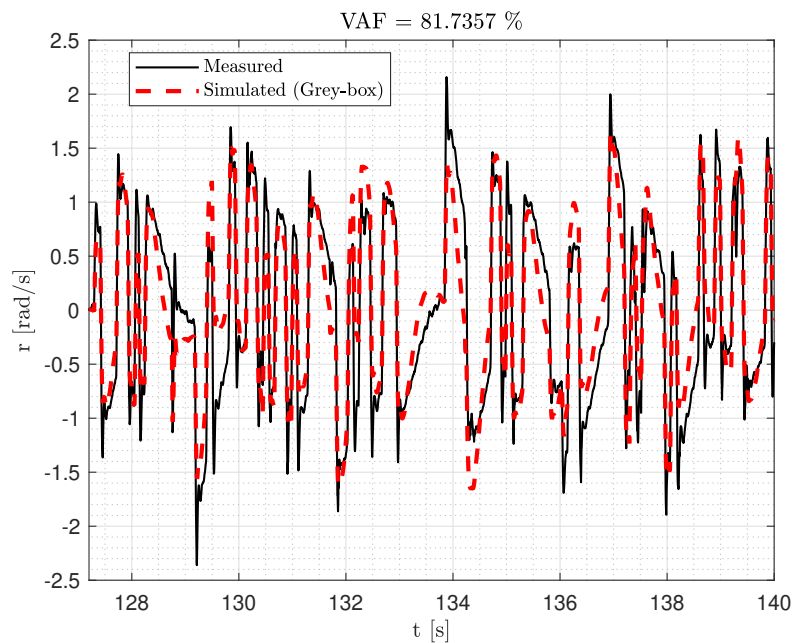


Figure 4.18: Grey-box directional model validation: yaw rate.

4.3.4 Vertical dynamics

Finally, as seen for the black-box model identification, vertical dynamics for the ANT-R was found to be stable, Bode diagram and eigenvalues are reported in Figures 4.19 and Table 4.2. As can be seen from Table 4.6, percentage standard deviations are extremely low. At the end, Figure 4.20 shows the comparison between the simulated and the measured outputs, which gave very good results both in term of vertical velocity and vertical velocity derivative.

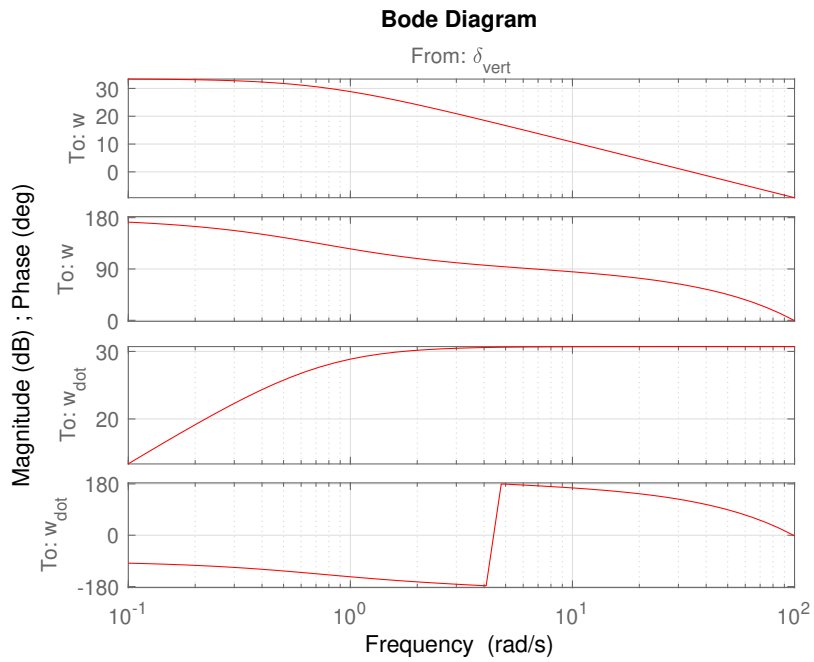


Figure 4.19: Bode diagram of vertical grey-box model: vertical velocity (top) and vertical velocity derivative (bottom).

Parameter	Value	$\sigma\%$
Z_w	-0.731 1/s	2.497
Z_δ	-34.351 m/s^2	0.401

Table 4.6: Vertical identified parameters.

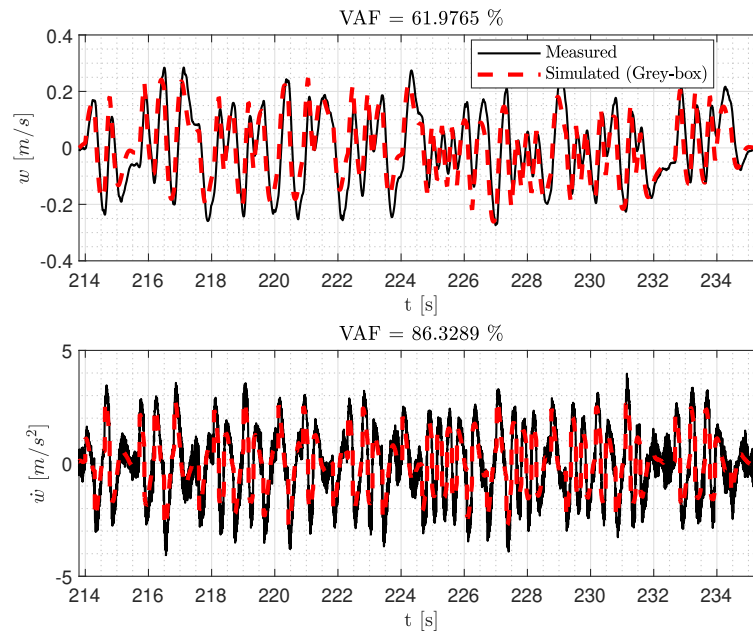


Figure 4.20: Grey-box vertical model validation: vertical velocity (top) and vertical velocity derivative (bottom).

4.4 Black-box and grey-box model identification results comparison

In this section a comparison between the obtained models is proposed. Starting from the vertical and directional dynamics, subspace model identification turned out to be more accurate than output-error model identification. This could be explained by the choice of n , p and f for the Predictor-Based System Identification technique: these were the results of several attempts with the goal of maximizing the validation parameters, without looking at the equation of motions described in Chapter 3.3, considering in fact a model order equal or higher than two. Results are shown in Table 4.7 for vertical dynamics and Table 4.8 for directional dynamics, reporting VAF metrics. The models were also compared with the estimate of the non-parametric frequency response functions obtained with the observed data. As an example, Figure 4.21 shows the comparison of the unstructured and structured directional models in the frequency domain, confirming the best accuracy of the unstructured one on a wider bandwidth. It is also reported the coherence function of the measured data for consistency, indicating over which frequency range the experimental data are valid, see Figure 4.22. It can be seen from the figure that a model of the first order can not correctly describe the physics of the real system. Similar results were obtained considering the vertical case. On the contrary, for the lateral and longitudinal cases, output-error identification

	SMI	Output-error
VAF_w	79.777 %	61.976 %
$VAF_{\dot{w}}$	90.925 %	86.329 %

Table 4.7: Validation parameters comparison between SMI and output-error approaches: vertical dynamics.

	SMI	Output-error
VAF_r	90.807 %	81.736 %

Table 4.8: Validation parameters comparison: directional dynamics.

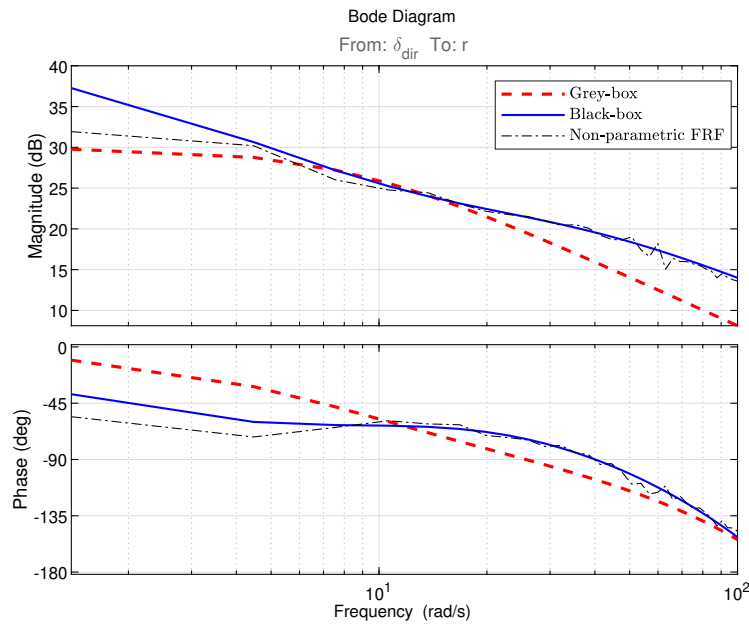


Figure 4.21: Directional model Bode diagram comparison between SMI and output-error approaches: yaw rate.

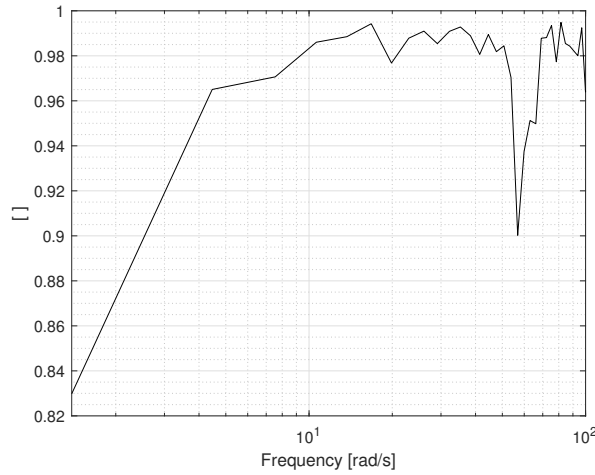


Figure 4.22: Coherence function of the directional non-parametric frequency response function.

	SMI	Output-error
VAF_p	92.906 %	93.761 %
VAF_{a_y}	20.767 %	67.110 %

Table 4.9: Validation parameters comparison between SMI and output-error approaches: lateral dynamics.

provided more accurate models considering both the outputs (angle rate and acceleration). The same procedure of the previous cases was followed: validation metrics are shown in Table 4.9 for lateral dynamics and Table 4.10 for longitudinal dynamics. As an example of the models comparison in the frequency-domain, Figure 4.23 is reported to show the lateral case (very similar results were obtained in the longitudinal plane), demonstrating the higher reliability of the structured model, especially in the bandwidth of interest (see also Figure 4.24 for the coherence function of the lateral non-parametric response function obtained from measured data). Finally, Figure 4.23 confirms results shown in Table 4.9: all the obtained models are more accurate considering the angular rate rather than the acceleration.

	SMI	Output-error
VAF_q	94.232 %	95.860 %
VAF_{a_x}	13.118 %	61.415 %

Table 4.10: Validation parameters comparison between SMI and output-error approaches: longitudinal dynamics.

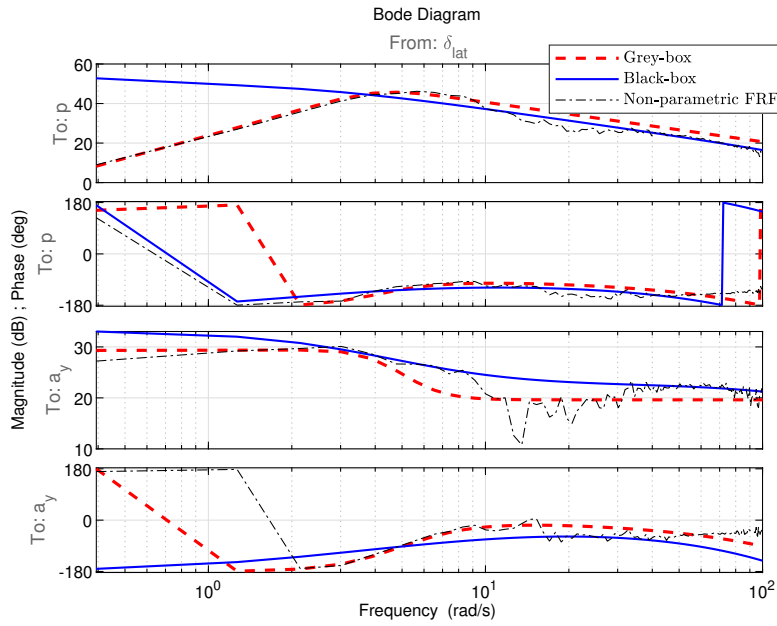


Figure 4.23: Lateral model Bode diagram comparison between SMI and output-error approaches: roll rate p (top), lateral acceleration a_y (bottom).

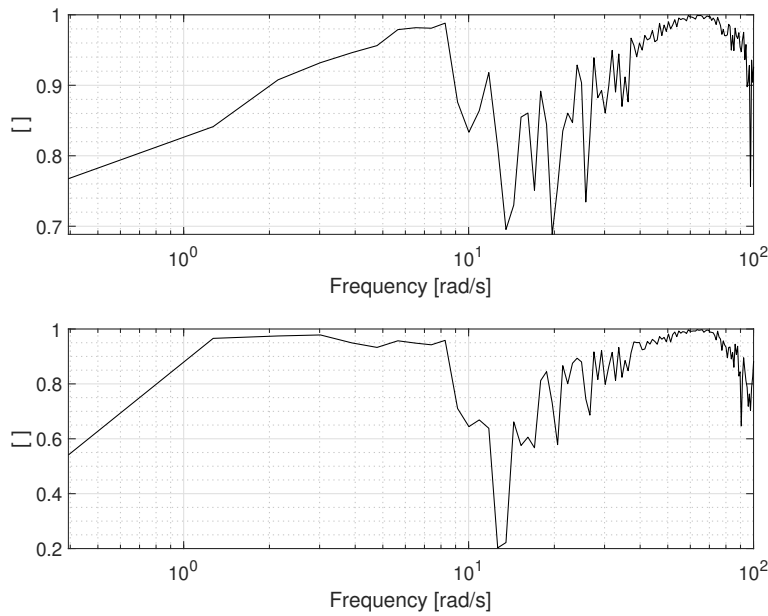


Figure 4.24: Coherence function of the lateral non-parametric frequency response functions: roll rate p (top), lateral acceleration a_y (bottom).

4.5 Model matching

In this final section, model matching was performed starting from the identified unstructured models. Obtained models are analyzed, comparing them with the output-error ones. As for the output-error method, different initial conditions for the physical parameters were tested to distinguish between local and global minima. Best results are related to the open-loop stable systems.

4.5.1 Directional dynamics

The case around the z axis is the simplest one, since it is constituted by a SISO system and parameters to be identified are two: the stability and control derivatives. Results are reported in Figure 4.25 and in Table 4.11, along with those obtained with output-error model identification. The two models are very similar both in terms of physical parameters and VAF metric.

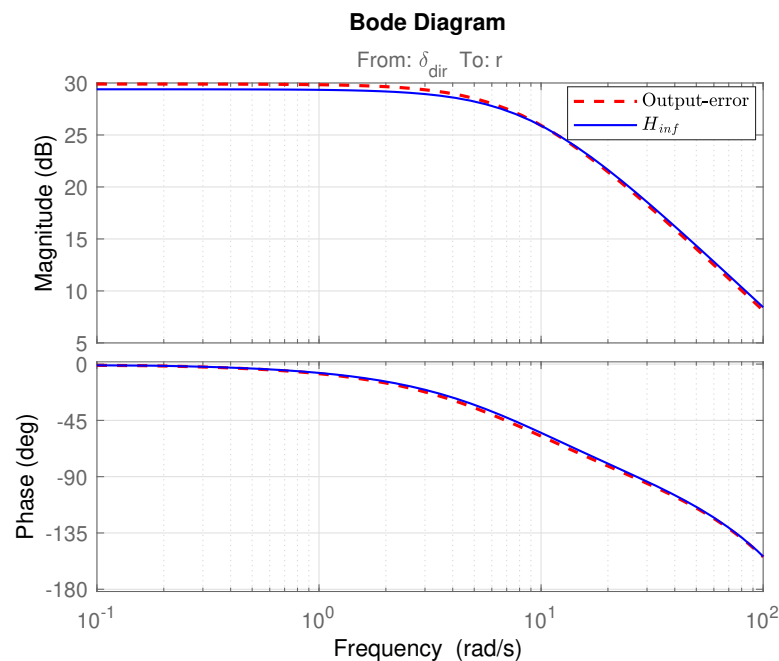


Figure 4.25: Directional model Bode diagram comparison between output-error and model-matching approaches: yaw rate.

4.5.2 Vertical dynamics

Regarding the vertical axis, very satisfactory results were obtained with model-matching identification both in terms of physical parameters and VAF: Bode diagram can be found in Figure 4.26 and the obtained parameters with the validation

Parameter	Output-error	Model matching
N_r	-8.178 1/s	-8.989 1/s
N_δ	255.590 rad/s ²	265.100 rad/s ²
VAF	81.736 %	81.307 %

Table 4.11: Comparison of directional identified parameters.

metrics are reported in Table 4.12. Also in this case, the two identified models are nearly the same.

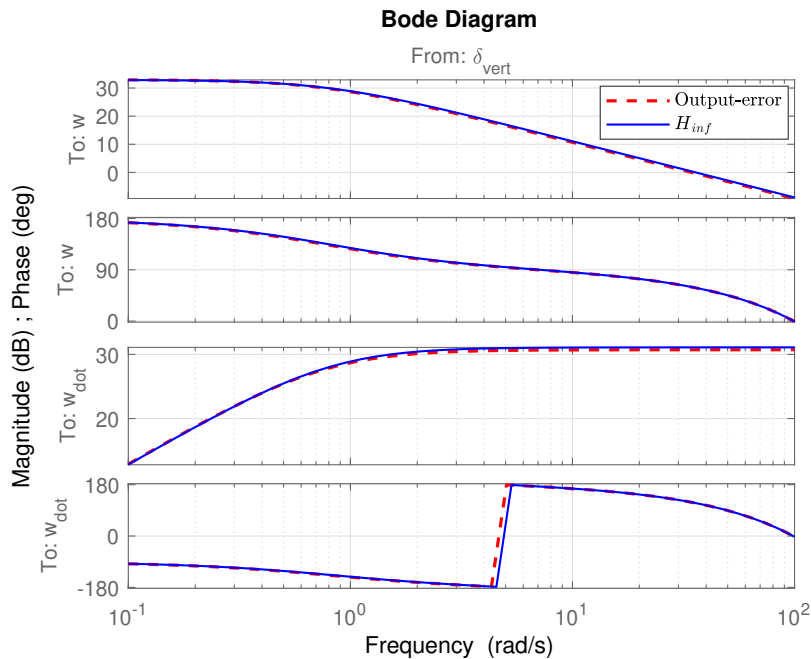


Figure 4.26: Vertical model Bode diagram comparison between output-error and model-matching approaches: vertical velocity (top) and vertical velocity derivative (bottom).

4.5.3 Lateral and longitudinal dynamics

The two remaining x and y axes are more complex since they have more unknown variables to be tuned and it must be taken care when dealing with unstable systems using *sysstune*. Unlike previous cases, physical parameters are more different than those obtained using the output-error approach, even if the identified models with model matching proved to be very similar. Results are shown in Table 4.13 for the lateral plane and in Table 4.14 for the longitudinal plane. Moreover, a comparison of the obtained lateral dynamic models with the output-error and the model matching approaches is reported in Figure 4.27 as an example. In these

Parameter	Output-error	Model matching
Z_w	-0.731 1/s	-0.776 1/s
Z_δ	-34.352 m/s ²	-34.385 m/s ²
VA_{F_w}	61.972 %	61.659 %
$VA_{F_{\dot{w}}}$	86.319 %	86.267 %

Table 4.12: Comparison of vertical identified parameters.

Parameter	Output-error	Model matching
Y_v	-0.264 1/s	-0.211 1/s
Y_δ	9.568 m/s ²	9.379 m/s ²
L_v	-7.349 rad s/m	-3.613 rad s/m
L_δ	1079.339 rad/s ²	851.287 rad/s ²
VA_{F_p}	93.761 %	93.539 %
$VA_{F_{a_y}}$	67.110 %	66.955 %

Table 4.13: Comparison of lateral identified parameters.

cases the physical parameters were too much sensitive to the choice of the initial unstructured model (therefore of n , p and f values) and of the focus imposed during the model matching. For these reasons, even if the obtained models were very accurate, these physical parameters were not considered reliable.

4.6 Uncertainty analysis

Before moving on to the flight-controller design, a more detailed analysis of the uncertainty parameters of the complete grey-box model must be done. As seen in Chapter 4.3, output-error model identification also gives as result the standard deviations of the quadrotor stability and control derivatives. The identified physical parameters were perturbed considering a Gaussian distribution and their standard deviations and the obtained models are shown in the frequency domain in Figure 4.28, Figure 4.29, Figure 4.30, Figure 4.31, Figure 4.32, Figure 4.33,

Parameter	Output-error	Model matching
X_u	-0.234 1/s	-0.282 1/s
X_δ	-10.237 m/s ²	-8.758 m/s ²
M_u	7.525 rad s/m	6.2967 rad s/m
M_δ	701.578 rad/s ²	473.0611 rad/s ²
VA_{F_q}	95.860 %	94.340 %
$VA_{F_{a_x}}$	61.415 %	66.660 %

Table 4.14: Comparison of longitudinal identified parameters.

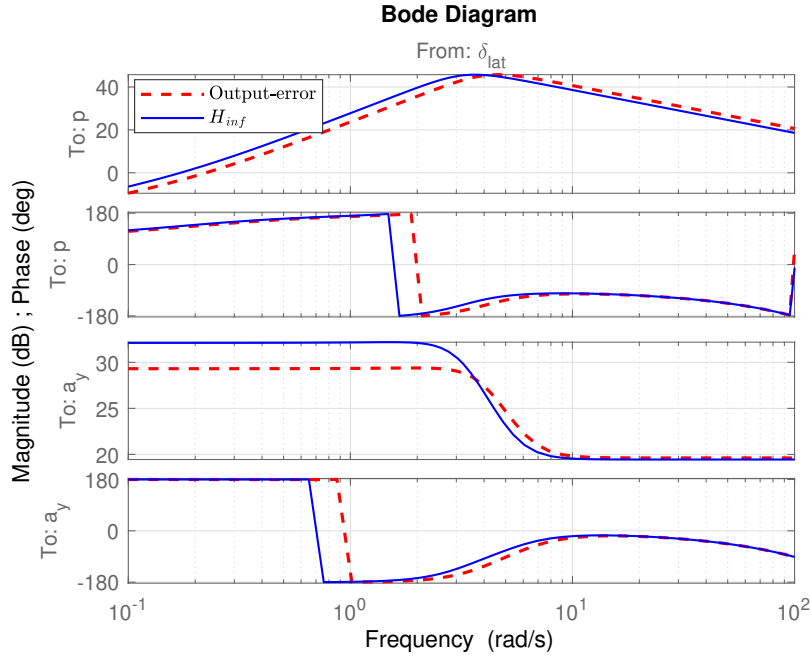


Figure 4.27: Lateral model Bode diagram comparison between output-error and model-matching approaches: roll rate p (top) and lateral acceleration a_y (bottom).

Figure 4.34 and Figure 4.35 (in all cases, a vector of 500 elements were obtained for each physical parameter). Results confirmed the accuracy of the identified models, especially in the interested frequency ranges. Higher dispersion must be observed in the directional pole-zero map (Figure 4.32). Finally, these models will be used to perform a robustness analysis of the controllers in Chapter 6.

4.7 Conclusions

In conclusion, system identification has been performed using the subspace model identification, the output-error model identification and a new model-matching technique. Both structured and unstructured identified models for all the axes were derived and discussed, proving the instability nature of the quadcopter around the x and y axes. All the structured models represented well the physics of the ANT-R, with simulations fitting the real data with very high accuracy. Moreover, an uncertainty analysis confirmed the fidelity of the obtained models. Finally, results agree with the existent literature (see [11], [14], [15] and [12] as examples). For these reasons, the complete plant model obtained with the output-error approach has been used for the design of the flight controllers discussed in Part II.

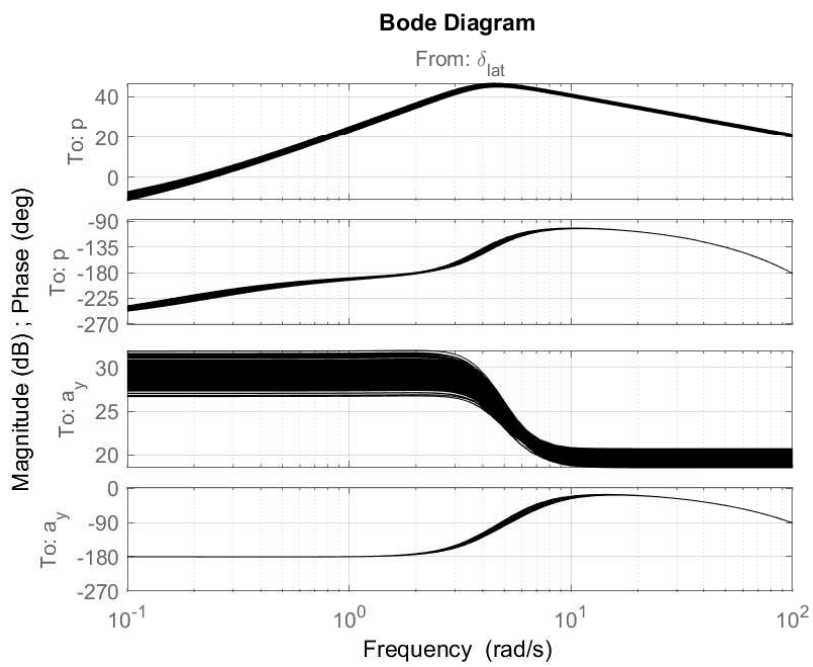


Figure 4.28: Grey-box uncertainty analysis of lateral dynamics: Bode diagram.

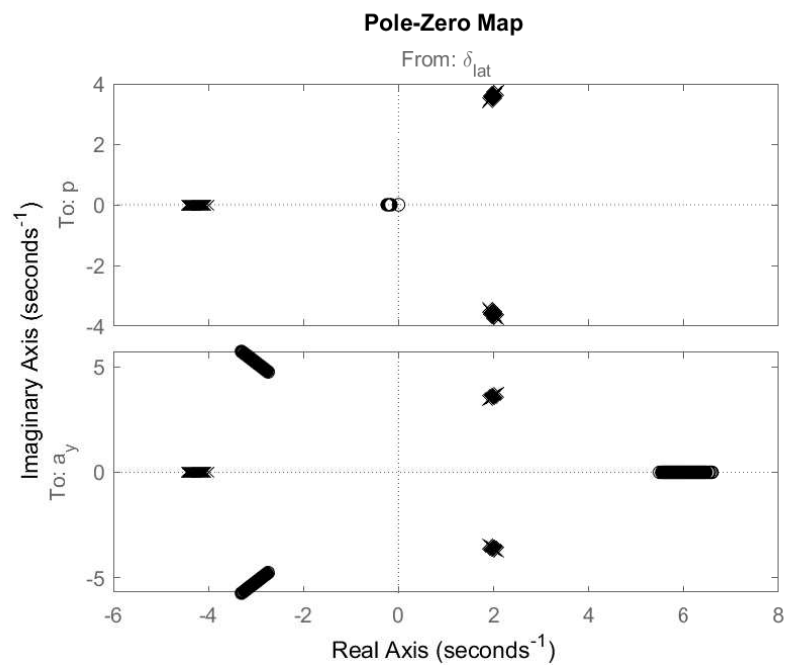


Figure 4.29: Grey-box uncertainty analysis of lateral dynamics: pole-zero map.

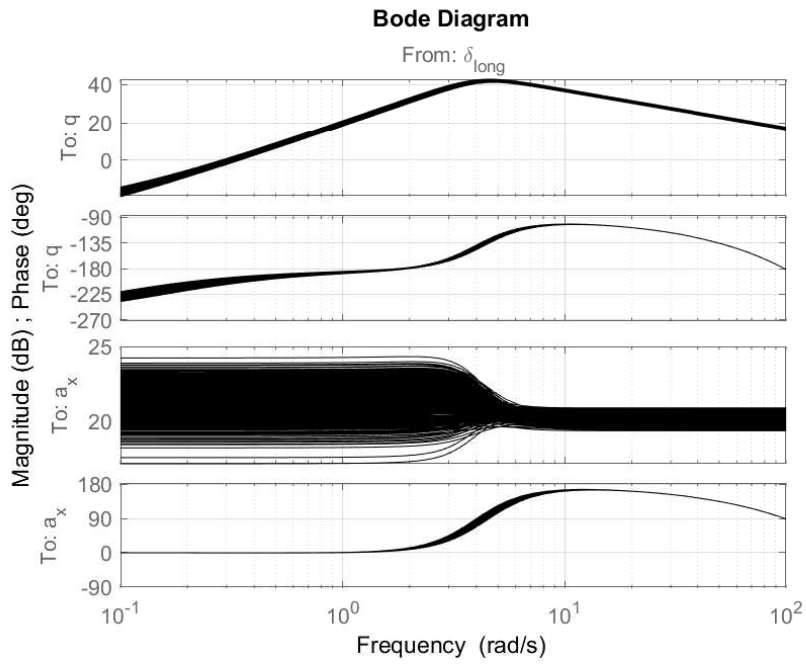


Figure 4.30: Grey-box uncertainty analysis of longitudinal dynamics: Bode diagram.

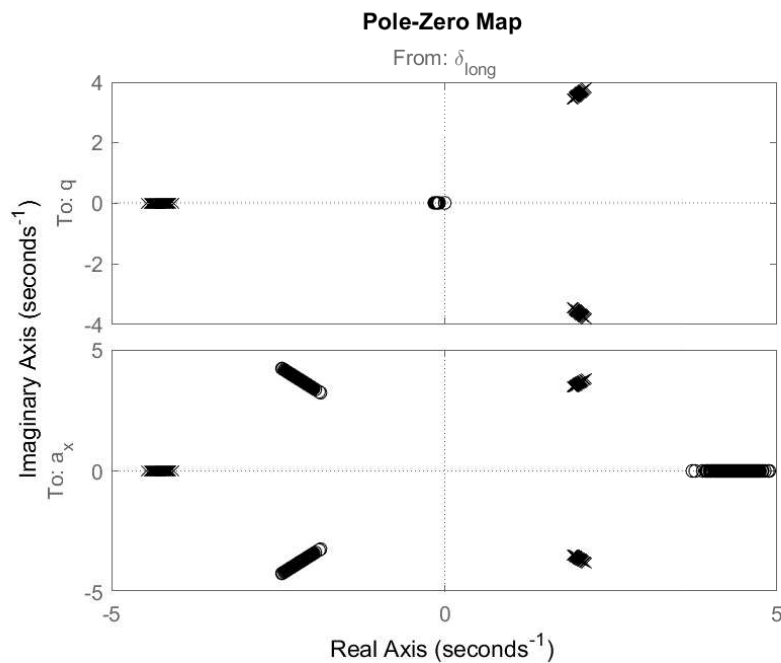


Figure 4.31: Grey-box uncertainty analysis of longitudinal dynamics: pole-zero map.

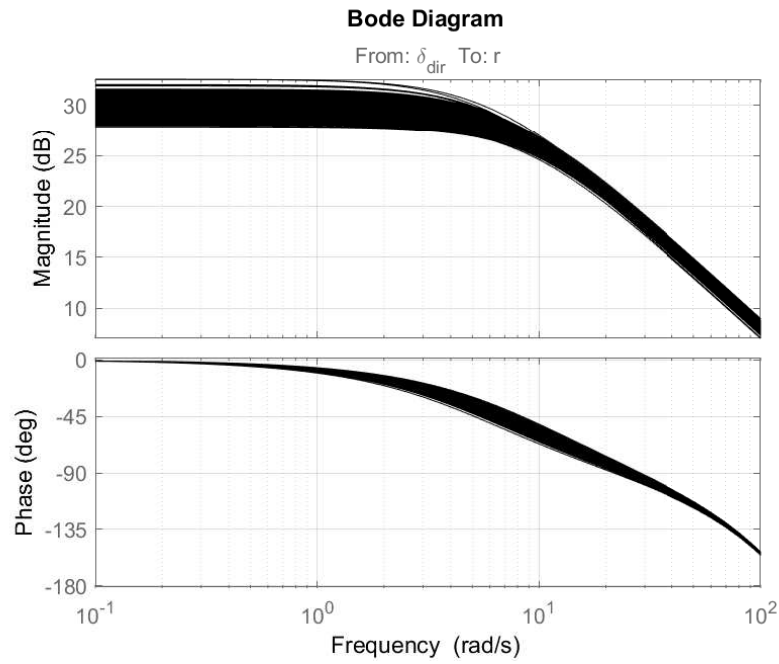


Figure 4.32: Grey-box uncertainty analysis of directional dynamics: Bode diagram.

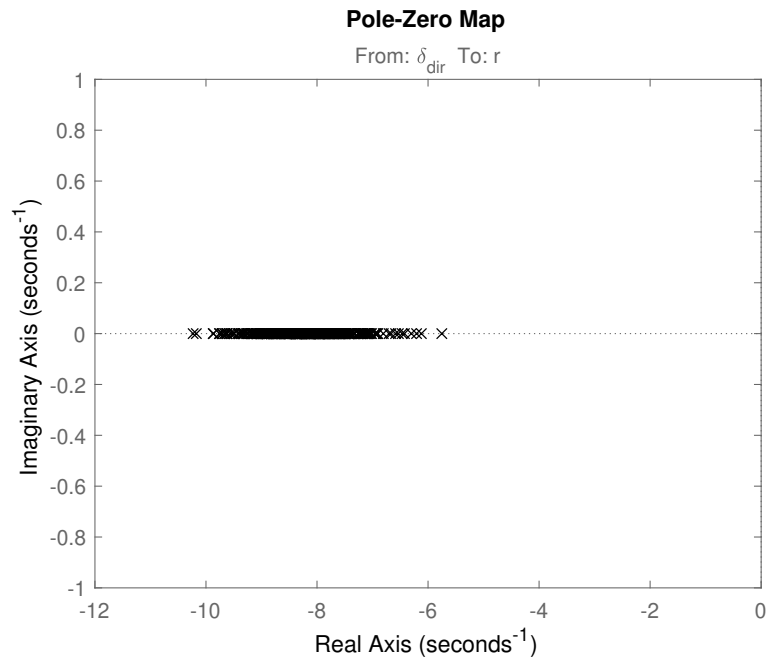


Figure 4.33: Grey-box uncertainty analysis of directional dynamics: pole-zero map.

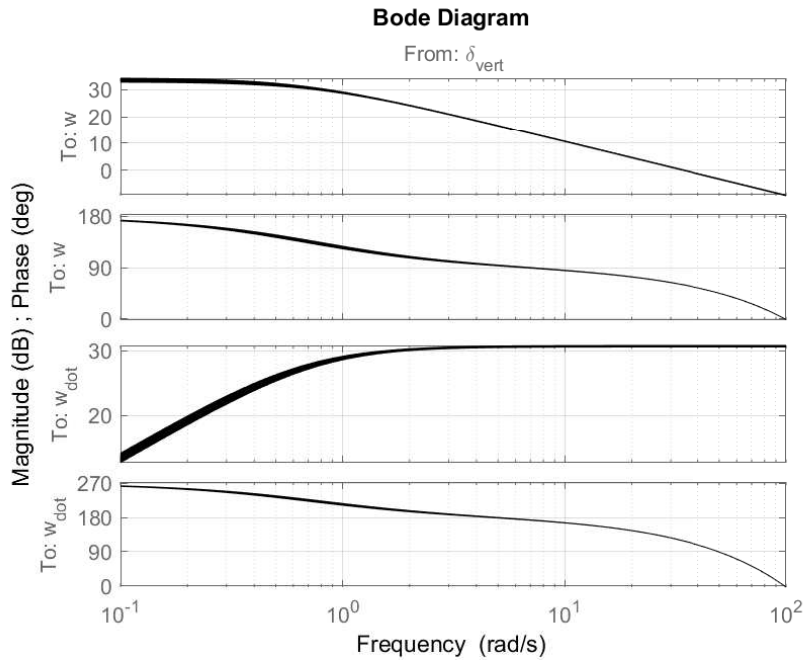


Figure 4.34: Grey-box uncertainty analysis of vertical dynamics: Bode diagram.

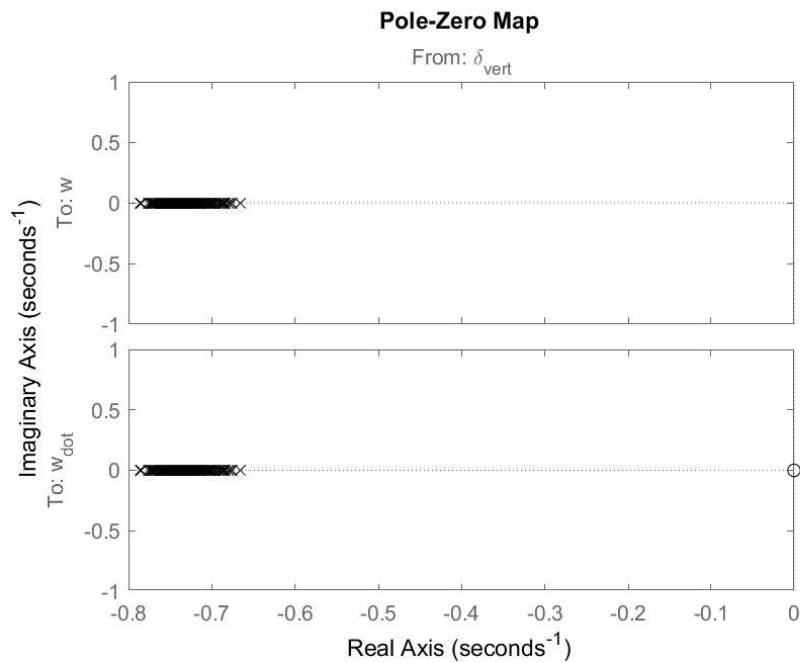


Figure 4.35: Grey-box uncertainty analysis of vertical dynamics: pole-zero map.

Part II

Quadrotor dynamics control

Chapter 5

Dynamic inversion and explicit model following control laws

The second part of the thesis deals with the problem of quadrotor control: this type of small aircraft is affected by non-linearity and instability; it is provided with four propellers which must control six degrees of freedom (three translations and three rotations around each axis), resulting as an underactuated system. For these reasons and for the increasing interest in unmanned aerial vehicles, high-performance control systems are required.

As a part of an accurate model to simulate the real system, the identification phase gave as results an estimate of the quadrotor physical parameters, which include the dimensional stability and control derivatives with respect to states and control inputs. The idea in fact is to replace the stock PID architecture of the ANT-R quadcopter with model inversion control laws: the Explicit Model Following (EMF) and the Dynamic Inversion (DI) control laws. These control systems have been much studied on full size and mid size helicopters in the last two decades (see [24], [33] and [34] for examples). On the contrary, few studies have been done on small scale quadrotors (see [11] and [15]).

The procedure for the design of EMF and DI control laws will be described. Then, flight controllers will be simulated with the identified model, implemented on the quadcopter flight control unit and tested in flight, comparing the results of the different architectures.

5.1 Complete plant model

The identification of the ANT-R quadrotor position and attitude dynamics (see Chapter 4) has been performed using Subspace Model Identification, output-error model identification and with a new model matching technique, considering as input the control actuators input, which is the sum of the injected excitation signal and the feed-back term, angular rates and accelerations as outputs for

lateral and longitudinal dynamics, yaw rate only for directional dynamics and vertical velocity with its derivative for the z axis.

The following complete plant model has been obtained by combining the models obtained with the output-error approach on each axis, since couplings have been neglected at low speed, see the state space system at equation (5.1), the state space matrices at equation (5.2) and equation (5.3), the state and input vectors at equation (5.4) and equation (5.5) respectively. The identified plant model proved to be minimum phase, unstable around x and y axes and accurate in all validation tests. Identified stability and control derivatives values with their percentage standard deviations are shown in Table 5.1, differences between lateral and longitudinal parameters are due to the asymmetry of the considered platform. Moment derivatives with respect to pitch and roll rate (M_q and L_p) were neglected since the associated uncertainty values were considered too large. These results are in agreement with the literature, see [11], [15] and [12].

$$\dot{x} = Ax + Bu \quad (5.1)$$

$$A = \begin{bmatrix} Y_v & Y_p & g & 0 & 0 & 0 & 0 & 0 & 0 \\ L_v & L_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_u & X_q & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & M_u & M_q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_w \end{bmatrix} \quad (5.2)$$

$$B = \begin{bmatrix} Y_\delta & 0 & 0 & 0 \\ L_\delta & 0 & 0 & 0 \\ 0 & X_\delta & 0 & 0 \\ 0 & M_\delta & 0 & 0 \\ 0 & 0 & N_\delta & 0 \\ 0 & 0 & 0 & Z_\delta \end{bmatrix} \quad (5.3)$$

$$x = [v \ p \ \phi \ u \ q \ \theta \ r \ \psi \ V_z]^\top \quad (5.4)$$

$$u = [\delta_{lat} \ \delta_{long} \ \delta_{dir} \ \delta_{ver}]^\top. \quad (5.5)$$

5.2 ANT-R stock flight controller

The quadrotor considered for identification and control has one of the most common feedback controller in multi-rotors aircraft: a cascade PID architecture. Both position and attitude dynamics controls are composed by two closed-loops: the

Parameter	Value	Unit	$\sigma\%$
Y_v	-0.264	1/s	4.837
Y_p	0	m/(s rad)	-
L_v	-7.349	rad s/m	4.927
L_p	0	1/s	-
Y_δ	9.568	m/s ²	4.647
L_δ	1079.339	rad/s ²	2.762
X_u	-0.234	1/s	4.651
X_q	0	m/(s rad)	-
M_u	7.525	rad s/m	4.353
M_q	0	1/s	-
X_δ	-10.237	m/s ²	3.204
M_δ	701.578	rad/s ²	2.573
N_r	-8.178	1/s	8.289
N_δ	255.590	rad/s ²	3.679
Z_w	-0.731	1/s	2.497
Z_δ	-34.351	m/s ²	0.401

Table 5.1: Complete plant identified parameters.

outer loop has a proportional gain $K_{P_{outer}}$ on the tracking error between the set-point and the feedback term (angles or positions), while the inner loop is a PID for the angular rates and the velocities respectively:

$$PID(s) = K_P + sK_D + \frac{1}{s}K_I \quad (5.6)$$

where K_P , K_D and K_I are respectively the proportional, derivative and integral gains. Regulator parameters are reported in Table 5.2 and Table 5.3 for all axes. These gains were calibrated starting from stock values to obtain satisfactory performance in terms of response, tracking accuracy and disturbance rejection; a final trial and error manual calibration was performed in flight. Considering the lateral attitude dynamics as example, a block diagram of the closed-loop attitude dynamics is shown in Figure 5.1. In the figure, the input of block UAV is the difference between opposite rotational motors speeds δ_Ω , which is obtained by multiplying the mixer matrix and the moment δ_L , which is the output of the PID controller (for more details about the mixer matrix see Chapter 1); e_ϕ is the error between the roll angle set-point and the measured signal (which is obtained integrating the roll rate p); e_p is the error between the desired roll rate (obtained as output of the proportional gain P_{inner}) and the measurement.

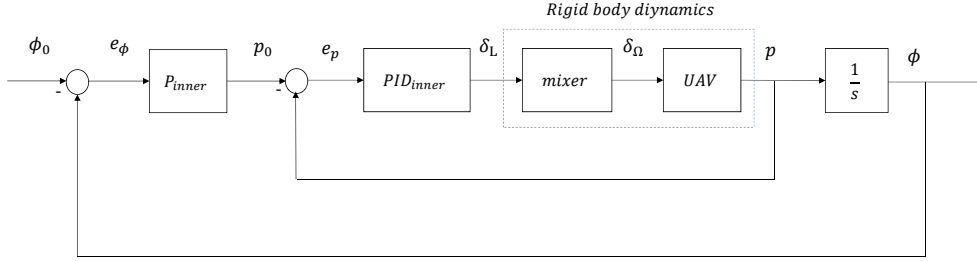


Figure 5.1: ANT-R stock flight controller: attitude dynamics.

Parameter	y	ϕ	x	θ	ψ	z
$K_{P_{outer}}$	0.95	6.5	0.95	6	2.8	1

Table 5.2: ANT-R stock flight controller: proportional gains.

5.3 Dynamic Inversion control law design

In this section the DI controller will be described in detail. DI inverts the equations which were used to obtain the plant model using feedback linearization. The procedure for the design of the control law (see [11] and [24]) can be divided in three steps:

1. Choice of the state vector x , input u and output y :

$$\dot{x} = Ax + Bu \quad (5.7)$$

$$y = Cx \quad (5.8)$$

$$x \in \mathbb{R}^n, u \in \mathbb{R}^m, y \in \mathbb{R}^m. \quad (5.9)$$

DI can deal with square MIMO systems of any order as long as measurements of the states exist for feedback.

2. Differentiation of the output equation until the explicit dependence of the control is observed in the output:

$$\dot{y} = CAx + CBu. \quad (5.10)$$

Parameter	V_y	p	V_x	q	r	V_z
$K_{P_{inner}}$	0.06	0.05	0.06	0.07	0.08	0.2
$K_{I_{inner}}$	0.02	0.05	0.02	0.05	0.1	0.02
$K_{D_{inner}}$	0.001	0.001	0.001	0.001	0	0

Table 5.3: ANT-R stock flight controller: PID parameters.

3. Inversion of the output equation

$$u = (CB)^{-1}(v - CAx) \quad (5.11)$$

$$v = \dot{y}_{cmd_f} + Ke \quad (5.12)$$

$$e = y_{cmd_f} - y \quad (5.13)$$

where v is the pseudo command vector and K is the compensator of e , which is the difference between the filtered desired output y_{cmd_f} and the measured output y .

The architecture instead consists in three blocks:

- feedback to achieve model inversion;
- command filter: the reference signal is filtered to obtain a smoother response to pilot command;
- feed-back compensation to govern disturbance rejection.

A schematic of the DI flight control system is shown in Figure 5.2.

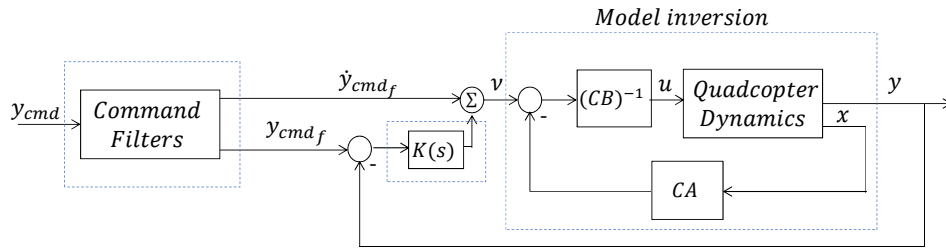


Figure 5.2: Dynamic Inversion block diagram.

5.3.1 Stability issues

The main problem of Dynamic Inversion could be the presence of zeros in the right-half complex plane (non minimum phase, NMP zeros). If equation (5.11) is substituted in equation (5.7):

$$\dot{x} = (I - B(CB)^{-1}C)Ax + B(CB)^{-1}v \quad (5.14)$$

it can be observed that the eigenvalues of the inverted system are the zeros of the initial one. Therefore, if the identified model is NMP, the controlled system is unstable. To solve this problem, the Approximate Dynamic Inversion method or simple inputs-outputs redefinition can be used (see [24] for more details).

5.3.2 Inner loop: attitude

The inner loop controls the pitch, roll, yaw and vertical dynamics. Following the procedure in the previous section:

1. States, inputs and outputs are chosen for the inner loop:

$$x = [p \ \phi \ q \ \theta \ r \ w]^\top \quad (5.15)$$

$$u = [\delta_{lat} \ \delta_{long} \ \delta_{dir} \ \delta_{vert}]^\top \quad (5.16)$$

$$y_{cmd} = [\phi_{cmd} \ \theta_{cmd} \ r_{cmd} \ V_{z_{cmd}}]^\top \quad (5.17)$$

$$y = [\phi \ \theta \ r \ V_z]^\top. \quad (5.18)$$

2. A partition of matrix C is needed to see the control in the output equation since pitch and roll angle equations must be differentiated twice while the yaw rate and vertical velocity equations only once:

$$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (5.19)$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.20)$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.21)$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \dot{r} \\ \dot{V}_z \end{bmatrix} = \begin{bmatrix} C_1 \hat{A}^2 x + C_1 \hat{A} \hat{B} u \\ C_2 \hat{A} x + C_2 \hat{B} u \end{bmatrix} \quad (5.22)$$

where \hat{A} and \hat{B} are the modified state space matrices:

$$\hat{A} = \begin{bmatrix} L_p & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_q & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & N_r & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_w \end{bmatrix} \quad (5.23)$$

$$\hat{B} = \begin{bmatrix} L_\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & M_\delta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & N_\delta & 0 \\ 0 & 0 & 0 & Z_\delta \end{bmatrix}. \quad (5.24)$$

3. Inversion of the output equation:

$$u = \begin{bmatrix} C_1 \hat{A} \hat{B} \\ C_2 \hat{B} \end{bmatrix}^{-1} \left(\nu - \begin{bmatrix} C_1 \hat{A}^2 \\ C_2 \hat{A} \end{bmatrix} x \right) \quad (5.25)$$

with the pseudo-command vector ν defined as:

$$\begin{bmatrix} \nu_\phi \\ \nu_\theta \\ \nu_r \\ \nu_{V_z} \end{bmatrix} = \begin{bmatrix} \ddot{\phi}_{cmd} \\ \ddot{\theta}_{cmd} \\ \dot{r}_{cmd} \\ \dot{V}_{zcmd} \end{bmatrix} + \begin{bmatrix} e_\phi \\ e_\theta \\ e_r \\ e_{V_z} \end{bmatrix} K_P + \begin{bmatrix} \dot{e}_\phi \\ \dot{e}_\theta \\ 0 \\ 0 \end{bmatrix} K_D + \begin{bmatrix} \int e_\phi \\ \int e_\theta \\ \int e_r \\ \int e_{V_z} \end{bmatrix} K_I \quad (5.26)$$

and

$$e = y_{cmd} - y. \quad (5.27)$$

The block diagram of the inner loop is illustrated in Figure 5.3, which was obtained with equations (5.25) and (5.26). A second order command filter of the second order was chosen for pitch and roll angle and PID regulators for feed-back compensation. For the yaw rate and the vertical velocity, a first order filter and PI compensator were used.

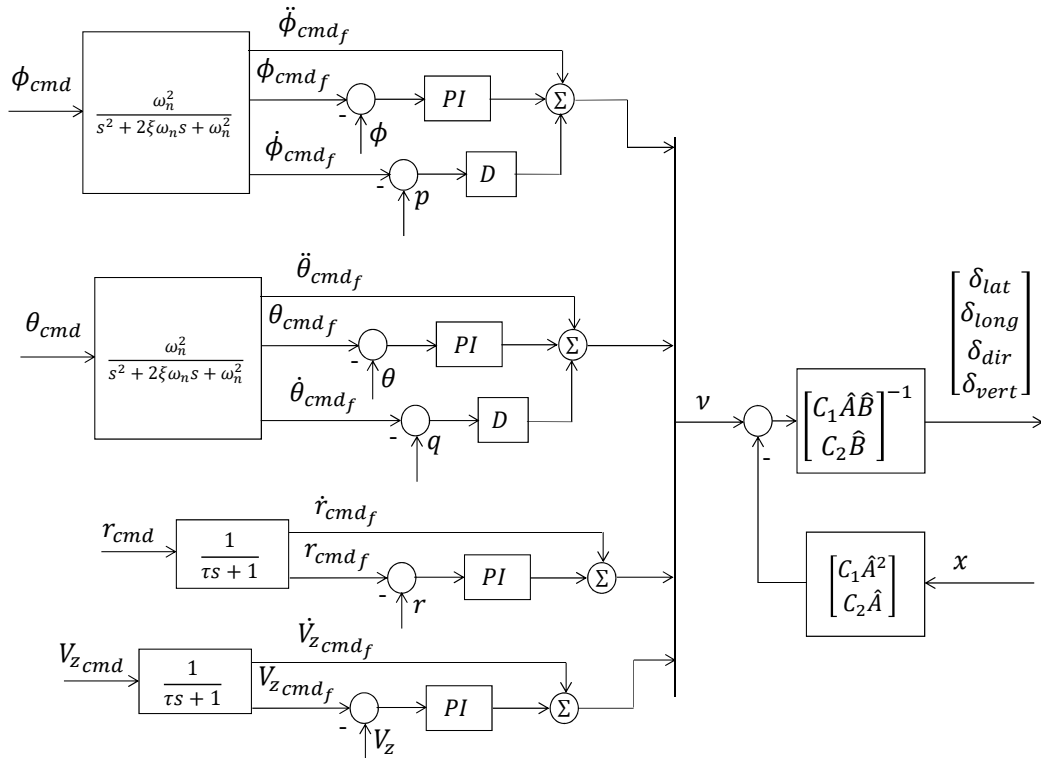


Figure 5.3: Dynamic Inversion block diagram: inner loop.

5.3.3 Outer loop: velocity

The outer loop tracks the lateral and the longitudinal velocities. Usually, these quantities are commanded in the Earth fixed frame (NED), while equations of motion are written in the body frame. A rotation matrix T_{EB} from the body frame F_B to NED frame F_E is so introduced in the flight control scheme:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = T_{EB} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5.28)$$

$$T_{EB} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \phi \sin \theta - \sin \psi \cos \phi & \cos \psi \cos \phi \sin \theta + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (5.29)$$

(See Chapter 1 for more details about three dimensional rotations). Now the DI law for the outer loop can be designed:

1. The vectors of states, inputs and outputs are:

$$x^\top = [u \ v], \quad u^\top = [\theta_{cmd} \ \phi_{cmd}] \quad (5.30)$$

$$y^\top_{cmd} = [u_{cmd} \ v_{cmd}], \quad y^\top = [u \ v]. \quad (5.31)$$

2. Considering the lateral and longitudinal equations of motion, the output equations are:

$$\dot{v} = Y_v v + g \phi_{cmd} \quad (5.32)$$

$$\dot{u} = X_u u - g \theta_{cmd}. \quad (5.33)$$

3. Equations (5.33) and (5.32) are inverted:

$$\phi_{cmd} = 1/g (\nu_v - Y_v v) \quad (5.34)$$

$$\theta_{cmd} = -1/g (\nu_u - X_u u) \quad (5.35)$$

with the pseudo-command vector defined as:

$$\begin{bmatrix} \nu_u \\ \nu_v \end{bmatrix} = \begin{bmatrix} \dot{u}_{cmd} \\ \dot{v}_{cmd} \end{bmatrix} + \begin{bmatrix} e_u \\ e_v \end{bmatrix} K_P + \begin{bmatrix} \int e_u \\ \int e_v \end{bmatrix} K_I. \quad (5.36)$$

For both lateral and longitudinal planes, a first order command filter of the first order is used to obtain a more desirable reference signal and a PI controller compensates the error between the desired and the measured velocity. Finally, the schematic of the outer loop for the lateral dynamics is shown in Figure 5.4 as an example and was derived from equations (5.36) and (5.34). Note that, since model inversion is written in body axes, set-point velocities must be rotated for consistency.

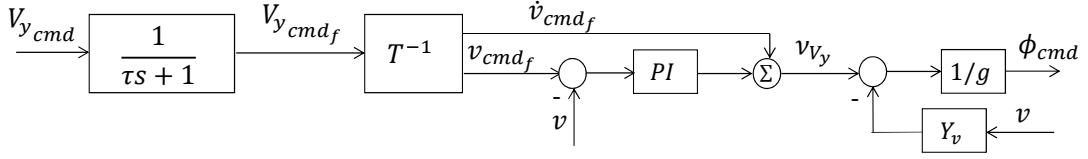


Figure 5.4: Dynamic Inversion block diagram: outer loop, lateral axis.

5.3.4 Error dynamics

The equations of motion which are used to identify and control the quadcopter are based on approximations. Model inversion in fact is not exact and, moreover, disturbances are present. For these reasons, error dynamics is introduced and it will be described in detail how to obtain controller gains for the inner attitude loop and the outer velocity loop starting from the definition of the pseudo-command vector:

$$\nu = \ddot{y}_{cmd} + K_P e + K_D \dot{e} + K_I \int_0^t e d\tau. \quad (5.37)$$

For a DI controller:

$$e^{(n)} = \nu - y^{(n)}_{cmd} \quad (5.38)$$

where n is the order of differentiation. Substituting equation (5.38) in (5.37):

$$\ddot{e} + K_P e + K_D \dot{e} + K_I \int_0^t e d\tau = 0 \quad (5.39)$$

In the Laplace domain:

$$E(s) \left(s^2 + K_D s + K_P + \frac{1}{s} K_I \right) = 0 \quad (5.40)$$

$$s^3 + K_D s^2 + K_P s + K_I = 0. \quad (5.41)$$

Considering a third order system constituted by a pair of complex conjugate poles (introduced with the damping ratio ξ and the natural frequency ω_n) and a real pole p :

$$(s^2 + 2\xi\omega_n s + \omega_n^2)(s + p) = 0 \quad (5.42)$$

$$s^3 + (p + 2\xi\omega_n)s^2 + (2\xi\omega_n p + \omega_n^2)s + \omega_n^2 p = 0. \quad (5.43)$$

It is possible to match equations (5.41) and (5.43):

$$K_D = 2\xi\omega_n + p \quad (5.44)$$

$$K_P = 2\xi\omega_n p + \omega_n^2 \quad (5.45)$$

$$K_I = \omega_n^2 p. \quad (5.46)$$

	ω_n [rad/s]	ξ
Pitch	10	0.9
Roll	10	0.9
Yaw	1	0.9
Heave	1	0.9

Table 5.4: Dynamic Inversion: natural frequencies ω_n and damping ratios ξ for inner loop.

	ω_n [rad/s]	ξ
Lateral velocity	1	0.9
Longitudinal velocity	1	0.9

Table 5.5: Dynamic Inversion: natural frequencies ω_n and damping ratios ξ for outer loop.

Initial values for ω_n and ξ were set equal to those of the command filters while the real pole p was considered equal to $p = \omega_n/5$ for frequency separation from the complex conjugate ones. In a similar way it is possible to derive values for the gains of the PI controllers:

$$K_P = 2\xi\omega_n \quad (5.47)$$

$$K_I = \omega_n^2. \quad (5.48)$$

Initial inner loop natural frequencies and damping ratios are reported in Table 5.4. For the outer loop, ω_n can be set at least at a decade away from the bandwidth of the inner loop, which means: $\omega_{n_{outer}} = \omega_{n_{inner}}/10$ in order to have a slower outer dynamics (see Table 5.5 for outer loop values).

5.4 Explicit Model Following control law design

Another flight controller which uses simplified model inversion is the Explicit Model Following. Unlike the previous one, Explicit Model Following is based on the idea of inverting the equations using feed-forward linearization. The architecture in fact is composed by:

- feed-forward to achieve model inversion;
- command filters in order to have a desirable response for the pilot;
- feed-back compensation.

An example of the block diagram is shown in Figure 5.5. As for Dynamic Inversion, a detailed description of control laws design for the velocity and the attitude loops will be provided.

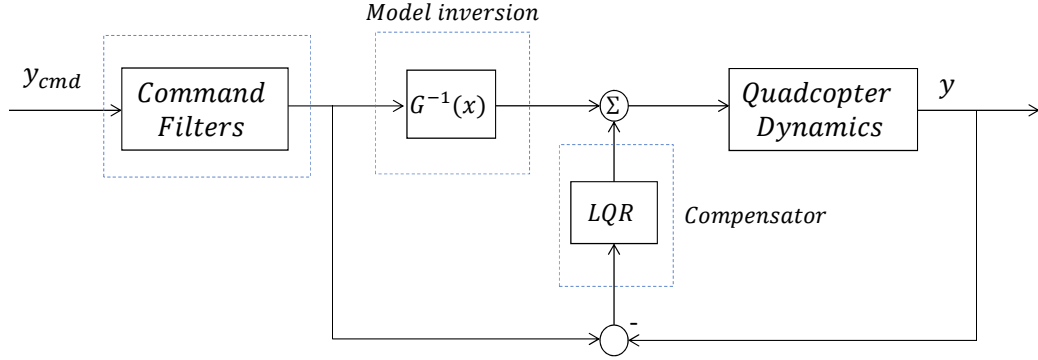


Figure 5.5: Explicit Model Following block diagram.

5.4.1 Inner loop: attitude

In order to construct the inner loop which tracks roll angle, pitch angle, yaw rate and velocity along the z axis, the following inverted transfer functions were obtained:

- Pitch:

$$\frac{\delta_{long}}{\theta} = \frac{s(s - M_q)}{M_\delta}; \quad (5.49)$$

- Roll:

$$\frac{\delta_{lat}}{\phi} = \frac{s(s - L_p)}{L_\delta}; \quad (5.50)$$

- Yaw:

$$\frac{\delta_{dir}}{r} = \frac{s - N_r}{N_\delta}; \quad (5.51)$$

- Heave:

$$\frac{\delta_{vert}}{V_z} = \frac{s - Z_w}{Z_\delta}. \quad (5.52)$$

The architecture of the inner loop can be found in Figure 5.6. Values for command filters are the same as for DI for both inner and outer loops in order to have the same command input. A Linear Quadratic Regulator (LQR) is used as dynamic compensator, see Section 5.4.3 for a detailed description.

5.4.2 Outer loop: velocity

Lateral and longitudinal velocities are controlled independently on each axis using LQR and the following first order decoupled linear models:

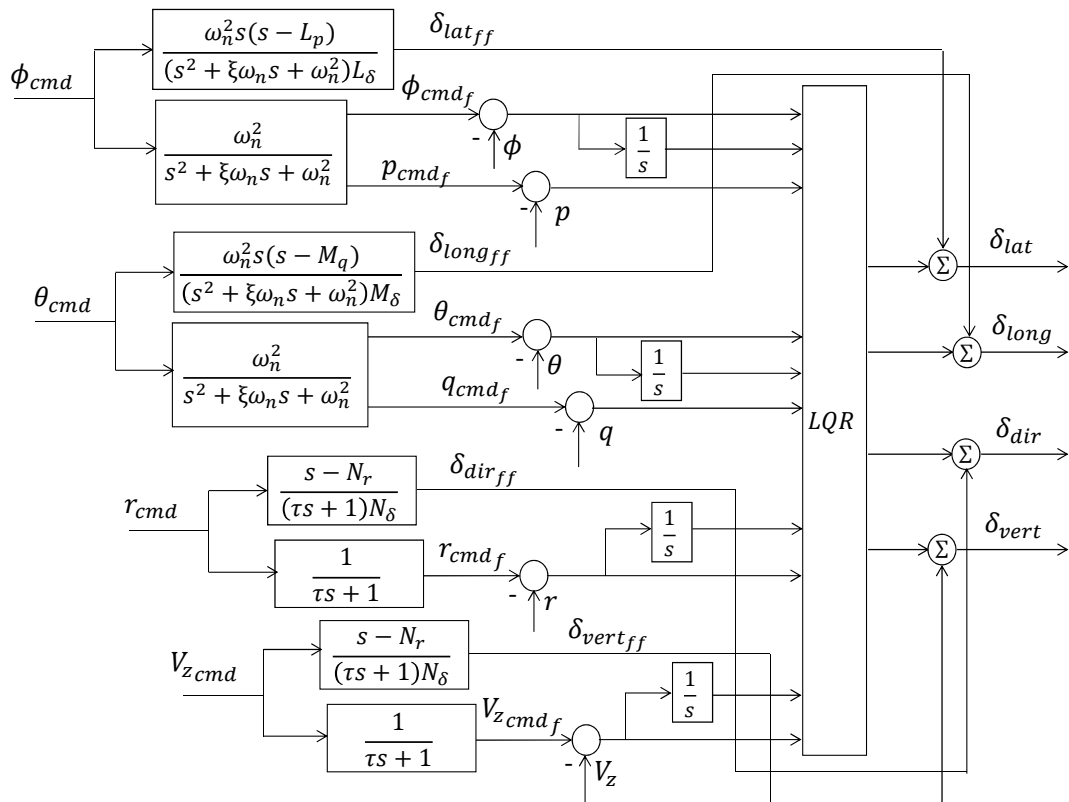


Figure 5.6: Explicit Model Following block diagram: inner loop.

- Longitudinal velocity:

$$\frac{\theta_{cmd}}{V_x} = \frac{s - X_u}{-g}; \quad (5.53)$$

- Lateral velocity:

$$\frac{\phi_{cmd}}{V_y} = \frac{s - Y_v}{g}. \quad (5.54)$$

A schematic of the controller for the lateral velocity is shown in Figure 5.7 as an example.

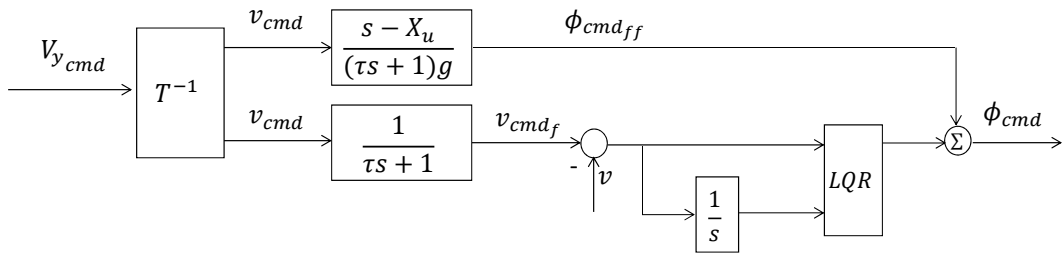


Figure 5.7: Explicit Model Following block diagram: outer loop.

5.4.3 Disturbance rejection

The optimal regulators were obtained minimizing the cost function:

$$J(x, u) = \int_0^t (x^T Q x + u^T R u) d\tau \quad (5.55)$$

where matrices Q and R were related to the largest desired responses and inputs, as suggested in [11]:

$$Q = \text{diag} \left[\frac{\alpha_1^2}{x_{1^2_{max}}} \quad \dots \quad \frac{\alpha_n^2}{x_{n^2_{max}}} \right] \quad (5.56)$$

$$R = \rho \text{diag} \left[\frac{\beta_1^2}{u_{1^2_{max}}} \quad \dots \quad \frac{\beta_m^2}{u_{m^2_{max}}} \right] \quad (5.57)$$

where $x_{i_{max}}$ is the desired maximum displacement from equilibrium of state x_i and similarly $u_{i_{max}}$ is the desired maximum control action displacement from equilibrium of control input u_i , the constant values ρ , α_i , $i = 1, \dots, n$ and β_i , $i = 1, \dots, m$ are additional weights. Values for Linear Quadratic Regulator penalties are reported in Tables 5.6 and Table 5.7 for the attitude loop, Table 5.8 and Table 5.9 for the velocity loop; the α and β parameters were chosen such that $\sum_{i=1}^n \alpha_i^2 = 1$ and $\sum_{j=1}^m \beta_j^2 = 1$, $\rho_{outer} = 0.25$ and $\rho_{inner} = 2$.

Finally, for the inner loop, the state vector and the modified state space matrices for the LQR disturbance rejection are:

$$x_{inner}^T = [p \quad \phi \quad \int \phi \quad q \quad \theta \quad \int \theta \quad r \quad \int r \quad V_z \quad \int V_z] \quad (5.58)$$

$$A_{inner} = \begin{bmatrix} L_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_q & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_w & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.59)$$

$$B_{inner} = \begin{bmatrix} L_d & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & M_d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & N_d & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_d \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \quad (5.60)$$

For the outer loop, the state space matrices used for the LQR regulators are:

$$A_v = \begin{bmatrix} Y_v & 0 \\ 1 & 0 \end{bmatrix}, B_v = \begin{bmatrix} g \\ 0 \end{bmatrix} \quad (5.61)$$

for the lateral plane, and

$$A_u = \begin{bmatrix} X_u & 0 \\ 1 & 0 \end{bmatrix}, B_u = \begin{bmatrix} -g \\ 0 \end{bmatrix} \quad (5.62)$$

for the longitudinal plane.

5.5 Conclusions

In conclusion, Dynamic Inversion and Explicit Model Following control laws for a quadcopter have been described and compared to the most common cascade PID architecture. Initial values for command filters and feedback compensations were provided. The next step will be the simulation and the comparison of the two model inversion control laws.

	x_{imax}
Pitch rate	$600\pi/180$
Pitch angle	$35\pi/180$
Integral pitch angle	$0.05\pi/180$
Roll rate	$600\pi/180$
Roll angle	$35\pi/180$
Integral roll angle	$0.05\pi/180$
Yaw rate	$\pi/180$
Integral yaw rate	$\pi/180$
Vertical velocity	0.1
Integral vertical velocity	1

Table 5.6: Linear Quadratic Regulator: inner loop state penalties.

	u_{imax}
Lateral input	0.001
Longitudinal input	0.001
Directional input	0.001
Vertical input	0.05

Table 5.7: Linear Quadratic Regulator: inner loop input penalties.

	x_{imax}
Lateral velocity	10/4
Integral lateral velocity	10/4
Longitudinal velocity	10/4
Integral longitudinal velocity	10/4

Table 5.8: Linear Quadratic Regulator: outer loop state penalties.

	u_{imax}
Lateral input	$10\pi/180$
Longitudinal input	$10\pi/180$

Table 5.9: Linear Quadratic Regulator: outer loop input penalties.

Chapter 6

Simulator results

Before implementing the Dynamic Inversion and the Explicit Model Following flight controllers on the ANT-R quadcopter, simulations of both the architectures were performed using MATLAB and Simulink [35]. Closed-loop simulation is important since it allows to predict the behavior of the real system, saving time and increasing the safety level of the experimental tests.

In this chapter, the identified model (see Chapter 5.1 and Part I for more details) will be inserted in both flight control architectures to simulate the responses of specific command inputs, studying all the axes and comparing the results. Finally, considering the uncertainty values of the identified physical parameters, a robustness analysis will be performed to study stability and performance of the flight controllers.

The accuracy of the designed simulators will be tested in the next chapter, comparing measured data to simulated data.

6.1 Dynamic Inversion simulator

Following the procedure explained in Chapter 5.3, the architecture of Dynamic Inversion control law is designed in Simulink. In order to control the position and the altitude of the quadrotor, further outer loops for lateral, longitudinal, vertical positions and for yaw angle are added with simple proportional gains which multiply the error between the set-point and the measurement. In this way, the architectures of the two proposed controllers become comparable to the one of the stock controller. All final regulator gains are reported in Table 6.1 (PID values), Table 6.2 (PI values) and Table 6.3 (P values). Moreover, saturation blocks have been inserted to impose limits on specific signals (velocities, angles and angular rates), see Table 6.4 for details. Finally, command filters values are shown in Table 6.5 and Table 6.6; the filter time constants were chosen in relation to the desired response time of the quadcopter along each axis. The block diagram of the complete Dynamic Inversion simulator is shown in Figure 6.1 as an example.

	K_P	K_I	K_D
Roll and pitch angle	136	200	20

Table 6.1: Dynamic Inversion: PID gains.

	K_P	K_I
Longitudinal velocity	1.8	1
Lateral velocity	1.8	1
Vertical velocity	10	1
Yaw rate	30	1

Table 6.2: Dynamic Inversion: PI gains.

Parameter	K_P
Longitudinal position	0.95
Lateral position	0.95
Vertical position	1
Yaw angle	20

Table 6.3: Dynamic Inversion: P gains.

Signal	Max	Min
Longitudinal velocity	3 m/s	-3 m/s
Lateral velocity	3 m/s	-3 m/s
Vertical velocity	1 m/s	-3 m/s
Roll angle	35 deg	-35 deg
Pitch angle	35 deg	-35 deg
Roll rate	600 deg / s	-600 deg / s
Pitch rate	600 deg / s	-600 deg / s
Yaw rate	600 deg / s	-600 deg / s

Table 6.4: Signal saturation values.

Command	τ
Longitudinal velocity	0.5
Lateral velocity	0.5
Vertical velocity	0.4
Yaw rate	0.1

Table 6.5: First order command filter values.

Command	ω	ξ
Roll angle	10	0.9
Pitch angle	10	0.9

Table 6.6: Second order command filter values.

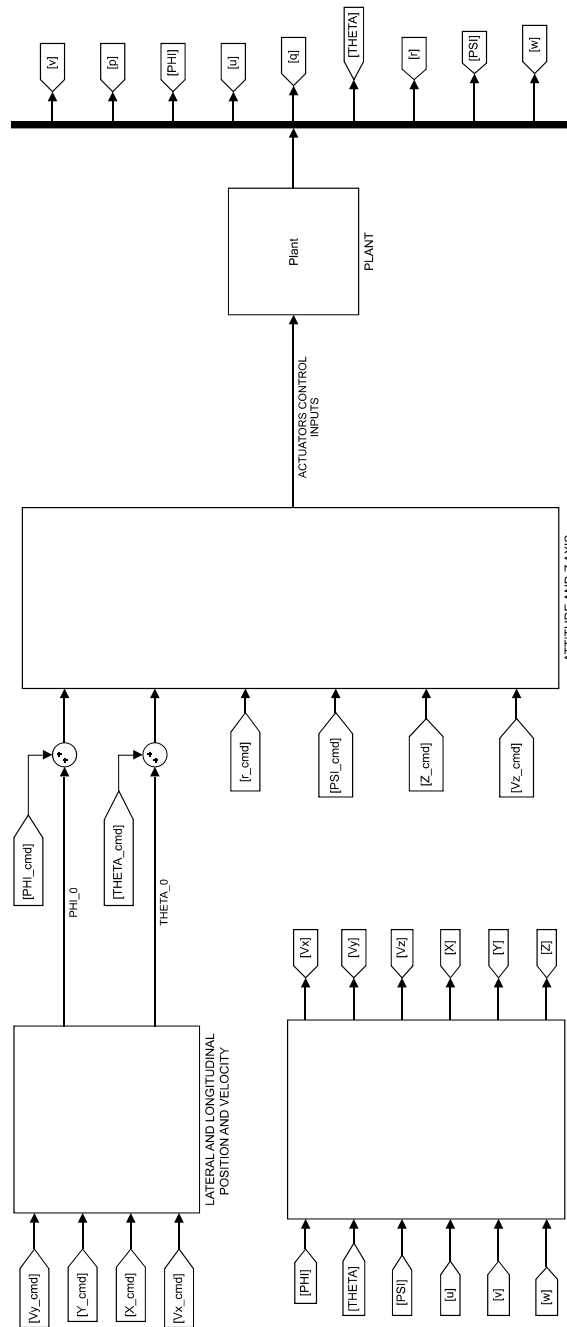


Figure 6.1: Dynamic Inversion simulator: Simulink architecture.

6.2 Explicit Model Following simulator

As for Dynamic Inversion, the procedure described in Chapter 5.4 for the design of the Explicit Model Following flight controller is followed, adding the same DI control outer loops for lateral, longitudinal, vertical positions and for yaw angle. Saturation blocks are also inserted for safety reasons. Command filters are the same chosen for DI controller in order to have the same input command and compare the responses. Linear Quadratic Regulator penalties described in Chapter 5.4.3 gave the following optimal controllers K as results:

$$K_{inner} = \begin{bmatrix} 0.015 & 0.12 & 0.51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.020 & 0.14 & 0.51 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.014 & 0.026 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.21 & -0.022 \end{bmatrix} \quad (6.1)$$

with the state space vector:

$$x_{inner}^T = [p \quad \phi \quad \int \phi \quad q \quad \theta \quad \int \theta \quad r \quad \int r \quad V_z \quad \int V_z] \quad (6.2)$$

for the inner loop, while for the outer loop (K_{V_x} for the longitudinal plane and K_{V_y} for the lateral plane):

$$K_{V_x} = [-0.151 \quad -0.0987] \quad (6.3)$$

$$K_{V_y} = [0.1480 \quad 0.0987] \quad (6.4)$$

with the state space vectors

$$x_{V_x}^T = [V_x \quad \int V_x] \quad (6.5)$$

$$x_{V_y}^T = [V_y \quad \int V_y]. \quad (6.6)$$

6.3 Simulator results comparison

Dynamic Inversion and Explicit Model Following control laws are compared simulating the responses of different set-points. As examples, speed command doublet responses along lateral and longitudinal axes are shown in Figure 6.2 and Figure 6.3 respectively, in order to analyze the complete architectures (both attitude inner loop and velocity outer loop). Considered outputs are the velocity along the specific axis, the angular rate, the angle and the actuators control input, perturbed from the trim condition. Results showed how the two different flight controllers gave similar results considering all the outputs. In both the lateral and longitudinal cases, the output velocity features a slight overshoot with the Explicit Model Following control law. Small differences between the two axes are

present due to the asymmetry of the controlled quadcopter. In fact, very similar values of angles and angular rates are reached during the same maneuvers, but the required applied moment is higher in the longitudinal plane.

Considering the attitude inner loop, angular command doublets around the x , y , z axes and speed command doublets along the z axis were studied. The lateral case is shown as an example in Figure 6.4. Also in this case, the angle response of the EMF controller is slightly different than the DI case. In order to study these differences of the outputs, the actuators control input is analyzed more in detail: this signal is the sum of the model inversion term and the error compensation term. The former is the same for both architectures, since it was constructed using the same equations of motion, the only difference is that the EMF term acts in feedforward while the DI term acts in feedback which contains the measured signal.

6.4 Robustness analysis

The main challenge of Dynamic Inversion and Explicit Model Following is not only the high accuracy of the identified model used to simulate and tune the flight controllers, but also the fidelity of the identified physical parameters of the quadrotor, namely the dimensional stability and control derivatives. During flight in fact, if these parameters differ too much from those of the real system, there will be a feed-back term for DI and a feed-forward term for EMF which will cause disturbances. In simulation this effect can not be seen if the nominal model is used. For this reason a robustness analysis has been performed in the time-domain. The stability and control derivatives of the quadcopter were perturbed considering again their standard deviations and a Gaussian distribution as seen in Chapter 4.6, while the control loops with model inversions remained fixed. Responses to doublet inputs on each axis were simulated 500 times with 500 plant models.

Results of 500 lateral speed command doublet response simulations are shown in Figure 6.5 as an example for the DI control law. As can be seen from the figure, even if the model is perturbed, the outputs remain close to the nominal ones, confirming the robustness of the controller. Very similar results were obtained considering the longitudinal plane and simulating the EMF flight controller with the same procedure.

6.5 Conclusions

The identified model was used to simulate the designed Dynamic Inversion and Explicit Model Following flight controllers, analyzing typical responses to different command inputs. This was an important phase of the work because allowed to

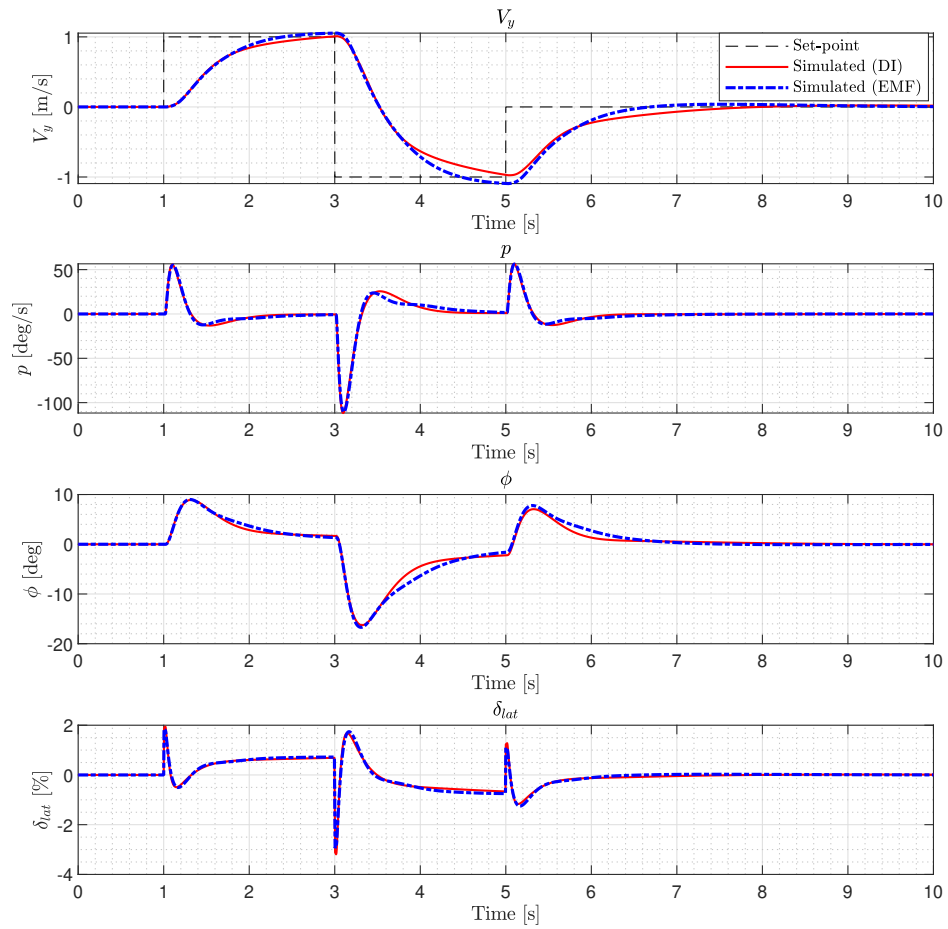


Figure 6.2: Simulation comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , roll rate p , roll angle ϕ and lateral actuators control input δ_{lat}).

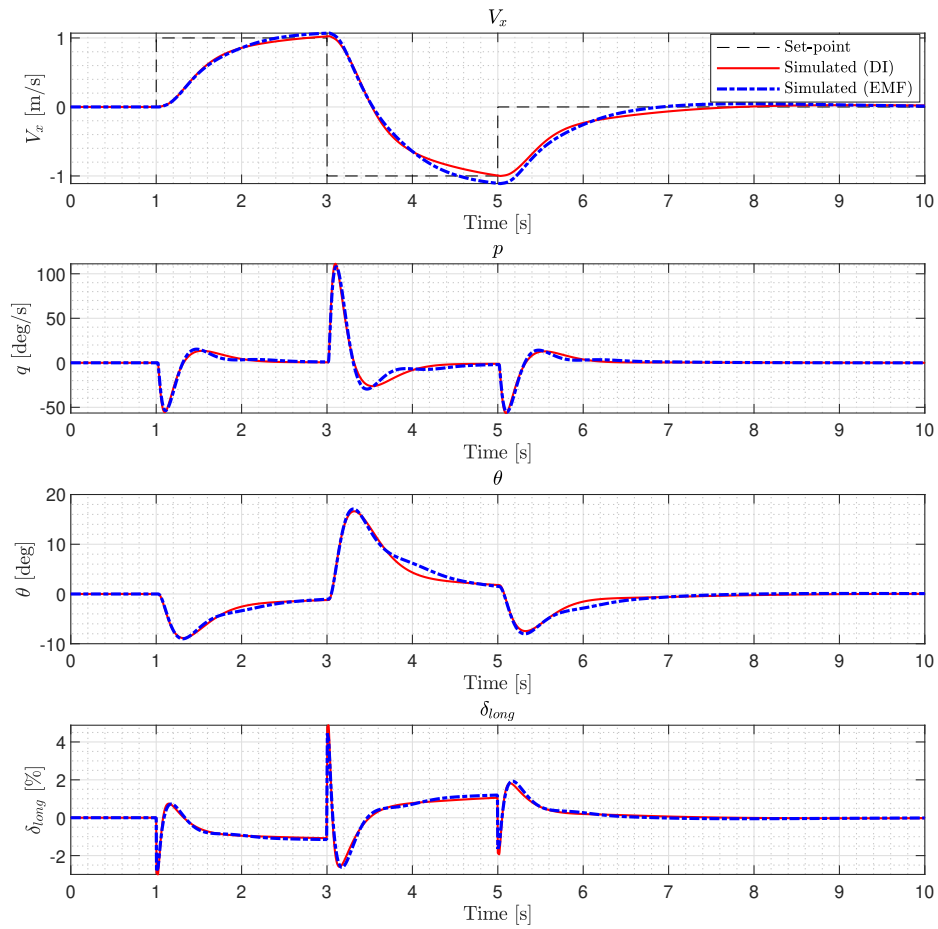


Figure 6.3: Simulation comparison: longitudinal speed command doublet response, from top to bottom (longitudinal velocity V_x , pitch rate q , pitch angle θ and longitudinal actuators control input δ_{long}).

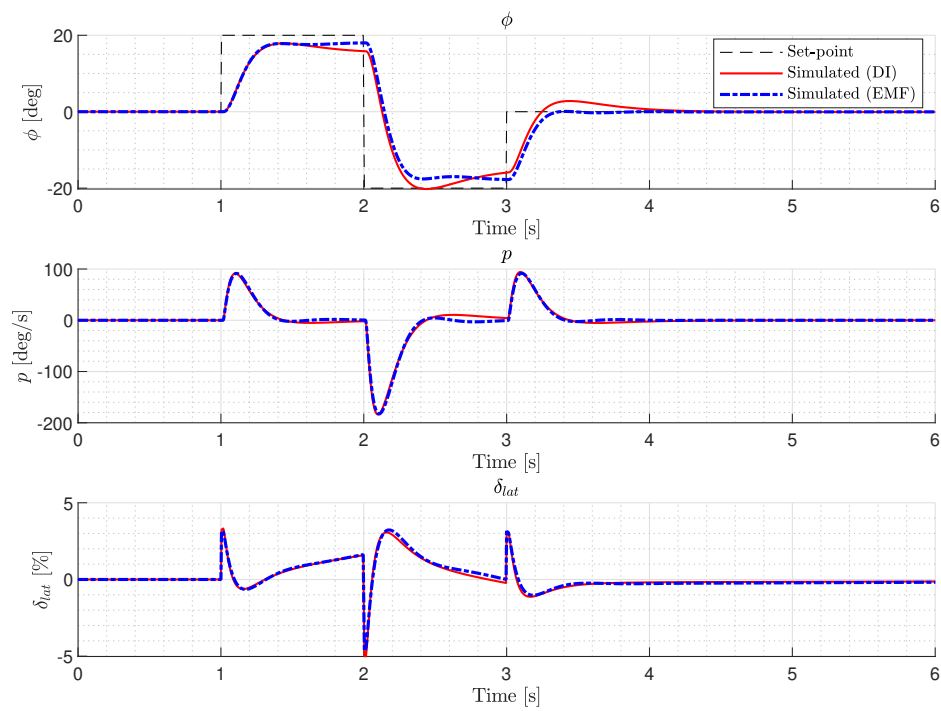


Figure 6.4: Simulation comparison: roll angle command doublet response (from top to bottom: roll angle ϕ , roll rate p and lateral actuators control input δ_{lat}).

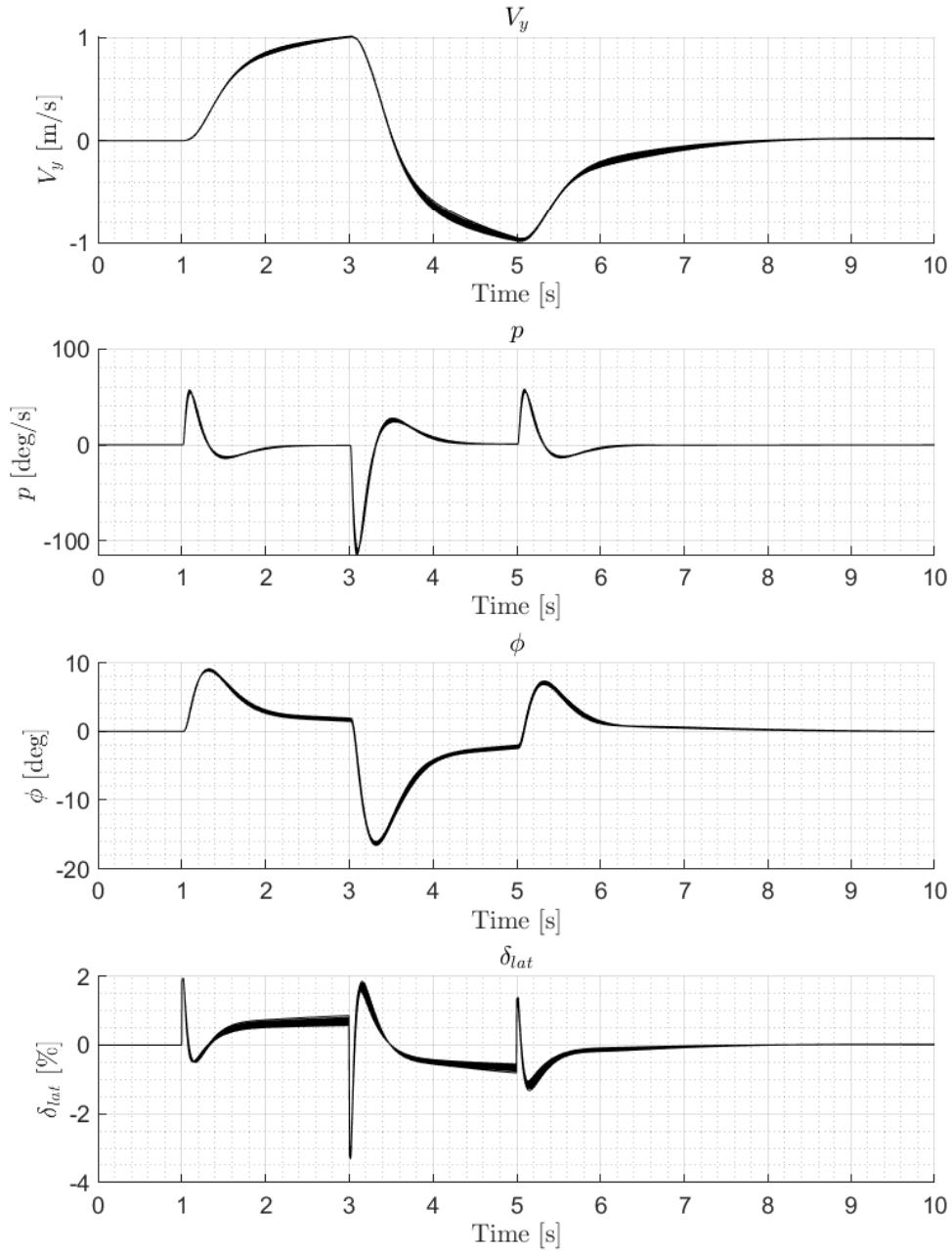


Figure 6.5: Dynamic Inversion robustness analysis: lateral velocity command doublet response (from top to bottom: lateral velocity V_y , roll rate p , roll angle ϕ and lateral actuators control input δ_{lat}).

obtain a preliminary tuning of the controllers, avoiding wasting time with numerous in-flight tests. All final simulation results were satisfactory, demonstrating the stability and the robustness of both the DI and EMF control laws. Last step of the thesis will be to test the controllers in flight, implementing the designed Simulink models on the ANT-R quadrotor flight control unit.

Chapter 7

Experimental results

The last part of the thesis deals with the in-flight experimental tests of the Dynamic Inversion and the Explicit Model Following control laws. At first, an overview of the application used to implement a custom flight control system on the quadcopter will be described. Then, the designed flight controllers will be imported in the ANT-R quadrotor, substituting the stock controller. The two different architectures will be tested and compared in details. In addition, a comparison between measured and simulated data will be provided, to show the level of accuracy of the identified model and the fidelity of the closed-loop simulators. Finally, Dynamic Inversion control law results will be compared to those obtained with the stock ANT-R autopilot.

7.1 Flight controllers design

As seen in Chapter 2, the ANT-R quadrotor uses the Pixhawk Mini as Flight Control Unit (FCU). The FCU hosts the PX4 firmware, an open-source autopilot system; the software can be downloaded from GitHub [20] and users can modify the autopilot at will. A MATLAB Graphic User Interface (GUI) application developed at the Aerospace System and Control Laboratory at Politecnico di Milano (see [36] for more details) was used to compile and import the controller designed in Simulink into the PX4 firmware, which was modified to automate the entire procedure. The Simulink Embedded Coder in fact can automatically generate C++ code from a Simulink model.

The first step of the procedure is to design a Simulink model with the same inputs-outputs defined by the modified PX4 firmware. Then, the model is inserted in a Simulink template and using the application it is possible to automatically generate the C++ code and deploy the new controller to the quadcopter FCU.

In this section, the complete Simulink models of Dynamic Inversion and Explicit Model Following to be tested in flight will be analyzed.

Starting from the Simulink models of DI and EMF control laws used for the

simulations and studied in the previous chapter, some changes were necessary before inserting the models in the template for the generation of the C++ library:

- The Pixhawk Mini flight control unit is able to receive the informations from the optical motion capture system (see Chapter 2). In particular, the information of the quadrotor velocity is expressed in the NED reference system; as it is necessary to have feedback terms expressed in the body-reference frame, a further rotation matrix must be included in the model;
- The identified model was obtained considering as inputs the normalized moments and thrust with respect to the maximum applicable inputs. A normalized mixer matrix is therefore used to obtain the required motors angular speeds Ω , considering the X configuration of the quadrotor platform:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = M_{Mixer} \begin{bmatrix} \delta_{vert} \\ \delta_{lat} \\ \delta_{long} \\ \delta_{dir} \end{bmatrix} \quad (7.1)$$

$$M_{Mixer} = \begin{bmatrix} -1/(4K_T) & -2^{1/2}/(4K_T b) & 2^{1/2}/(4K_T b) & 1/(4K_Q) \\ -1/(4K_T) & 2^{1/2}/(4K_T b) & -2^{1/2}/(4K_T b) & 1/(4K_Q) \\ -1/(4K_T) & 2^{1/2}/(4K_T b) & 2^{1/2}/(4K_T b) & -1/(4K_Q) \\ -1/(4K_T) & -2^{1/2}/(4K_T b) & -2^{1/2}/(4K_T b) & -1/(4K_Q) \end{bmatrix} \quad (7.2)$$

with the constant values $K_T = K_Q = 0.25$ and $b = 2$;

The obtained complete architecture is shown in Figure 7.1. Finally, the Simulink models are imported in the quadcopter using the MATLAB GUI application.

7.2 In-flight tests results

The only problem encountered in flight with both the control laws was a disturbance acting on the yaw angle, causing an uncontrollable rotation around z axis of the quadcopter. This problem was solved by imposing the derivative N_r of the yaw moment with respect to the yaw rate equal to zero. This term in fact generated a positive feed-back component in the Dynamic Inversion control system and a feed-forward component in the Explicit Model Following control system meant to cancel the open-loop yaw dynamics, which implied that the identified physical parameter was not accurate. Imposing N_r equal to zero is in agreement with the literature (see [12] as an example) when the quadrotor is identified around the hovering condition (it must be remembered that also the derivatives of the moments with respect to roll rate and pitch rate were set to zero, see Chapter 4.3 for more details).

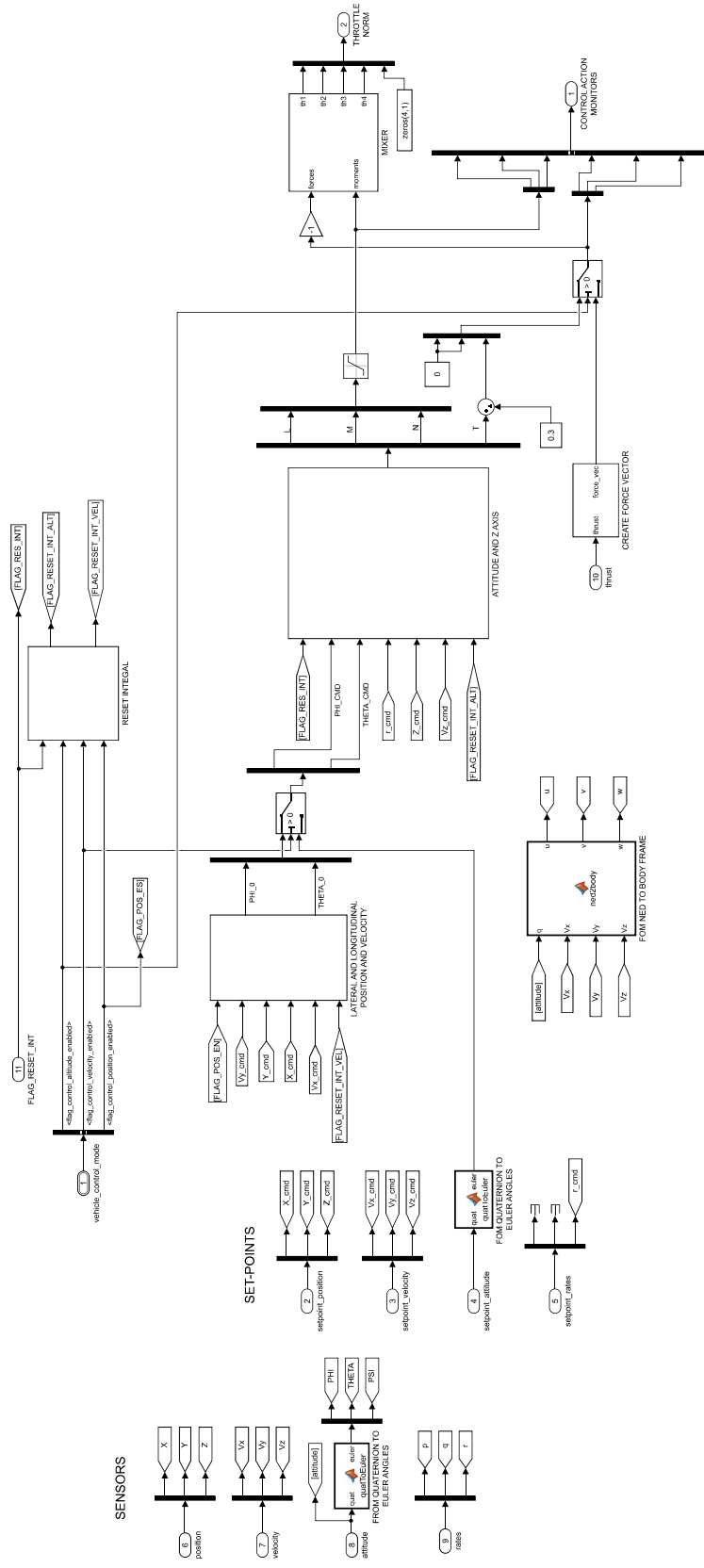


Figure 7.1: Simulink flight controller architecture example.

To compare the two different flight controllers, speed command doublets were chosen as inputs, in order to have a comparison of both the outer velocity loop and the inner attitude loop. In this case, off-board automated inputs were preferred rather than manual radio controls in order to have precise, fast, repeatable and therefore comparable commands. Figure 7.2 shows the response of the ANT-R to a lateral speed command doublet of 1 m/s amplitude with a semi-period of 2 s .

As seen in Chapter 6.3 studying the simulations, the output velocity of the EMF flight controller features a slight overshoot with respect to the desired output while DI control system follows correctly the set-point. Very similar angular rates and angles were reached during the maneuvers.

In order to study these small differences of the two flight controllers, the control input actuator is studied once again in details. This signal is the sum of the model inversion and the error compensation terms of both the inner and the outer loops (see Chapter 5 for a detailed description of the control laws). A comparison of these quantities is shown in Figure 7.4 and Figure 7.5 for the outer velocity loop, where data are expressed in *degrees* as inputs for the inner attitude loop and in Figure 7.6 and Figure 7.7 for the inner loop, where data are expressed in percentage of the maximum applicable moment as inputs for the mixer matrix.

Considering the outer loop as an example (Figure 7.4 and Figure 7.5), the two considered terms are defined more in depth in Figure 7.3:

- For the control action term, the output of the PI regulator a_{PI} (Figure 7.3a), which has unit of measurements m/s^2 , is divided by g and compared to the output of the LQR compensation ϕ_{LQR} (Figure 7.3b), which has unit of measurements of an angle;
- For the model inversion term, the signal a_{PI} (Figure 7.3a) divided by g is subtracted from the output of the dynamic inversion ϕ_{cmd} and compared to the EMF feed-forward term $\phi_{cmd_{ff}}$ (Figure 7.3b).

Differences in the error compensation term for both the loops are clearly due to the different control systems: PID regulators are present in the DI flight controller while LQR is used in the EMF flight controller. This could be solved by adjusting the LQR penalties in such a way that they would give results more similar to the Dynamic Inversion control law. More interesting are the model inversion terms: as can be seen in Figure 7.5 and Figure 7.7 the noise is an important factor in the results; in fact despite having used the same equations of motion in both DI and EMF model inversions, the Dynamic Inversion flight controller introduces these terms in feedback and, as seen in the previous section, these signals come from the optical motion capture system and noise disturbances are inevitable. Moreover, these quantities must be rotated from the NED to the body reference system, introducing also possible numerical errors.

Finally, concerning the z axis, the quadrotor maintained correctly the altitude with both the implemented flight controllers, with no significant differences. Results along the vertical axis are reported in Figure 7.8.

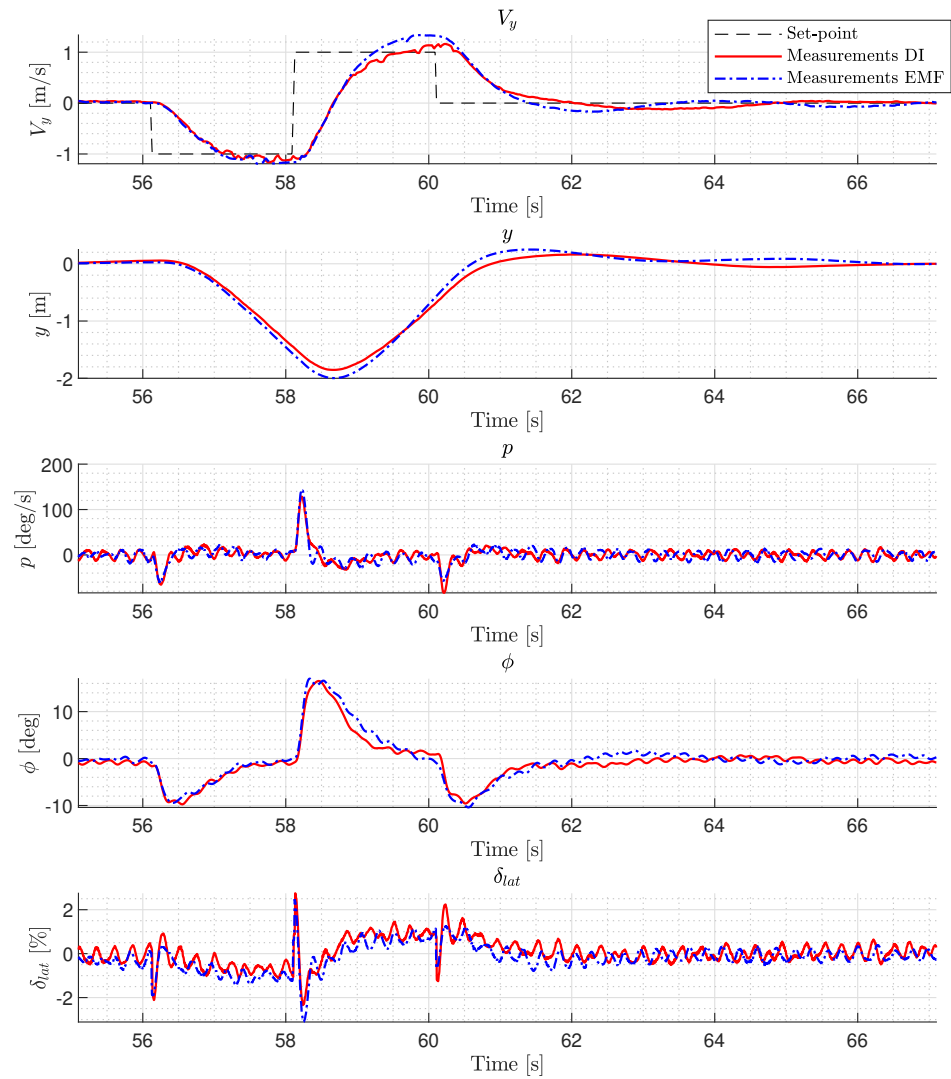


Figure 7.2: DI and EMF comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , lateral position y , roll rate p , roll angle ϕ , lateral control actuator input δ_{lat}).

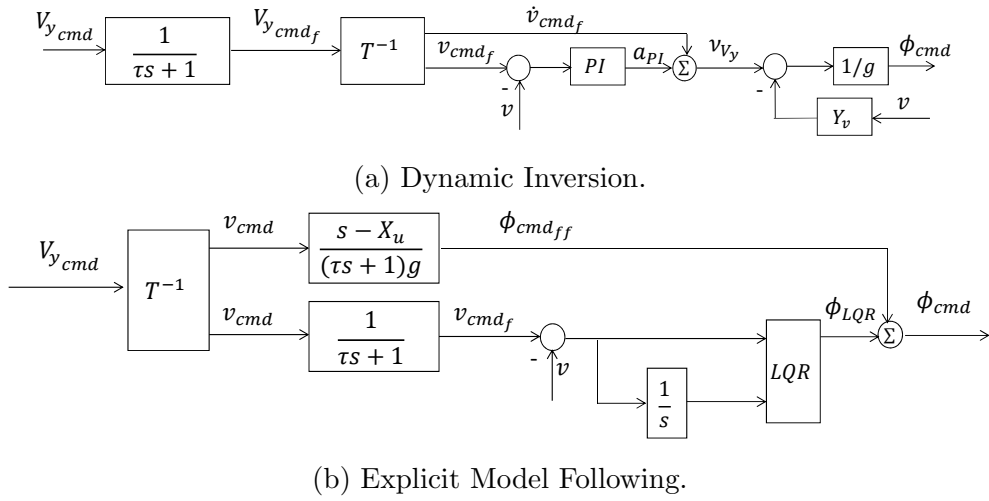


Figure 7.3: DI (a) and EMF (b) outer loop control action and model inversion terms comparison.

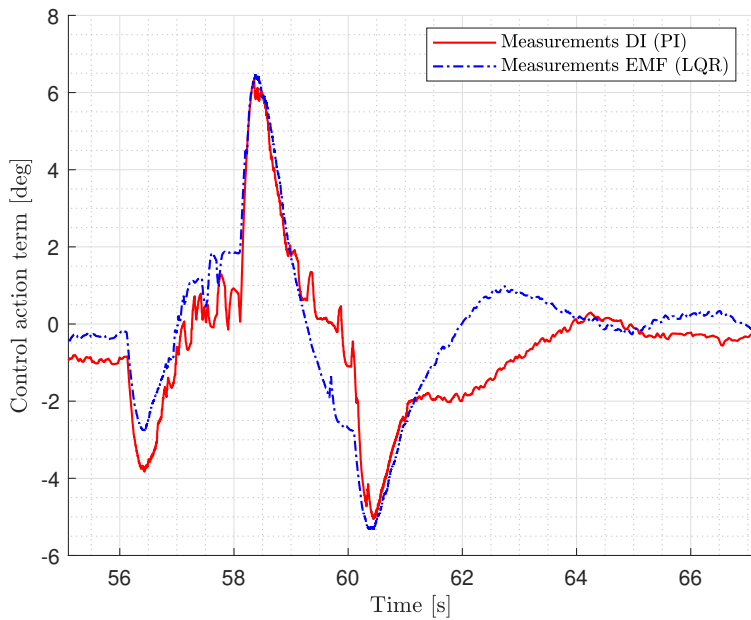


Figure 7.4: DI and EMF comparison: outer loop control action term.

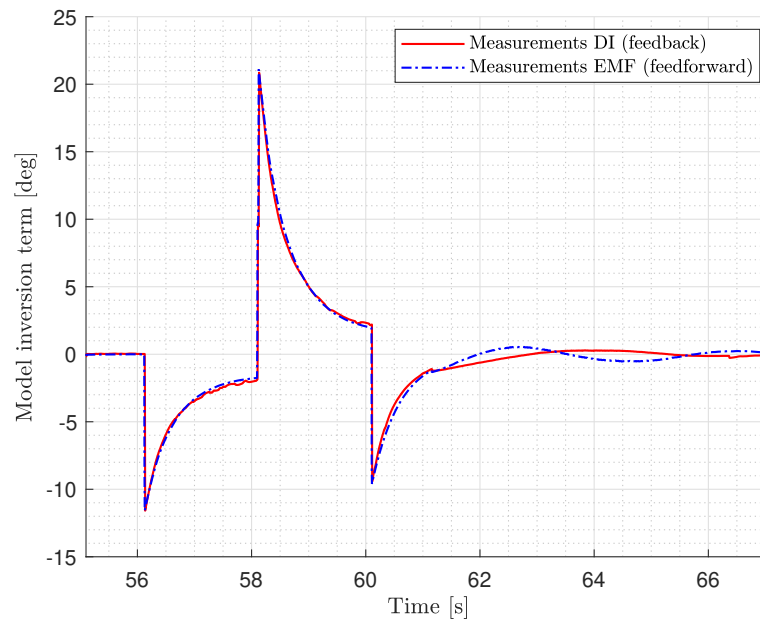


Figure 7.5: DI and EMF comparison: outer loop model inversion term.

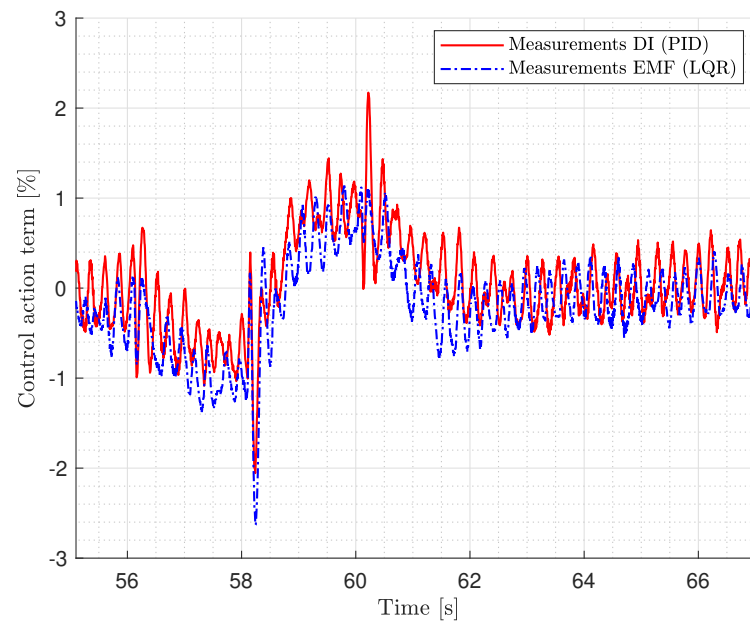


Figure 7.6: DI and EMF comparison: inner loop control action term.

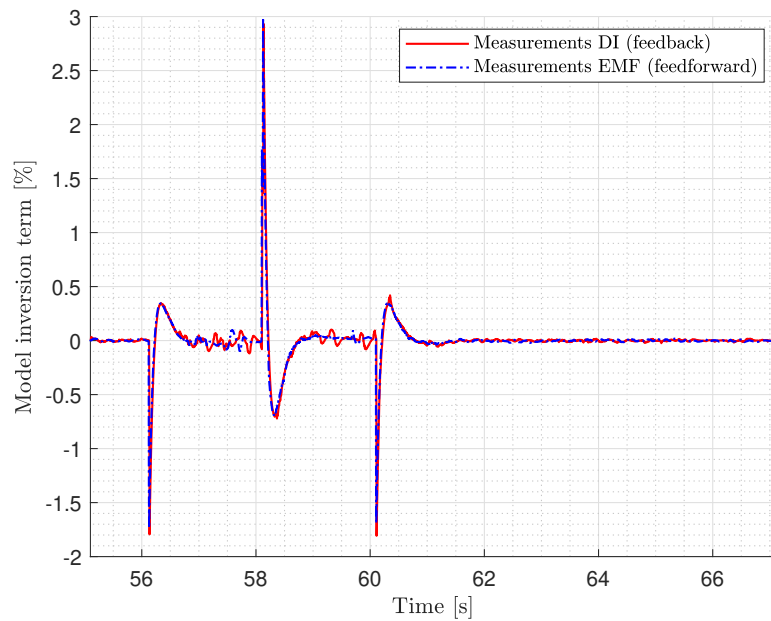


Figure 7.7: DI and EMF comparison: inner loop model inversion term.

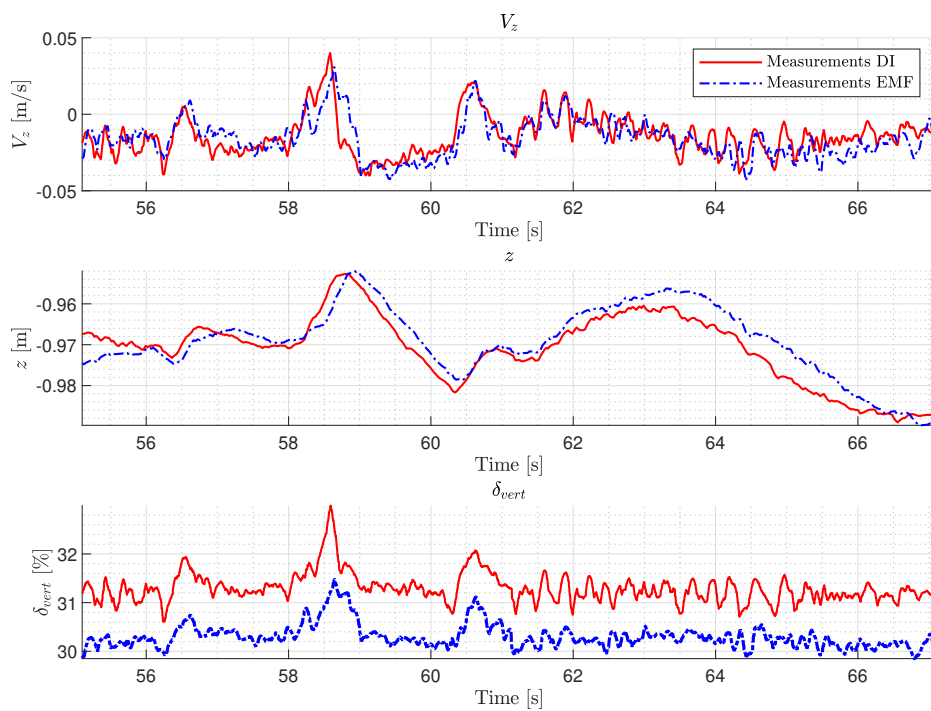


Figure 7.8: DI and EMF comparison: vertical axis (from top to bottom: vertical velocity V_z , vertical position z , vertical actuators control input δ_{vert}).

7.3 Simulations and in-flight tests comparison

In this section, the accuracy of the ANT-R quadrotor simulators studied in Chapter 6 will be verified. Data were collected commanding manually the quadcopter with the radio-controller with attitude control activated (no comparable and repeatable data obtained with off-board command inputs were required in this case, rather the focus was put on verifying the accuracy of the identified model when considering realistic manual command inputs). The angular set-point was used as input for the simulations. Dynamic Inversion control law results are shown in Figure 7.9, considering the lateral axis as an example. The considered outputs are the same used for the model identification (see Chapter 4), except for the simulated angle which was obtained integrating the angular rate. The same procedure has been followed for the Explicit Model Following controller, see Figure 7.10 for the results.

As can be seen from these tests, all the simulated outputs fit the real data with very high accuracy, confirming once again the fidelity of the identified model and therefore of the simulators.

Furthermore, thanks to these tests with the ANT-R quadrotor commanded manually with the radio-controller, it was also possible to get informations from the pilot about the flying qualities. The quadcopter in fact proved to be easy to control along each axis, confirming not only the high-performance control systems but also the correct choices of the command filter values.

7.4 Dynamic Inversion and stock ANT-R autopilot comparison

As seen in Chapter 5.2, the stock quadrotor used a cascade PID architecture as flight controller. For this reason and for the the good results obtained in the previous sections, Dynamic Inversion control law is compared with the stock ANT-R control system. The same speed doublet command input and the same outputs described in Section 7.2 are used for the comparison and results are shown in Figure 7.11. Very similar measured outputs are obtained, except for the peak of the roll rate p which is higher with the DI flight controller. The vertical axis is studied in Figure 7.12; although differences are very small, the quadrotor with the Dynamic Inversion installed onboard seems to better maintain the altitude.

7.5 Directional dynamics

As seen in Section 7.2 by describing the in-flight tests, the directional structured identified model was not reliable since the physical parameters (the stability and

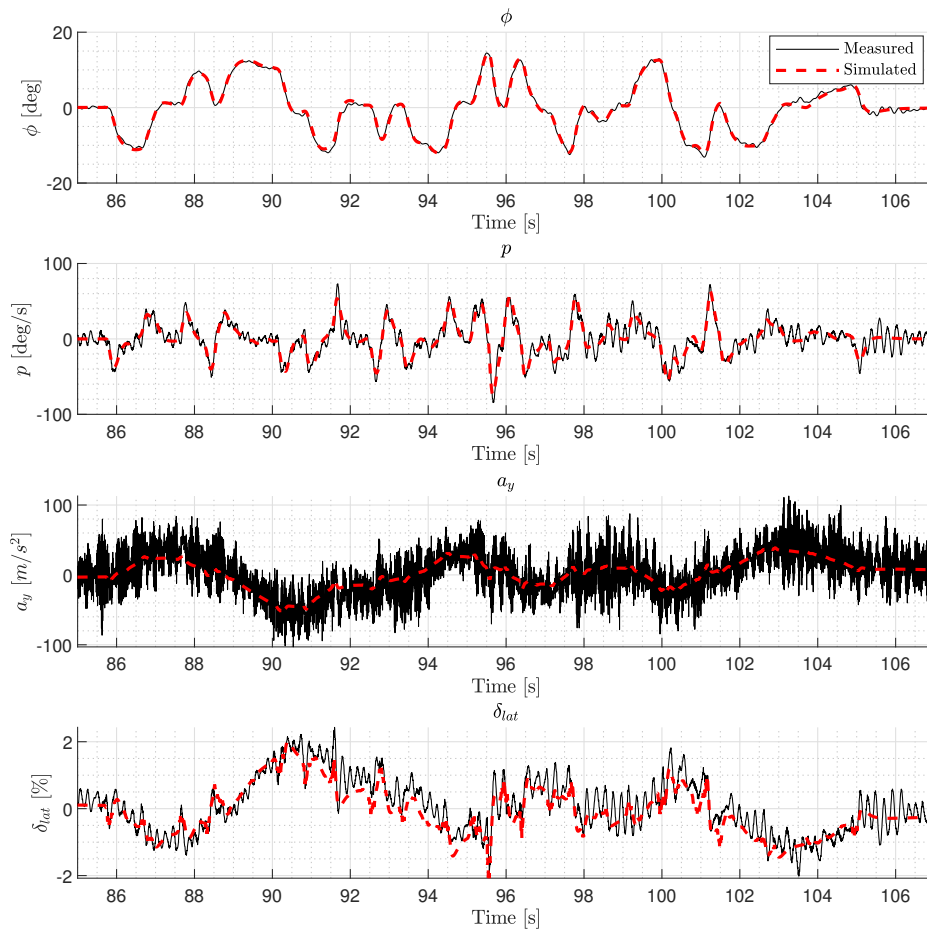


Figure 7.9: DI measured and simulated data comparison (from top to bottom: roll angle ϕ , roll rate p , lateral acceleration a_y , lateral control input actuator δ_{lat}).

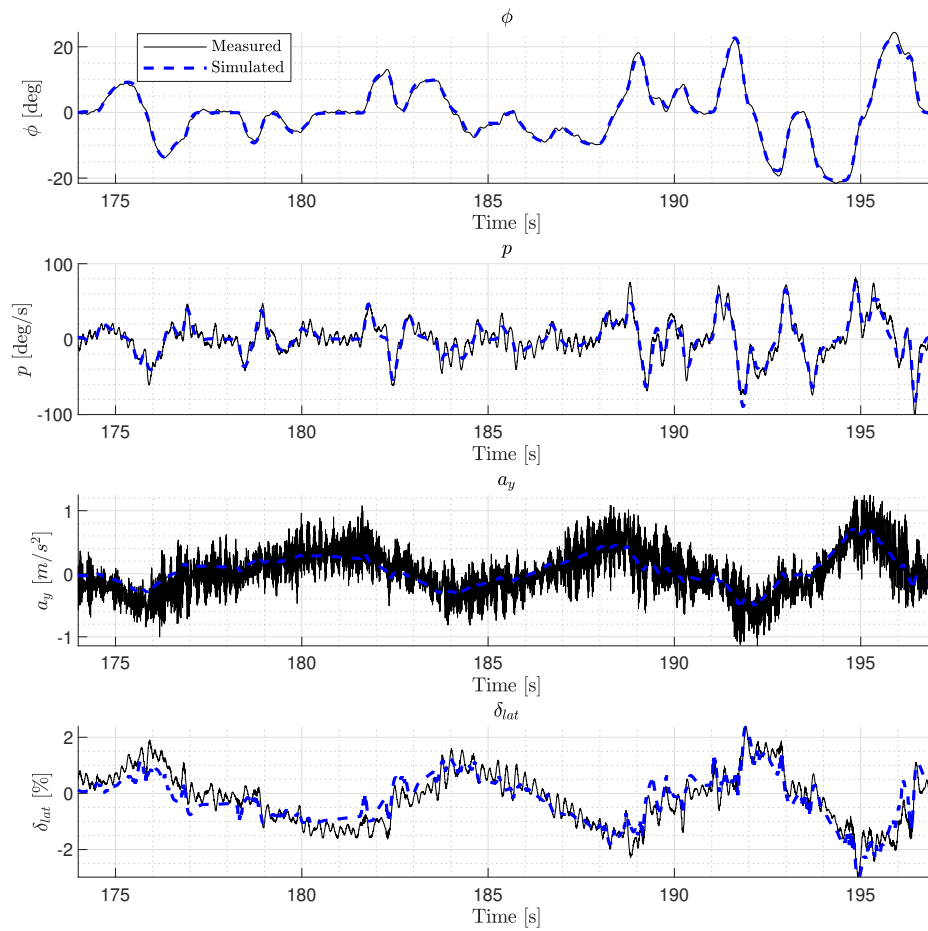


Figure 7.10: EMF measured and simulated data comparison (from top to bottom: roll angle ϕ , roll rate p , lateral acceleration a_y , lateral control input actuator δ_{lat}).

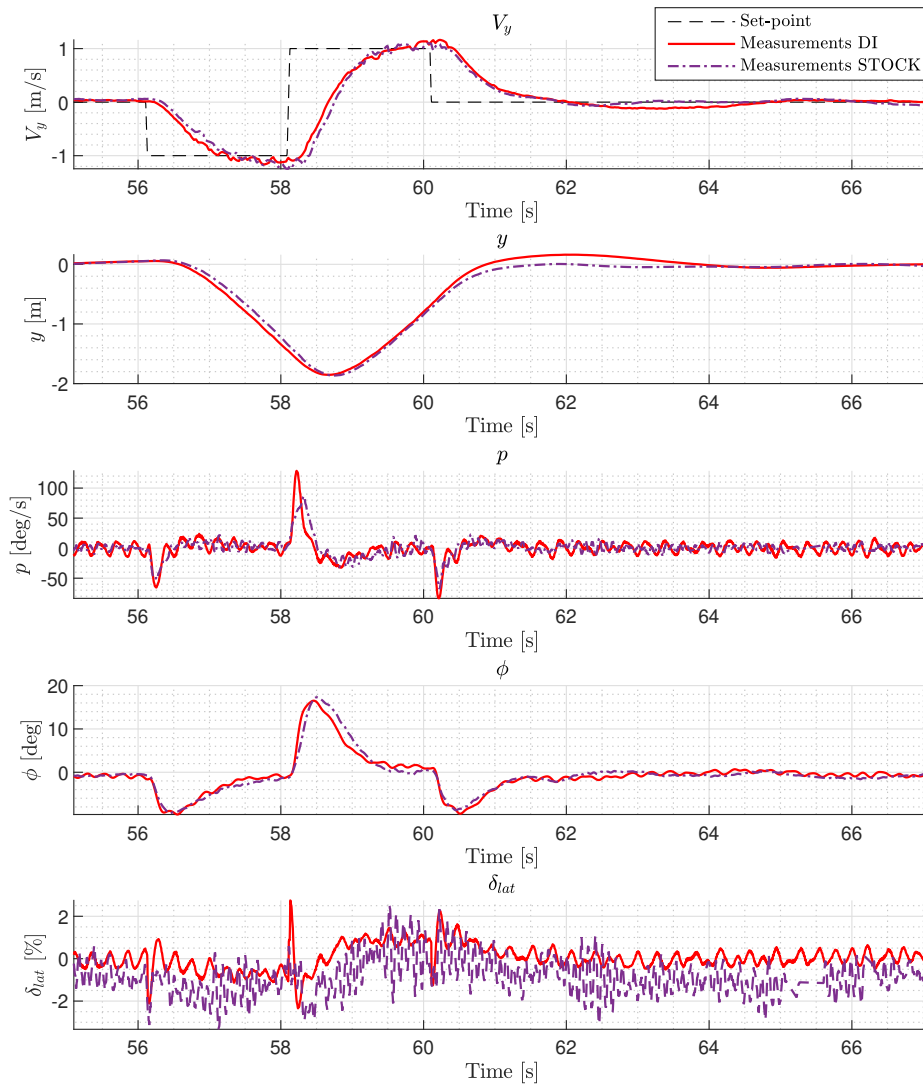


Figure 7.11: DI and stock control laws comparison: lateral speed command doublet response (from top to bottom: lateral velocity V_y , lateral position y , roll rate p , roll angle ϕ , lateral control actuator input δ_{lat}).

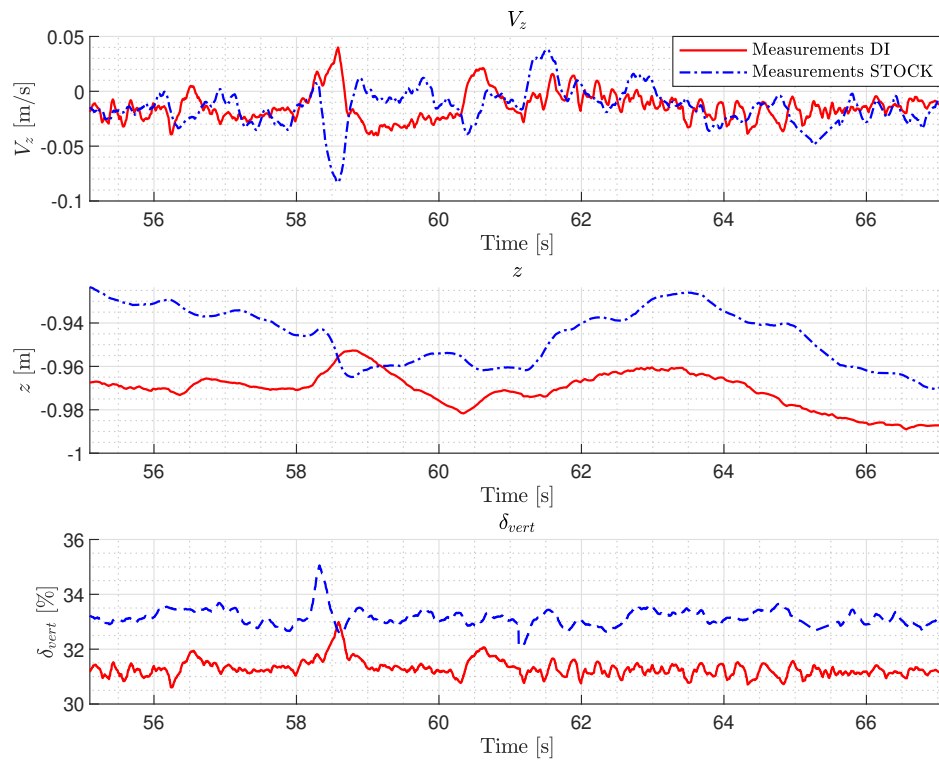


Figure 7.12: DI and stock control laws comparison: vertical axis (from top to bottom: vertical velocity V_z , vertical position z , vertical actuators control input δ_{vert}).

control derivatives N_r and N_δ) inserted in the block diagram caused disturbances around the z axis.

Considering the unstructured model obtained with subspace model identification and described in Chapter 4.2.3, a model order equal to 3 proved to be very accurate for the directional dynamics, both in term of VAF metric and by comparing the frequency response of the identified transfer function with the non-parametric frequency response function extracted from input/output measured data. If the unstructured model is used to simulate the plant model and the model inversion controllers are designed with the first structured model order, the resulting system is in fact closed-loop unstable, explaining the initial flight tests results and confirming that a first model order for the directional dynamics is not reliable.

7.6 Conclusions

The Simulink models of both Dynamic Inversion and Explicit Model Following flight controllers have been imported in the PX4 firmware of the ANT-R quadcopter, substituting the stock autopilot control system. Then, the two different control laws have been tested in flight and compared, giving very satisfactory results: the quadcopter proved to be stable and controllable by studying both off-board input responses and qualitative impressions of the pilot. The identified derivative of the yaw moment with respect to the yaw rate was set to zero, a choice in agreement with the literature. The small differences between the two control laws have been analyzed in details. Moreover, a comparison between measured data and simulated data has been performed, demonstrating not only the high accuracy of the identified model used to simulate the complete architectures, but also of the identified physical parameters of the quadcopter which are present in the model inversion terms. Finally, the Dynamic Inversion control law has been compared with the stock ANT-R autopilot, obtaining very similar results and confirming the good levels of performance obtained with the designed controller.

Conclusions

The problem of system identification of a multi-rotor UAV has been considered. A quadcopter has been used to perform structured model identification for both the angular and the linear dynamics in order to replace the stock autopilot system with model inversion-based controllers. Closed-loop data were collected in flight in laboratory conditions. Previously, only the attitude dynamics of the quadcopter has been identified with unstructured models and considering SISO problems.

First of all, an overview of the most recently discussed model identification techniques was provided, together with a description of the possible experimental tests. Then, subspace model identification was performed obtaining accurate unstructured models especially for the attitude and vertical dynamics. For the purpose of this thesis, the physical parameters of the aircraft model were estimated using the output-error model identification with satisfactory results. Additionally, to overcome the disadvantage of the SMI method (which consists in the absence of the physical meaning of the state space representation) and the limits of the output-error (in the use of closed-loop data and in the iterative nature of the approach), a new model-matching technique was investigated to obtain structured models; results were compared with the output-error ones in the time and frequency domain and in terms of the estimated stability and control derivatives. The novel approach resulted sensitive to the unstructured identified model to be matched for the lateral and longitudinal cases, and the physical parameters were not considered reliable. On the contrary, very similar results were obtained studying both the directional and the vertical dynamics. Output-error models were also compared with those obtained with SMI method. The former proved to be very accurate on the lateral and longitudinal axes, while for the vertical and the directional dynamics best performance were obtained with the unstructured models.

Due to the accuracy and to the uncertainty values of the physical parameters obtained with the output-error approach, the structured models were used to design inversion-based controllers. The complete model proved to be unstable around x and y axes and minimum phase. The Dynamic Inversion and the Explicit Model following control laws were studied and simulated to tune the gains of the Dynamic Inversion PID and the penalties of the Explicit Model Following LQR regulators. DI and EMF architectures were compared in the time domain giving

very similar results. A robustness analysis confirmed the stability and performance of the designed control systems.

Finally, the two flight controllers were implemented on the flight control unit of the quadcopter replacing the stock PID cascade architecture. The first order directional model proved to be unreliable, and the derivative of the yaw moment with respect to the yaw rate was set to zero since it introduced undesired effects in the control systems. Then, using off-board automated inputs it was possible to compare in detail the responses of DI and EMF to the same speed command doublets. Similar results were obtained in terms of response and tracking accuracy. A slight overshoot was observed in the velocity output of the EMF controller as in simulations. Additional datasets were used to validate the designed simulators. Simulated outputs fitted all the measured outputs with high accuracy, confirming once again the fidelity of the identified models. Dynamic Inversion control law was also compared to the stock flight controller considering again speed doublets as inputs, observing appreciable improvements in the angular rates.

To summarize the results: structured model identification for the angular and linear dynamics of a multi-rotor UAV was studied in order to design high-performance controllers based on model inversion, starting from closed-loop measured data. This approach can be used to replace the stock autopilot systems. Furthermore, this study confirmed the importance of system identification used to build accurate aircraft simulators.

Further developments

System identification can be improved to increase the performance of model inversion-based controllers by:

- Considering higher orders for the structured models, especially for the directional dynamics which proved to be very accurate considering an unstructured model of the third order. This could also lead to better results with the novel model-matching approach;
- Performing an experimental identification campaign exciting the multi-rotor UAV from a condition of forward flight, to study how the physical identified parameters change with respect to an hovering initial condition, especially considering the stability derivatives with respect to the angular rates which were set to zero due to their high-uncertainty values;
- Optimizing the injection sequences used for identification;
- Studying approaches to identify the moments of inertia of the aircraft to include this information in the identified model and hence improve its accuracy.

Bibliography

- [1] Airframes reference. https://dev.px4.io/v1.9.0/en/airframes/airframe_reference.html.
- [2] 3DR Pixhawk Mini autopilot. https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk_mini.html.
- [3] NanoPi computer companion. http://wiki.friendlyarm.com/wiki/index.php/File:NanoPi_NEO-AIR-1.jpg.
- [4] OptiTrack. <https://optitrack.com/>.
- [5] L. Ljung. Perspectives on System Identification. *IFAC-PapersOnLine*, 41:7172–7184, 07 2008.
- [6] M. Bergamasco and M. Lovera. Identification of linear models for the dynamics of a hovering quadrotor. *IEEE Transactions on Control System Technology*, 22:1696–1707, 09 2014.
- [7] P. Panizza, F. Riccardi, and M. Lovera. Black-box and grey-box identification of the attitude dynamics for a variable-pitch quadrotor. *IFAC-PapersOnLine*, 48, 06 2015.
- [8] F. Riccardi. *Model identification and control of variable pitch quadrotor UAVs*. PhD thesis, Politecnico di Milano, 2015.
- [9] M. Wu and M. Lovera. Time domain vs frequency domain identification for a small scale helicopter. *5th CEAS Conference on Guidance, Navigation and Control, Milano, Italy*, 2019.
- [10] M. Bergamasco and M. Lovera. State space model identification: from unstructured to structured models with an H_∞ approach. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, pages 202–207, 02 2013.
- [11] U. Saetti, S. Lakhmani, T. Berger, J. F. Horn, and C. Lagoa. Design of Dynamic Inversion and Explicit Model Following control laws for quadrotor inner and outer loops. *AHS Forum 74, At Phoenix, AZ*, 05 2018.

- [12] C. Ivler, R. Niemiec, F. Gandhi, and F.C. Sanders. Multirotor electric aerial vehicle model validation with flight data: Physics-Based and System Identification Models. *Proceedings of the 75th Vertical Flight Society Annual Forum*, 05 2019.
- [13] C.M. Ivler, E.S. Rowe, J. Martin, M.J.S. Lopez, and M. B. Tischler. System identification guidance for multirotor aircraft: dynamic scaling and test techniques. *Proceedings of the 75th Vertical Flight Society Annual Forum*, 2019.
- [14] W. Wei, M.B. Tischler, and K.B. Cohen. System identification and controller optimization of a quadrotor unmanned aerial vehicle in hover. *Journal of the American Helicopter Society*, 62(4), 10 2017.
- [15] K.K. Cheung, J.A. Wagster, M.B. Tischler, C.M. Ivler, M.G. Berrios, T. Berger, O. Juhasz, E.L. Tobias, C.L. Goerzen, P.S. Barone, F.C. Sanders, M.J.S. Lopez, and R.M. Lehman. An overview of the U.S. Army Aviation Development Directorate Quadrotor Guidance, Navigation, and Control Project. *AHS 73rd Annual Forum*, 05 2017.
- [16] G. Guglieri and C.E.D. Riboldi. *Introduction to flight dynamics*. Celid, 2016. pp. 71–80.
- [17] B.L. Stevens, F.L. Lewis, and E.N. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Wiley, third edition, 2015.
- [18] M.B. Tischler and R.K. Remple. *Aircraft and rotorcraft system identification*. Amer Inst of Aeronautics, 2012, second edition. pp. 359–371.
- [19] L. Meier. Mavlink. <https://mavlink.io/en/>.
- [20] GitHub. PX4. <https://github.com/PX4>.
- [21] Microsoft. Windows 10 pro. <https://www.microsoft.com/it-it/p/windows-10-pro/df77x4d43rkt>.
- [22] Canonical. Ubuntu 16.04. <https://ubuntu.com>.
- [23] MathWorks. MATLAB. <https://it.mathworks.com/products/matlab.html>.
- [24] J. F. Horn. Non-linear dynamic inversion control design for rotorcraft. *Aerospace*, 6:38, 03 2019.
- [25] V. Klein and E.A. Morelli. *Aircraft System Identification: Theory and Practice*. Washington, DC, USA: AIAA, 2006.

-
- [26] A. Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43:1034–1048, 06 2007.
- [27] MathWorks. Linear grey-box model estimation. <https://it.mathworks.com/help/ident/ref/greyest.html>.
- [28] P. Apkarian and D. Noll. Nonsmooth optimization for multiband frequency domain control design. *Automatica*, 43:724–731, 04 2007.
- [29] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transaction on Automatic Control*, 51, 05 2006.
- [30] MathWorks. Control system tuning. <https://it.mathworks.com/help/slcontrol/ug/sltuner.systune.html>.
- [31] P. Gahinet and P. Apkarian. Structured H_∞ synthesis in MATLAB. *Proceedings IFAC World Congress, Milan*, 18:1435–1440, 08 2011.
- [32] G. Gozzini. Uav autonomous landing on moving aerial vehicle. Master’s thesis, Politecnico di Milano, 2019.
- [33] U. Saetti and J.F. Horn. Use of harmonic decomposition models in rotorcraft flight control design with alleviation of vibratory loads. *European Rotorcraft Forum, Milan, Italy*, 09 2017.
- [34] U. Saetti. *Rotorcraft flight control design with alleviation of unsteady rotor loads*. PhD thesis, The Pennsylvania State University, 2019.
- [35] MathWorks. Simulink. <https://it.mathworks.com/products/simulink.html>.
- [36] G. Bressan. Hardware/software architecture, code generation and control for multirotor uavs. Master’s thesis, Politecnico di Milano, 2018.

