POLITECNICO
MILANO 1863

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science in Aeronautical Engineering

Data-driven multivariable attitude control design of multirotor UAV platforms

Advisor:
Prof. Marco LOVERA

Co-advisors:
Dr. Davide INVERNIZZI
Dr. Simone PANZA
Eng. Mattia GIURATO

Candidate:
Angelo ZANGARINI
Matr. n. 883107

Academic Year 2017 – 2018

A chi mi è stato vicino …

Ringraziamenti

In conclusione di questo lavoro vorrei spendere qualche parola per ringraziare tutte le persone che mi hanno sostenuto e aitutato, iniziando con coloro che mi sono stati vicini nel quotidiano. Grazie a Jacopo, Luca, Paolo, Samy, Stefano, Tania e Tomeu, per la loro amicizia e per avermi appoggiato in questo periodo turbolento. Ringrazio anche tutti gli amici e compagni di studi con i quali ho percorso questa strada piena di fatiche e sfide, ma anche molte soddisfazioni.

Un ringraziamento particolare và alla mia famiglia, che non solo mi ha permesso di intraprendere gli studi, ma mi ha spronato a impegnarmi costantemente, stando al mio fianco.

Ringrazio il professore Marco Lovera per avermi dato l'opportunità di mettermi alla prova e per la sua pazienza, chiarezza, disponibilità e competenza che mi ha permesso di affrontare serenamente il lavoro di tesi grazie ai suoi consigli.

Voglio ringraziare anche tutto il gruppo del laboratorio ASCL per avermi trasmesso la loro passione per i droni e non solo. Vorrei ringraziare Davide per la sua disponibilità, i suoi suggerimenti e per avermi supportato, soprattutto nella parte iniziale e conclusiva della stesura. Non posso che ringraziare anche Simone e Mattia per avermi affiancato nella parte sperimentale e nell'affrontare, con la loro infinita pazienza e professionalità, i problemi di tutti i giorni, sempre dietro l'angolo quando meno te l'aspetti.

Infine, ma non per importanza, un sentito ringraziamento va a Elisa, una persona speciale che mi è stata sempre vicina (purtroppo spesso a distanza). Grazie per avermi supportato, e anche sopportato, dandoci a vicenda la spinta e la motivazione per dare il nostro massimo giorno per giorno.

Grazie a tutti.

Angelo Zangarini

Milano, 16/04/2019

Abstract

Nowadays the use of Unmanned Aerial Vehicles (UAVs) is widespread for a broad range of applications. At the core of their operation, different autopilots and controllers are used. The first layer of an autopilot is the attitude controller, meaning that its performance is essential for unmanned flight operations.

Typically the controllers are synthesised by employing methodologies reliant on the knowledge of a model of the system to be controlled. Their performance, however, depends on its fidelity. For this reason, robust control techniques allow accounting for model uncertainty or under-modelling, which might hinder the performance of the real plant.

Data-driven methods represent an alternative approach. In fact, they do not rely on a model of the system, but they are applied directly to experimental data, eliminating the aforementioned issues. In this thesis, a data-driven synthesis method is proposed and tailored for the application on multirotor UAV, formulated in a multivariable environment to allow considering coupling effects which typically affect rotorcraft. The Virtual Reference Feedback Tuning (VRFT) algorithm has been extended to cascaded control architectures.

Experimental tests have been performed in flight using a baseline controller in closed-loop conditions, ensuring safe and efficient data collection tests. Testing on two different UAV platforms shows that this method delivers consistent results, with a satisfactory level of performance when compared to the decoupled axes approach of the VRFT scheme. An additional benefit is represented by the reduction of algorithm input parameters, selected by the user, simplifying the synthesis process. This result can pave the way to automatic data-based tuning since it is a non-iterative algorithm, or the application on helicopters, which feature coupling effects that generally require a multivariable approach in designing a control law.

Sommario

Oggigiorno l'impiego di Aeromobili a Pilotaggio Remoto (APR o UAV) è sempre più esteso, abbracciando una gamma molto ampia di applicazioni. Al centro del loro funzionamento esistono una serie di controllori e sistemi di autopilota la cui base è rappresentata dal controllo d'assetto, il cui corretto funzionamento permette la definizione di leggi più avanzate.

Tipicamente la sintesi dei controllori è effettuata con metodologie che si affidano alla presenza di un modello del sistema da controllare. Tecniche di controllo robusto possono essere impiegate per considerare gli effetti dell'incertezza o di un'errata scelta del modello, che possono inficiare le prestazioni del sistema reale.

I metodi data-driven rappresentano un approccio alternativo in quanto non richiedono un modello del sistema, ma sono applicati direttamente a dati sperimentali, eliminando le problematiche esposte. In questa tesi, un metodo di sintesi data-driven è proposto e adattato ad applicazioni su UAV multirotore, formulato in una struttura multivariable per considerare i possibili effetti di accoppiamento tipici dell'ala rotante. L'algoritmo Virtual Reference Feedback Tuning (VRFT) è stato esteso per essere applicato a diverse architetture di controllo.

I dati di prova sono stati acquisiti in volo in retroazione con un controllore di base, assicurando un'acquisizione efficiente e sicura. Prove su due differenti piattaforme UAV mostrano come il metodo proposto restituisca risultati consistenti, con un livello prestazionale soddisfacente comparato con l'approccio VRFT su assi disaccoppiati. Un ulteriore beneficio è rappresentato dalla riduzione del numero dei parametri dell'algoritmo selezionati dall'utente, semplificando il processo di sintesi. Questo primo risultato può rappresentare una base per metodi di taratura automatica basati sui dati, in quanto l'algoritmo non è iterativo, o l'applicazione su elicotteri, dove gli effetti di accoppiamento richiedono un approccio multivariabile nella realizzazione delle leggi di controllo.

Contents

Ringraziamenti **iii**

Abstract **v**

Sommario **vii**

1 Introduction **1**

2 Attitude control of multirotor UAV **5**
2.1 Rigid body dynamics 5
2.1.1 Reference frames 5
2.1.2 Three dimensional rotations 6
2.1.3 Rotational kinematics 8
2.1.4 Equations of angular motion 10
2.1.5 Actuator model 11
2.2 Attitude control paradigm 12
2.2.1 PID controller 12
2.2.2 Cascaded PID feedback loops 15

3 Data-driven methods **17**
3.1 Introduction 17
3.2 Model-reference control 18
3.3 Virtual Reference Feedback Tuning 20
3.3.1 Mathematical derivation 21
3.3.2 Closed-loop experiments 22
3.3.3 Multivariable extension 22
3.3.4 Cascaded controller structure 25
3.4 Experiment design 27
3.5 Reference models 28

4 Drone platforms **31**
4.1 ADAM-0 31
4.2 ANT-R 32
4.3 Controller structure 33
4.4 Common hardware 34
4.5 Software and firmware 37

5 Simulation **39**
5.1 Benchmarks 39
5.1.1 LV100 gas turbine 39
5.1.2 Numerical example results 44
5.2 ADAM-0 simulator 44

6 Experimental testing and results **49**
6.1 ADAM-0 49
6.1.1 Nominal configuration 50
6.1.2 Perturbed configuration 55
6.2 ANT-R 65
6.2.1 Parameter results comparison 67

7 Conclusions **71**

Bibliography **77**

A Kronecker Product **I**
A.1 PID output in linearised form II

List of Tables

4.1 Default controller parameters. 34

5.1 LV100 parameter results. 41
5.2 LV100: parameters statistical analysis. 42
5.3 Numerical example controller parameters. 44
5.4 ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop simulation data. 47

6.1 Experimental data reference models for the inner and outer loops. 50
6.2 ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data. 52
6.3 ADAM-0: roll performance cost functions from experimental data. 55
6.4 ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data of the perturbed configuration. 59
6.5 ADAM-0: pitch performance cost functions from experimental data of the perturbed configuration. 60
6.6 ANT-R: initial controller parameters. 65
6.7 ANT-R: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data. 67
6.8 ANT-R: roll performance cost functions from experimental data. 70

List of Figures

2.1 Body frame. 6
2.2 Quadrotor X configuration, with motor numbering and sense of rotation (CW clockwise - CCW counter-clockwise). 11
2.3 Continuous-time PID block diagram. 13
2.4 Stability region of Forward Euler's discretisation. 14
2.5 Discrete-time PI controller with derivative action on the output. 15
2.6 Block diagram of the pitch control system, with a feed-forward gain and a derivative action based on angular rate feedback. . . 16

3.1 Feedback problem. 18
3.2 Feedback problem with measurement noise v and input injection $\bar{u}$. 23
3.3 Cascaded feedback control. 25
3.4 An example of a PRBS input sequence. 27

4.1 The ADAM-0 UAV platform. 32
4.2 The ANT-R UAV platform. 32
4.3 Block diagram of the MIMO control system, with a feed-forward gain and a derivative action based on angular rate feedback. 33
4.4 Illustration of the assembly of the various components. 35
4.5 Common hardware. 36
4.6 The Fly-ART laboratory. 36
4.7 *OptiTrack* position data updates and *MATLAB* control setup using ROS nodes. 38

5.1 LV100: Step response comparison with different controllers. . . 41
5.2 LV100: Step response of 500 simulation runs (in grey) and the mean result. 43
5.3 Numerical example step response. 45
5.4 ADAM-0 *Simulink* model. 46

5.5 Simulation of a pitch angle doublet. 48

6.1 ADAM-0: closed-loop experimental dataset. 51
6.2 ADAM-0: Roll doublet experiment with SISO method parameters. 53
6.3 ADAM-0: Roll doublet experiment with MIMO method parameters. 54
6.4 ADAM-0: Roll axis performance of the nominal configuration. . 56
6.5 ADAM-0: perturbed configuration. 57
6.6 ADAM-0: closed-loop experimental dataset of the perturbed configuration. 58
6.7 PWM input during the identification test flight. 59
6.8 ADAM-0: Pitch axis performance of the perturbed configuration. 60
6.9 ADAM-0: Roll doublet experiment with SISO method parameters on the perturbed configuration. 61
6.10 ADAM-0: Roll doublet experiment with MIMO method parameters on the perturbed configuration. 62
6.11 ADAM-0: Pitch doublet experiment with SISO method parameters on the perturbed configuration. 63
6.12 ADAM-0: Pitch doublet experiment with MIMO method parameters on the perturbed configuration. 64
6.13 ANT-R: closed-loop experimental dataset used by MIMO data-driven method. 66
6.14 ANT-R: Roll doublet experiment with SISO method parameters. 68
6.15 ANT-R: Roll doublet experiment with MIMO method parameters. 69
6.16 ANT-R: Roll axis performance. 70

List of Acronyms

CbT	Correlation-based Tuning
DC	Direct Current
DCM	Director Cosine Matrix
DoF	Degree of Freedom
EIV	Extended Instrumental Variable
ESC	Electronic Speed Controller
FCU	Flight Control Unit
GPS	Global Positioning System
IFT	Iterative Feedback Tuning
IMU	Inertial Measuring Unit
I/O	Input and Output
IV	Instrumental Variable
LTI	Linear Time-Invariant
MIMO	Multiple-Input Multiple-Output
NED	North East Down
PID	Proportional Integral Derivative
PRBS	Pseudo-Random Binary Sequence
PWM	Pulse-Width Modulation
SISO	Single-Input Single-Output
SMI	Subspace Model Identification
UAV	Unmanned Aerial Vehicle
VRFT	Virtual Reference Feedback Tuning

Chapter 1

Introduction

Multirotor systems allow for simpler construction and control of rotary-wing aircraft, making their design extremely flexible in terms of configuration, size and capabilities. For this reason, the interest for unmanned aerial vehicles (UAVs) has overgrown in the recent years: their low cost has enabled their use in a wide range of applications, even creating new ones, while also making them readily available to consumers for recreational flying. A critical aspect of their design, as for any aircraft, is the attitude dynamics which require thorough analysis and introduces a control problem.

The attitude control problem is relevant for the design of any automatic controller on-board an aircraft, be it either fixed-wing or rotorcraft, since it represents the first layer of more complex control laws, starting from attitude rate control up to velocity and position controllers. Moreover, the angular dynamics are the fastest and the most uncertain, due to assumptions and different aspects that are involved.

Attitude variations in multicopter UAVs are easily obtained by suitably reducing thrust on some of the rotors, typically by reducing angular speed and increasing it on others, providing a torque around the desired axis of rotation. This behaviour can be specialised to an indefinite number of rotors, where configurations of 4 rotors (quadcopters) and 6 (hexacopters) are more widespread. The generated torque is applicable to roll and pitch dynamics, while the yaw dynamics are considered separately. By controlling the pitch and roll angular rate, the attitude itself can be changed as desired, allowing the resulting thrust to provide a lateral component which can accelerate the aircraft linearly. By closing an additional feedback loop, the linear velocity can be set, and finally, the position can be controlled with one last feedback loop.

The dynamics of most aircraft, especially when considering fixed-wing, can be decoupled under some assumptions, allowing for the simplification

of the controller synthesis, not just in terms of complexity but also for the definition of required performance and behaviour. The decoupling hypothesis typically leads to the definition of separate single-input control problems for the longitudinal, lateral and directional planes. In more complex machines, *e.g.* helicopters, attitude variations arise from periodic local changes to the angle of attack of the main rotor blades, introducing coupling effects that cannot be neglected, and these interactions must be considered for stability and other requirements, such as passenger comfort or necessary handling qualities. Multirotor design generally features symmetric builds and mass distribution, which allows for dynamics decoupling, making single axis controller synthesis viable. However, when such symmetry is not present, coupling effects arise, which can also be present in symmetric builds due to imperfect mass distributions, deformation effects, aerodynamic interactions and so on. The approach to study the couplings requires the definition of a multivariable framework, both for the system to be controlled and the controller itself. The controller synthesis can be carried out using a model-based approach, where a model of the system is obtained. This model can be obtained through model identification techniques, either via *black-box* or *grey-box* methods.

The former determine a strictly mathematical relationship between input and output data of the plant. The latter involves a physically-motivated model of the system obtained from first principles, such as rigid body dynamics and energy conservation, and its parameters are then recovered from experimental data. The main issues and limitations of model-based control synthesis are related to:

- System under-modelling, that is the presence of dynamics in the real system which are absent in the model, due to approximations, wrong assumptions or errors in the mathematical derivation. Under-modelling often occurs when the system is highly nonlinear or of a very high order, and it is modelled as a low order linear system.

- Parameter uncertainty, that is the fact that the obtained parameters are affected by uncertainty as they are the result of experiments, typically afflicted by noise, disturbances or other operating conditions which might have not been considered. This can also be from the propagation of uncertainty of some physical parameters of the system, like mass or dimensions.

- Identification errors, that is the identification of a model which can fit the realisation of the single experiment, but it is not representative of the system. For instance, if the input provided for identification is not

able to excite the dynamics of the system, it might lead to wrongly assuming a different behaviour.

- Performance, which can be affected by the quality and derivation of the system model. For instance, a model derived for a certain bandwidth could lead to unpredictable result at higher frequencies.

- Poor adaptability in the case of parameter variations, for example, components ageing or changes in operating conditions, requiring the identification of parameter-varying models, the analysis of these effects on the system or the introduction of model stitching. These solutions introduce additional complexity and other issues, such as the requirement for additional stability and performance analysis.

Data-driven methods aim to eliminate the requirement to define a system model, and thus overcome these limitations, by casting the controller tuning problem into a parameter optimisation one. The efficiency and performance of these methods can be comparable to model-based controllers [1], especially for systems that feature high parameter uncertainty or changes in operating conditions or configurations, as it is possible to retune the controllers easily.

Thesis structure

The scope of this thesis is to implement and validate a multivariable data-driven method for fast retuning of the attitude controllers for multirotor aircraft. The work has been structured as follows:

- **Chapter 2 - Attitude control of multirotor UAV**: an overview of the attitude dynamics of an aircraft is presented, introducing the required formalisms and deriving the equations of motion from first principles. Furthermore, the actuator modelling is introduced for a quadrotor UAV, without loss of generality. PID controllers are also briefly covered, introducing the discrete time realisation and the main issues concerning their implementation.

- **Chapter 3 - Data-driven methods**: data-driven methods will be introduced, illustrating the state-of-the-art of this approach and the main limitations of model-based controller synthesis that these methods can overcome. The model reference technique is illustrated, and the structured control problem is explained, that is a fixed-structure

controller opposed to an unconstrained optimisation problem. Finally, the method is extended to cover closed-loop experiments and alternative controller architectures.

- **Chapter 4 - Drone platforms**: the multirotor platforms used for testing and evaluation are presented, discussing their main construction features. A detailed discussion on the existing controller structure is proposed, along with the main modifications introduced as part of the scope of this thesis.

- **Chapter 5 - Simulation**: the algorithm results are validated and compared to other controller synthesis methodologies. The main benchmarks used for validation of MIMO systems will be covered, as well as results from a nonlinear simulator of the ADAM-0 platform.

- **Chapter 6 - Experimental testing and results**: the experimental testing approach for the validation of the proposed algorithm is described. The two UAV platforms mentioned above, ADAM-0 and ANT-R, will be employed in a series of experiment to test the tracking performance of the synthesised regulator using the SISO and the extended MIMO version of the VRFT algorithm.

- **Chapter 7 - Conclusions**: conclusions are drawn in light of simulation and experimental results, introducing ideas for future developments.

Chapter 2

Attitude control of multirotor UAV

The attitude controller aims at controlling the flight dynamics of the aircraft, allowing to stabilise unstable modes, enhancing stability properties and achieving predetermined levels of performance. In this chapter, an overview of the attitude dynamics of an aircraft is presented, introducing the required formalisms and deriving the equations of motion from first principles. Furthermore, the actuator modelling is proposed for a quadrotor UAV, without loss of generality. PID controllers are also briefly covered, introducing the discrete time realisation and the main issues concerning their implementation.

2.1 Rigid body dynamics

2.1.1 Reference frames

The description of the motion of a rigid body in space is bound to the choice of reference frames, from which rotations and displacement can be defined. In the case of aircraft, many reference frames have been defined in the literature to cope with different applications, as some allow to simplify the notation and to make the study of the attitude dynamics more intuitive.

Local NED Frame The local North-East-Down frame $\mathscr{F}_N = \{O_B, N, E, D\}$ is widely used in aeronautics. Its origin O_B coincides with the aircraft centre of mass and, as the name suggests, features axes that are orientated such that the N axis points to the geographic North, the D axis pointing downwards towards the centre of the Earth, and the E axis such that it completes the right-handed frame set.

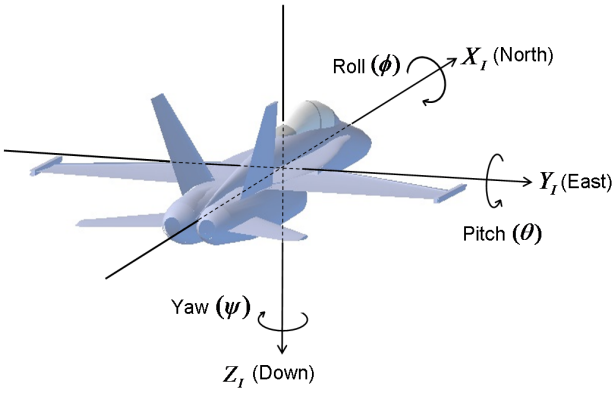


Figure 2.1: Body frame.

Earth Fixed Frame The Earth-Fixed Frame $\mathscr{F}_E = \{O_E, N, E, D\}$, not to be confused with Earth-Fixed Earth-Centred frame (ECEF), is a frame whose origin O_E is placed in a fixed point on the surface of Earth, and its axes are orientated as defined in a NED system. This frame can be assumed to be an inertial reference frame. It is noted that if small displacements from the Earth frame are considered, the N and D axes of $\mathscr{F}_E$ and $\mathscr{F}_N$ can be assumed aligned, so the local tangent plane of the Earth surface can be considered, neglecting the effects of the Earth curvature. The position of the aircraft from the inertial frame under this assumption can be easily determined as the axes are aligned.

Body Frame The Body frame $\mathscr{F}_B = \{O_N, X_B, Y_B, Z_B\}$ can be defined in several ways. The most common way is to define origin O_B coincident with the aircraft centre of mass and the X_B axis lying on the vertical plane of symmetry of the aircraft, pointing towards the front of the aircraft. The Y_B axis is normal to the vertical plane of symmetry, pointing towards the right wing, and Z_B is such that it completes the set of axes, pointing downwards. A different definition, for instance, could consider the principal axes of inertia.

2.1.2 Three dimensional rotations

Another aspect related to rigid body kinematics is how rotations can be described, as they can be referred to different frames. Rotations can be expressed by means of rotation matrix R such that it belongs to the *3D rotation*

group, also called *Special Orthogonal* $SO(3)$.

$$SO(3) \triangleq \left\{ R \in \mathbb{R}^{3\times 3}, RR^T = I, \det(R) = 1 \right\}. \tag{2.1}$$

A rotation matrix can express different behaviours, in particular:

- it can express the orientation of a reference frame with respect to another one;
- it can transform coordinates amongst coordinate frames;
- it can express the rotation of a vector around an axis.

The second property in Equation (2.1) indicates that rotation matrices must be orthogonal, while the third property indicates that the magnitude of the rotated object is left unchanged, as expected from a rotation. The fact that rotations matrices are orthonormal grants them some interesting properties which can be exploited. Orthonormal matrices can be easily inverted, as their inverse can be shown to be its transpose, such that $R^{-1} = R^T$. This property leads to a simple definition of the inverse rotation. For instance, by indicating the rotation matrix from frame 1 to frame 2 as R_{21}, the transformation from frame 2 to frame 1, R_{12}, is its transpose $R_{12} = R_{21}^T$.

Euler parametrisation

One additional step can be defining a complex rotation as several consecutive rotations: any given rotation can be decomposed in three consecutive rotations around the axes of the frame as it is being rotated. This formulation has the advantage of leading to a simpler definition of the rotation matrix, as each rotation step can be easily described with a Director Cosine Matrix (DCM). An example of the DCM for the rotations around each one of the axes of the frame follows:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{2.2}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{2.3}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.4}$$

where α, β, γ are the angles by which the frame is rotated around the specified axis, defined positive for counter-clockwise rotations as in a right-handed frame. For example, if a consecutive rotation consisting of a first rotation of α around the X axis and a second rotation of β around the Y axis of the newly rotated frame is considered, then the complete rotation matrix is $R = R_y(\beta)R_x(\alpha)$. It is noted that since each one of the rotation matrices must comply with the definition in (2.1), the concatenated rotation matrix is also compliant. The resulting matrix, however, is dependent on the order of the rotations as the matrix product does not commute, meaning that different orders lead to different rotation matrices.

The most common minimal representation of a rotation used in flight dynamics is the *Euler* parametrisation, which defines the rotation angles. This convention is generally used to define the rotation matrix from the Earth or local NED frame to the body frame and vice-versa. It is standard practice in aeronautics to use the *3-2-1* right-handed rotation [2], that is a first rotation around the Z_b axis, followed by Y_b and finally the X_b axis, defining the *attitude matrix*:

$$R_{BE}(\phi,\theta,\psi) = R_x(\phi)R_y(\theta)R_z(\psi), \tag{2.5}$$

where ϕ is the *pitch* angle, θ is the *bank* angle and ψ is the *yaw* angle. The subscript BE indicates the rotation from the Earth frame $\mathscr{F}_E$ to the body frame $\mathscr{F}_B$. For this purpose, the rotation from the Earth frame and the local NED frame $\mathscr{F}_N$ coincide. The attitude matrix terms can be expanded:

$$R_{BE}(\phi,\theta,\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix}, \tag{2.6}$$

where $c_x = \cos(x)$ and $s_x = \sin(x)$. The rotation matrix from the body frame to the Earth frame can be obtained by transposition $R_{EB} = R_{BE}^T$.

2.1.3 Rotational kinematics

The Euler parametrisation can be used to describe the rotational kinematics of the rigid-body. It is common practice to describe the attitude angles in vector notation, the Euler angles vector:

$$\Phi = \begin{Bmatrix} \phi \\ \theta \\ \psi \end{Bmatrix}. \tag{2.7}$$

Differentiation of (2.7) leads to the definition of the Euler angular rates:

$$\dot{\Phi} = \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}. \tag{2.8}$$

In kinematics, the body angular velocity vector is:

$$\omega_b = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}. \tag{2.9}$$

The relationship between Euler rates and body rates is of practical interest, as body rates can be measured with sensors installed on-board in a much easier way. The relation can be obtained by analysing consequential rotations of elemental vectors:

$$\omega_{\dot{\phi}} = \begin{Bmatrix} \dot{\phi} \\ 0 \\ 0 \end{Bmatrix}, \quad \omega_{\dot{\theta}} = \begin{Bmatrix} 0 \\ \dot{\theta} \\ 0 \end{Bmatrix}, \quad \omega_{\dot{\psi}} = \begin{Bmatrix} 0 \\ 0 \\ \dot{\psi} \end{Bmatrix}. \tag{2.10}$$

Following the 3-2-1 rotation order, we can define the body angular rate from Euler rates by summing the rotated components one at a time:

$$\omega_3 = \omega_{\dot{\psi}} \tag{2.11}$$

$$\omega_2 = \omega_{\dot{\theta}} + R_y(\theta)\omega_3 = \omega_{\dot{\theta}} + R_y(\theta)\omega_{\dot{\psi}} \tag{2.12}$$

$$\omega_1 = \omega_b = \omega_{\dot{\phi}} + R_x(\phi)\omega_2 = \omega_{\dot{\phi}} + R_x(\phi)\left[\omega_{\dot{\theta}} + R_y(\theta)\,\omega_{\dot{\psi}}\right]. \tag{2.13}$$

This product can be rearranged with the use of the Euler vector notation:

$$\omega_b = E(\phi,\theta)\dot{\Phi} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi s_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \dot{\Phi}. \tag{2.14}$$

It is noted that since the angular velocity is the sum of several angular velocities, rotated at each step for the sake of simplicity, it does not represent a rotation matrix. It is not guaranteed that it has the same properties, in fact, this matrix is not orthogonal, meaning that the inverse transformation is not trivial. In addition, its inverse can become singular.

$$E^{-1}(\phi,\theta) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}, \tag{2.15}$$

where $t_x = \tan(x)$. It can be shown that (2.15) is singular for $\theta = \pm 90$ deg, which is often referred to as *gimbal lock*. This parametrization can lead to problems with highly manoeuvrable vehicles, such as military and aerobatic aircraft and rockets. Other types of parametrization have been conceived to solve this issue, for example the *quaternion* parametrization [2].

2.1.4 Equations of angular motion

The multicopter motion can be described as the dynamics of a rigid body, which can be derived from Newton-Euler equations, starting from the Newton's second law:

$$J_b\dot{\omega}_b + \omega_b \times (J_b\omega_b) = M_{ext}, \tag{2.16}$$

where J represents the inertia tensor of the object and M_{ext} is the external moment. The inertia tensor is a positive definite matrix defined as:

$$J_b = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ J_{yx} & J_{yy} & -J_{yz} \\ J_{zx} & -J_{zy} & J_{zz} \end{bmatrix} \tag{2.17}$$

$$\begin{array}{ccc} J_{xx} = \int (y^2+z^2)dm & J_{yy} = \int (x^2+z^2)dm & J_{zz} = \int (x^2+y^2)dm \\ J_{xy} = \int (xy)dm & J_{xz} = \int (xz)dm & J_{yz} = \int (yz)dm \\ J_{yx} = J_{xy} & J_{zx} = J_{xz} & J_{zy} = J_{yz} \end{array} \tag{2.18}$$

If the frame is symmetric by construction, it is possible to define a change of basis which simplifies the inertia tensor components, yielding to the principal axes of inertia. For principal axes of inertia, the products of inertia J_{ij}, $i \neq j$ are equal to zero, thus the tensor becomes diagonal:

$$J_b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix}. \tag{2.19}$$

The external moments M_{ext} applied to the system about the body axes, following the right-hand convention:

$$M_{ext} = \begin{Bmatrix} \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{Bmatrix}, \tag{2.20}$$

where $\mathscr{L}$ is the *roll* moment, $\mathscr{M}$ is the *pitch* moment and $\mathscr{N}$ is the *yaw* moment.

When a diagonal inertia tensor is considered, the equations of attitude dynamics shown in Equation (2.16) can be simplified into:

$$\mathscr{L} = J_{xx}\dot{p} + (J_{zz} - J_{yy})qr \tag{2.21}$$

$$\mathscr{M} = J_{yy}\dot{q} + (J_{xx} - J_{zz})rp \tag{2.22}$$

$$\mathscr{N} = J_{zz}\dot{r} + (J_{yy} - J_{xx})pq. \tag{2.23}$$

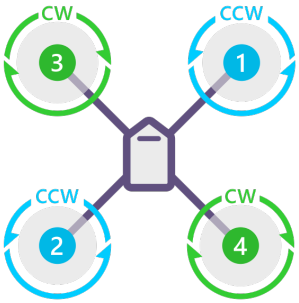


Figure 2.2: Quadrotor X configuration, with motor numbering and sense of rotation (CW clockwise - CCW counter-clockwise).

It must be noted that even for symmetric constructions, the dynamics are still coupled. This undesired effect is related to the gyroscopic term $\omega_b \times (J_b \omega_b)$ in Equation (2.16). Furthermore, the external moment present nonlinearities and couplings due to a number of dynamic and aerodynamic effects, which might depend on attitude and angular rates.

2.1.5 Actuator model

Fixed-pitch propellers are used in the considered UAV platforms due to their low cost and ease of maintenance. Variations of the rotational speed can change the thrust generated by the fixed-pitch propeller. This change, in turn, requires applying a torque to balance the effect of drag on each blade. For near-hovering conditions, the static propeller action is studied using non-dimensional coefficients through Buckingham's π theorem, obtaining a quadratic form of the propeller angular speed [3]:

$$T_i = K_T \Omega_i^2, \quad K_T = C_T \rho A R^2 \tag{2.24}$$

$$Q_i = K_Q \Omega_i^2, \quad K_Q = C_Q \rho A R^3, \tag{2.25}$$

where T_i and Q_i are respectively the thrust and torque generated by the i-th propeller, Ω_i is the rotational speed. C_T and C_Q are the dimensionless thrust and torque coefficients, which are a property of the propeller. A and R are the rotor disk area and the rotor radius, ρ is the air density.

The actuator model requires the definition of the geometry of the UAV, in terms of number of rotors, their rotation direction and distance from the centre of mass. For a symmetrical quadrotor with an X arrangement (shown in Figure 2.2), an equal arm b for each motor can be defined. The force and moments applied by the motors, expressed in the body frame, can be ob-

tained:

$$F_{prop} = -\begin{Bmatrix} 0 \\ 0 \\ K_T(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{Bmatrix} \tag{2.26}$$

$$M_{prop} = \begin{Bmatrix} K_T \frac{b}{\sqrt{2}}(-\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ K_T \frac{b}{\sqrt{2}}(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \\ K_Q \frac{b}{\sqrt{2}}(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{Bmatrix}. \tag{2.27}$$

A *mixer* matrix is often used to describe the total thrust T and moments generated by the motors as a function of the propeller speed:

$$\begin{Bmatrix} T \\ \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{Bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -K_T \frac{b}{\sqrt{2}} & K_T \frac{b}{\sqrt{2}} & K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} \\ K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} & K_T \frac{b}{\sqrt{2}} & -K_T \frac{b}{\sqrt{2}} \\ K_Q & K_Q & -K_Q & -K_Q \end{bmatrix} \begin{Bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{Bmatrix}. \tag{2.28}$$

This relation is then inverted to regulate the speed for a required thrust and moment:

$$\begin{Bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{Bmatrix} = \begin{bmatrix} \frac{1}{4K_T} & -\frac{\sqrt{2}}{4K_T b} & \frac{\sqrt{2}}{4K_T b} & \frac{1}{4K_Q} \\ \frac{1}{4K_T} & \frac{\sqrt{2}}{4K_T b} & -\frac{\sqrt{2}}{4K_T b} & \frac{1}{4K_Q} \\ \frac{1}{4K_T} & \frac{\sqrt{2}}{4K_T b} & \frac{\sqrt{2}}{4K_T b} & -\frac{1}{4K_Q} \\ \frac{1}{4K_T} & -\frac{\sqrt{2}}{4K_T b} & -\frac{\sqrt{2}}{4K_T b} & -\frac{1}{4K_Q} \end{bmatrix} \begin{Bmatrix} T \\ \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{Bmatrix}. \tag{2.29}$$

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

In the framework of this thesis, the controller output is the required thrust and moments, meaning that the mixer and electrical actuator dynamics will be annealed in the system.

2.2 Attitude control paradigm

In this section, the main aspects involved in a feedback control problem are explained. The actuator model for a fixed-pitch propeller quadrotor is presented, but it can be generalised to different configurations.

2.2.1 PID controller

The Proportional-Integral-Derivative (PID) controller is one of the most common linear industrial controllers for its simple structure and good

performance for most applications. An extensive literature, in particular regarding linear systems, and methodologies covering the tuning of these types of controllers are available, meaning that the behaviour and limitations of PID controllers are well known and documented.

Several implementations and forms exist, which typically differ in how the parameters are defined, as well as their digital, or discrete-time, counterpart. The parallel, or non-interactive, continuous-time PID controller [4] can be expressed as follows:

$$u(t) = K_i e(t) + K_i \int_{\tau=t_0}^{t} e(\tau) d\tau + K_d \frac{de(t)}{dt}, \tag{2.30}$$

where K_p, K_i, K_d are respectively the proportional, integral and derivative gain, $e(t)$ represents the error at time t which feeds the controller, and $u(t)$ represents the output of the controller, which generally is the input of the system to be controlled.

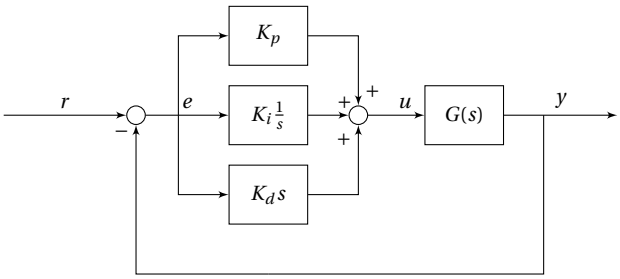


Figure 2.3: Continuous-time PID block diagram.

The Laplace operator s is often used to express this form, which leads to a simple definition of the transfer function of the controller $C(s)$:

$$U(s) = K_p E(s) + K_i \frac{1}{s} E(s) + K_d s E(s) \tag{2.31}$$

$$C(s) = \frac{U(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d s \tag{2.32}$$

The derivative action is typically filtered for implementation issues, but it carries the advantage of reducing the effects of noise, which corresponds to the high frequency content of the error signal.

Finally, the digital realisation of the PID controller is obtained by choosing a suitable discretisation method. Online applications, such as feedback controllers, need fast calculations; hence the Forward Euler discretisation

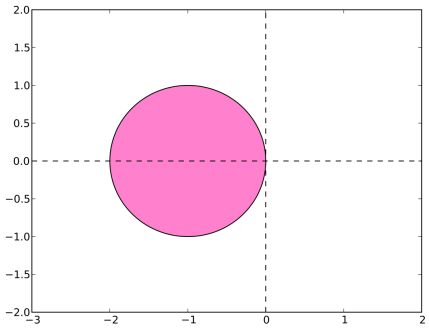


Figure 2.4: Stability region of Forward Euler's discretisation. The y-axis represents the imaginary part of λT_s, while the x-axis represents the real part.

can be used:

$$z = e^{sT_s} \simeq 1 + sT_s \tag{2.33}$$

$$C(z) = K_p + K_i \frac{T_s}{z-1} + K_d \frac{z-1}{T_s}. \tag{2.34}$$

Stability issues arise using forward differences, as stable continuous-time dynamics can be mapped into an unstable discrete-time system. This can be seen from the stability region of the method, shown in Figure 2.4, as it covers only a part of the left hand plane, whereas the stability region of a dynamical system covers all of the left hand plane. This can be an issue for stiff systems, that are systems which feature clearly separated eigenvalue time scales. These dynamics require a greater sampling frequency f_s, thus a smaller sampling time $T_s = 1/f_s$.

Amongst the main issues of the PID architecture there is the windup of integrators and sudden setpoint variations, which can be solved through additional logics:

- **Windup**: when the control action is saturated, the error might not be reduced further. The integral action increases over time, and when the error changes sign its contribution can significantly reduce the performance. This nonlinear effect can be reduced by limiting the integral action, but more advanced techniques can be used.

- **Error discontinuities**: when the setpoint is changed abruptly, *e.g.* a step, a spike in error lead to a high derivative action, which could potentially yield to excessive control inputs. This effect can be solved by either limiting the reference rate of variation or by considering only the

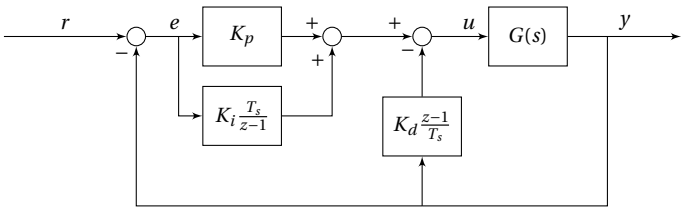


Figure 2.5: Discrete-time PI controller with derivative action on the output.

output for the derivative action. The latter solution has been implemented for the attitude controller. The resulting controller diagram is shown in Figure 2.5.

Advanced PID implementations have been studied to increase robustness and performance of this controller, like gain-scheduling algorithms or adaptive controllers, which however are outside the scope of this thesis.

2.2.2 Cascaded PID feedback loops

A cascaded PID structure is often used to increase the dynamic performance of the controlled system when long dynamics between the control input and the variable to be controlled are present. Improvements can be obtained while different measurements of the process are available for a single control input [4].

The required roll or pitch setpoint represent the attitude controller input, and the process variable is the roll and pitch angle. Measurement for both the angular rates and attitude angles are available, making it possible to define multiple nested loops. Typically the attitude rate dynamic is significantly faster than the angular dynamic, which determines the evolution over time of the attitude. Two nested PID controllers are employed for which we can define two different feedback loops, referred to as the inner (or primary) and outer (or secondary) loops. The external loop controller provides feedback for the attitude angle, defining the angular rate reference for the inner loop regulator, which in turn provide input for the angular rate dynamics.

The resulting architecture is applied to both the roll and pitch axes as is common practice in multirotor UAVs, thanks to the geometrical symmetry of quadrotors. Focusing on the pitch axis, the outer loop (measured angle ϑ, set-point ϑ^o) is a P controller, while the inner controller is a complete PID with an additional feed-forward term. More specifically, the feed-forward gain is directly calculated on the pitch angle setpoint and the derivative action of the inner loop is computed starting from the pitch rate q and not from

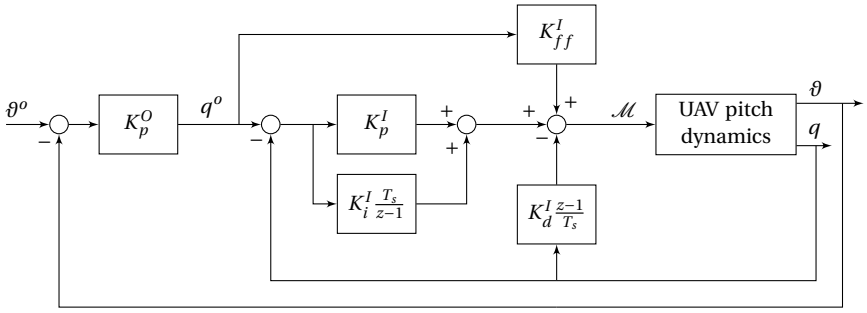


Figure 2.6: Block diagram of the pitch control system, with a feed-forward gain and a derivative action based on angular rate feedback.

the pitch angular rate error (see the block diagram in Figure 2.6, where the pitch control loop is represented), as seen in Section 2.2.1.

Until now only the pitch attitude controller was considered. The roll Degree of Freedom (DoF) is controlled with the same regulator scheme in Figure 2.6, but the involved signals are different: the user provides the roll angle setpoint $\phi^o(t)$ and the proportional outer controller generates the roll angular rate reference signal. The inner regulator, starting from this information, computes the roll pitch moment $\mathscr{L}(t)$.

This decoupled architecture for the pitch and roll axes is justified by the fact that if the body axes are principal axes of inertia, then when the quadrotor is in near-hovering conditions the roll and pitch degrees of freedom could be assumed decoupled. The aforementioned axes decoupling implies that the pitch and roll axes can be tuned independently. If, however, the system does not have decoupled attitude dynamics between the degrees of freedom, the tuning problem is coupled, and the corresponding controllers must be tuned at the same time.

Chapter 3

Data-driven methods

In this chapter, data-driven methods will be introduced, illustrating the state-of-the-art of this approach and the main limitations of model-based controller synthesis that these techniques can overcome. The model reference framework is illustrated, and the structured control problem is explained, that is a fixed-structure controller opposed to an unconstrained optimisation problem.

Then the Virtual Reference Feedback Tuning (VRFT) algorithm is rigorously derived for Single-Input Single-Output (SISO) Linear Time-Invariant (LTI) problems, and the presence of measurement noise is accounted for by employing Instrumental Variable (IV) methods. Finally, the method is generalised to cover Multiple-Input Multiple-Output (MIMO) LTI systems, and some expedients are used to be specially adapted for cascaded controller structures.

3.1 Introduction

Data-driven methods have been formulated to overcome limitations of model-based controller synthesis, avoiding the definition of a system model, either derived from first principles or experimental data. Other names by which these methods are known include *model-free* and *data-based*, highlighting the fact that only experimental data is used.

Current algorithms implementations of this concept have mostly focused around open loop excitation of single variable systems which have obtained significant results, *e.g.* in [1], [5], [6], [7]. In addition, all of the proposed solutions apply to fixed-structure controllers, leading to a result which is a set of parameters or gains for the given controller. The choice of the controller structure is beyond the scope of this work, and the analysis of PID structures

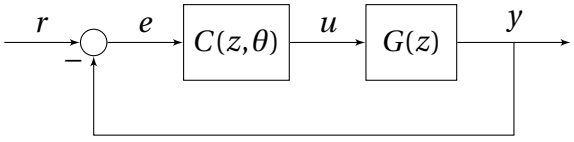


Figure 3.1: Feedback problem.

will follow.

The most relevant state-of-the-art methods are Virtual Reference Feedback Tuning (VRFT), Iterative Feedback Tuning (IFT) and Correlation-based Tuning (CbT). The former will be detailed in the following sections, for the IFT method the reader might be interested in the following articles [8], [9], while [10], [11], [12] can be consulted for the CbT method. The main differences amongst them, other than their derivation and assumptions, boil down to the nature of the optimisation algorithms, as IFT and CbT are iterative algorithms which require different datasets, while VRFT provides a one-shot solution. This feature makes VRFT extremely interesting for fast retuning of controllers, which could potentially be even carried out online.

IFT has some significant drawbacks, including the requirement for a large number of experiments, needed to compute the cost gradient and hessian. Furthermore, these algorithms do not guarantee optimal results as they might converge to local minima, making IFT even less appealing. For this reason, the IFT algorithm will not be covered. Since the aim of the thesis work is to design a technique for the fast retuning of controllers, the CbT algorithm was not considered as well, in favour of the VRFT method which is a non-iterative scheme.

3.2 Model-reference control

Model-following or model-reference control is one of the possible approaches to define the desired behaviour of a closed loop system. An example of traditional performance requirements could be a desired maximum overshoot and a given settling time.

In model reference, the desired closed-loop behaviour is expressed as a dynamical system, either in state-space form or as a matrix transfer function. It is generally easy to express the requirements on the output complementary sensitivity. The model-reference control problem can thus be formalised in the following way:

Problem 1. *Given the system plant $G(z)$, the reference model $M(z)$, a frequency weighting factor $W(z)$ and controller $C(z)$, find $\hat{C}(z)$ such that*

$$J_{MR} = \left\| W(z) \left[M(z) - \left(I + G(z)C(z) \right)^{-1} G(z)C(z) \right] \right\|_2^2 \tag{3.1}$$

$$\hat{C}(z) = \arg\min_{C} J_{MR}. \tag{3.2}$$

A possible solution of this problem can be obtained directly if the controller is not constrained, resulting in:

$$\hat{C}(z) = \left[G(z)\left(I - M(z) \right) \right]^{-1} M(z). \tag{3.3}$$

This solution might not be feasible to implement in real world applications, due to its potentially high order, presence of high frequency dynamics, and typically high control effort. Stability properties are also an issue, and finally, the relative degree of the transfer function, that is the difference between the number of poles and zeros, can be such that the system is not proper.

Model reduction of the obtained controller or system might not be a solution either, due to the impossibility to guarantee the same performance and stability properties. Thus, in most circumstances, the structure of the controller is fixed, and only its parameters can be varied, leading to a weak formulation of the optimisation problem. This formulation is also referred to as a *structured*, as opposed to the *unstructured* one presented in Problem 1. The structured problem can be formalised as follows:

Problem 2. *Given the system plant $G(z)$, the reference model $M(z)$, a frequency weighting factor $W(z)$ and a controller family $C(z,\theta)$ with vectors of parameters θ, find $\hat{\theta}$ such that*

$$J_{MR}(\theta) = \left\| W(z) \left[M(z) - \left(I + G(z)C(z,\theta) \right)^{-1} G(z)C(z,\theta) \right] \right\|_2^2 \tag{3.4}$$

$$\hat{\theta} = \arg\min_{\theta} J_{MR}(\theta). \tag{3.5}$$

There are some remarks to be made regarding the solution of this problem. Firstly, in this criterion no stability requirement is imposed, only the minimization of the discrepancy between the reference model and the output complementary sensitivity. Secondly, this generally represent a non-convex optimization problem, which means that a global minimum might not be achieved as several local minima can be present. A convex approximation of the problem can be achieved by making the following assumptions:

Assumption 1. *The sensitivity function $S(z) = I - M(z)$ is close to the closed-loop sensitivity function for $\theta = \hat{\theta}$.*

Assumption 2. *The controller family $C(z,\theta)$ can be linearly parametrized with the vectors of parameters $\theta \in \mathbb{R}^n$, such that $\mathscr{C}(\theta) = \{C(z,\theta) = \beta^T(z)\theta\}$.*

Finally, by replacing these assumptions in equation (3.4), the cost function becomes:

$$J_{MR}(\theta) = ||M(z) - (I - M(z))\beta(z)\theta)||_2^2. \tag{3.6}$$

Note that depending on the specific application, the chosen controller structure might lead to an error as low as zero. However, it is usually greater than zero for structured problems. The assumptions that have been made should be checked a posteriori. Assumption 1, for instance, states that the controlled system could be made arbitrarily close the desired model behaviour, which might not be the case for the structured case.

3.3 Virtual Reference Feedback Tuning

Virtual Reference Feedback Tuning (VRFT) is an alternative approach to model reference problems for structured controller synthesis. It was introduced in [13] for a generic nonlinear system, formally implemented and specialised in [14] for LTI discrete SISO systems. This approach was further developed in [5] for a generic MIMO LTI case, with a particular focus on dealing with data affected by noise and a thorough analysis of data filtering. The main features of this approach are:

- it only requires a single experiment, thus a single dataset of input and output;
- it is developed into a non-iterative algorithm;
- it requires limited prior knowledge of the system.

The need for limited prior knowledge is required for the definition of a reasonable reference model which must be compatible with the achievable system dynamics. VRFT is named after the signal which is exploited to define the model reference cost function to be minimised, the *virtual reference*. This signal is recreated from output data as the equivalent setpoint applied to a virtual closed-loop system that would be required to produce such output for the given reference model. This idea was first introduced with open-loop experiments in mind; however, such experiments can be dangerous in the case of open-loop unstable dynamics. It is possible to extend the approach to closed-loop experiments employing instrumental variable methods.

3.3.1 Mathematical derivation

The SISO formulation will be presented and then, after some due considerations, extended to the multivariable case. In the framework of data-driven methods, the system $G(z)$ is unknown, and a finite dataset of input u and output y measurements $d_N = \{u(k), y(k)\}_{k=1,\dots,N}$ is available. The reference signal can be reconstructed by imposing that the complementary sensitivity of the closed loop is equal to the reference model, thus defining the setpoint that would result in the same output of the experiment.

$$r_V(t) = M^{-1}(z)\,y(t) \tag{3.7}$$

$$e_V(t) = r_V(t) - y(t) = (M^{-1}(z) - I)\,y(t). \tag{3.8}$$

It is possible to define the virtual reference problem cost function J_{VR}, described in [5], as the square of the error between the applied system input and the controller output, which represents the system input. The optimal controller would lead to the same applied input:

$$J_{VR}(\theta) = \frac{1}{N}\sum_{k=1}^{N} ||u_L(k) - C(z,\theta)e_L(k)||_2^2, \tag{3.9}$$

where u_L and e_L are the filtered input u and virtual error e_V respectively.

$$e_L(k) = L(z)e_V(k) \tag{3.10}$$

$$u_L(k) = L(z)u(k). \tag{3.11}$$

By defining suitable filters and frequency weights [14] it can be showed that the cost function (3.9) is equivalent to the convex approximation of the model reference problem shown in equation (3.6). The optimal filter $L(z)$ is:

$$|L(e^{j\omega})|^2 = |1 - M(e^{j\omega})|^2 |M(e^{j\omega})|^2 |W(e^{j\omega})|^2 \frac{1}{\Phi_u^2(\omega)}, \quad \forall\omega \in [-\pi, +\pi], \tag{3.12}$$

where $\Phi_u(\omega)$ is the spectral density of $u(t)$.

Similarly, the controller can be defined by means of a linearly parametrized class, such that $\mathscr{C}(\theta) : \{C(z,\theta) = \beta^T(z)\theta, \theta \in \mathbb{R}^n\}$, obtaining:

$$J_{VR}(\theta) = \frac{1}{N}\sum_{k=1}^{N} ||u_L(k) - \beta^T(z)\theta e_L(k)||_2^2 = \frac{1}{N}\sum_{k=1}^{N} ||u_L(k) - \varphi^T(k)\theta||_2^2. \tag{3.13}$$

This minimization problem can be solved in closed form, resulting in:

$$\hat{\theta}_N = \arg\min_{\theta} J_{VR}(\theta) = \left[\sum_{k=1}^{N} \varphi^T(k)\varphi(k)\right]^{-1} \sum_{k=1}^{N} \varphi^T(k)\,u_L(k). \tag{3.14}$$

3.3.2 Closed-loop experiments

The introduction of additive noise in the realization would affect the output of the system if the experiments are carried out in closed-loop:

$$\bar{y}(t) = G(z)u(t) + v(t) \tag{3.15}$$

$$e_L(t) = (I - M(z))G(z)u(t) + (I - M(z))v(t). \tag{3.16}$$

Thus, also the system input would be affected by noise, introducing correlation which leads to a biased estimate of the parameter vector [15].

Instrumental Variable (IV) methods have been introduced to overcome this problem, allowing for the reduction of the correlation between input and output due to noise. IV methods can be implemented in two ways:

1. **Second experiment**: it is possible to achieve an asymptotically unbiased estimate of the parameters by using output data of another experiment with the same input. If noise is uncorrelated with the input that is generated for the experiment, the noise realisation can be assumed to be different, and in the case of Gaussian white noise the expected value of the correlation would be zero for $N \to \infty$;

2. **Instrumental variable**: it is possible to reduce the correlation by recreating the output data based on an estimate of the open-loop or closed-loop system, thus creating a virtual signal unaffected by noise. Such identification procedure would go against the rationale of VRFT; however, it may be argued that the identified system is not used to tune the controller but to construct an instrumental variable.

The latter can be generated by injecting a known control input $\bar{u}$ between the controller and system, as shown in Figure 3.2. This input must be uncorrelated to the measurement noise and be able to excite the system in the frequency range of interest. From this, the instrumental variable for the SISO algorithm can be created, whose derivation is thoroughly presented in [16] and [17].

3.3.3 Multivariable extension

The VRFT algorithm can be extended to the multivariable case [5], where the initial formulation is the same, but an additional step is introduced. A different instrumental variable method is employed, the Extended Instrumental Variable (EIV), which is easily implemented for multivariable problems.

The discrete MIMO LTI problem requires a redefinition of the state matrices and transfer functions, where $u \in \mathbb{R}^{n_u}$, $y, r \in \mathbb{R}^{n_y}$, $G(z) \in \mathbb{R}^{n_y \times n_u}$, $C(z) \in$

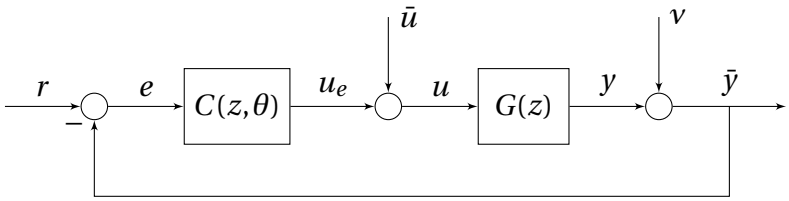


Figure 3.2: Feedback problem with measurement noise ν and input injection $\bar{u}$.

$\mathbb{R}^{n_u \times n_y}$, $T(z) \in \mathbb{R}^{n_y \times n_y}$. The model reference problem is then applied to the input complementary sensitivity $T(z)$, thus the reference model is such that $M(z) \in \mathbb{R}^{n_y \times n_y}$.

By analysing the frequency-wise counterpart of the cost function defined in (3.9) and the model reference one in Problem 1, the filters which make them equivalent can be defined as:

$$L_u(z) = M(z)\Phi_{uu}^{-1/2}(z), \quad L_e = C^{-1}(z,\theta)M(z), \quad L_y = C(z,\theta)\Phi_{uu}^{-1/2}(z). \quad (3.17)$$

These filters however require the knowledge of the controller, turning it into a nonlinear problem. By choosing the filter above as:

$$L_u(z) = L_e(z) = L(z) = M(z), \quad L_y(z) = I, \quad (3.18)$$

the cost function becomes linear in the parameter vector, and is equivalent to the approximated model reference cost function [25] shown in equation (3.6). Besides, this choice of the filters also ensures that the filtered virtual error can be computed for any proper or strictly-proper reference model.

It is noted that the filters for the MIMO extension differ from the ones derived for the SISO problem, shown in equation (3.12), due to Assumption 1 being used at the beginning of the derivation, obtaining the filter for the convex model reference problem instead of deriving the optimal filter first for the original model reference problem.

The regressor vector structure $\varphi(k)$ must be redefined as well. Another parametrisation of the controller class is introduced for the MIMO problem, such that:

$$u(k) = u(k-1) + \sum_{i=0}^{n} B_i e(k-i) \quad (3.19)$$

$$= u(k-1) + B_0 e(k) + B_1 e(k-1) + \cdots + B_n e(k-n), \quad (3.20)$$

where $B_i \in \mathbb{R}^{n_u \times n_y}, i = 1, \ldots, n$. The linear parametrization PID class can be obtained by exploiting the properties of the Kronecker product, denoted with

$\otimes$, as follows:

$$u(k) = u(k-1) + \sum_{i=0}^{n} B_i e(k-i) = u(k-1) + \varphi^T(k)\theta \tag{3.21}$$

$$\sum_{i=0}^{n} B_i e(k-i) = \left[e^T(k) \otimes I, \cdots, e^T(k-n) \otimes I\right] \operatorname{vec}\left([B_0, \cdots, B_n]\right) = \varphi^T(k)\theta, \tag{3.22}$$

where:

$$\theta = \operatorname{vec}\left([B_0, \cdots B_n]\right) \tag{3.23}$$

$$\varphi(k) = \left[e^T(k) \otimes I, \cdots, e^T(k-n) \otimes I\right]^T. \tag{3.24}$$

The vec operator is the standard vectorization operator for a matrix. The definition of the regressor φ and the parameter vector $\theta \in \mathbb{R}^{n_\theta}$, $n_\theta = n \times n_u \times n_y$ in Equation (3.22) can be further manipulated obtaining:

$$u(z) = \frac{1}{1-z^{-1}}\varphi^T(z)\theta = \frac{z}{z-1}\varphi^T(z)\theta = \varphi_F^T(z)\theta. \tag{3.25}$$

Now the extended IV is added to develop a new cost function. Unlike the SISO case, the length of the instrumental variable is not the same as the input vector. This is due to the fact that it is not built using data from another experiment or an uncorrelated signal, as presented in Section 3.3.2. In the case of EIV, the same control input is used with a window of length $\pm\, l$:

$$\zeta(k) = \begin{Bmatrix} u(k+l) \\ \vdots \\ u(k-l) \end{Bmatrix} \qquad \zeta_L(k) = \begin{Bmatrix} u_L(k+l) \\ \vdots \\ u_L(k-l) \end{Bmatrix}. \tag{3.26}$$

The previously mentioned instrumental variable can now be used to define a decorrelation cost function, as described in [15]:

$$J_D(\theta) = (r - R\theta)^T \hat{W}^{-1} (r - R\theta) \tag{3.27}$$

$$R = \frac{1}{N}\sum_{k=1}^{N} \zeta_L(k) \otimes \varphi_L(k) \tag{3.28}$$

$$r = \frac{1}{N}\sum_{k=1}^{N} \zeta_L(k) \otimes u_L(k), \tag{3.29}$$

where φ_L is the regressor defined from signals filtered with (3.18) and $\hat{W}$ is a positive semi-definite weight, optimally a consistent estimate of the residual covariance matrix $\bar{W}$:

$$\bar{W} = \mathbb{E}\left[(r - R\theta)(r - R\theta)^T\right]. \tag{3.30}$$

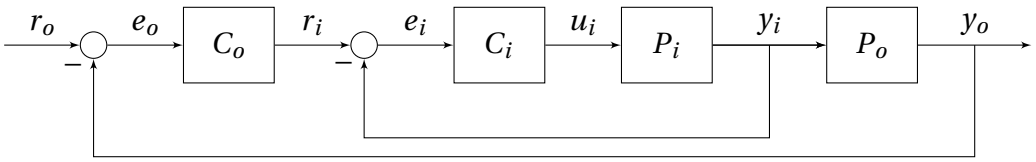


Figure 3.3: Cascaded feedback control.

The decorrelation function in the absence of noise, for large windows l, leads asymptotically to $R\theta - r = 0$. Thus, the minima of the decorrelation cost function (3.27) are equivalent to the minima of the virtual reference cost function (3.9), and are given by:

$$\hat{\theta} = \arg\min_{\theta} J_D(\theta) = \left(R^T W^{-1} R\right)^{-1} \left(R^T W^{-1} r\right). \tag{3.31}$$

The length of the window for the EIV method represents a tuning knob of the algorithm. However, an arbitrarily large number can be used, such that R is an accurate sample-based estimate of the correlation matrix of $\zeta(t)$ and $\phi(t)$.

By making suitable assumptions on the noise realisation, as detailed in [15], it is possible to study the asymptotic behaviour of the parameters. For instance, by assuming a filtered version of a white noise, which can cover a wide variety of situations, for dataset length $N \to \infty$, the IV estimate is asymptotically Gaussian distributed around the optimal parameter. This method is thus asymptotically unbiased, but it might be biased for finite N. A lower-bound for the parameter variance for $N \to \infty$ can also be determined, which is dependent both on the optimal parameter and the white noise filter. An estimate can still be obtained by substituting them with the estimated values.

3.3.4 Cascaded controller structure

The direct tuning of cascaded control systems is tackled to allow for fast tuning of nested loops. It may be necessary to control additional dynamics, typical of kinematics where velocity is derived from acceleration, and it is then integrated into position. The desired model reference for each loop is defined.

An intuitive way to approach this problem is presented in [18], where the loops are tuned separately, starting from the inner dynamics. The system is excited in this loop, and input and output data are collected. Once the inner controller is defined, the outer feedback loop data must be determined.

While the output data of the outer loop is saved during the data-collecting experiment, the input data is affected by the change of controller, so the input, which represents the setpoint of the inner loop, is reconstructed:

$$r_i(k) = e_i(k) + y_i(k) \tag{3.32}$$

$$e_i(k) = C_i^{-1}(z,\hat{\theta}_i)\,u_i(k), \tag{3.33}$$

where $\hat{\theta}_i$ is the estimate of the optimal controller for the inner loop. Note that this operation can only be made if $C_i(z)$ is of minimum-phase, that is that the system and its inverse are stable and causal. After the inner reference signal is obtained, the outer loop cost function is formulated in terms of the new dataset:

$$J_{VR}^{o}(\theta_o) = \frac{1}{N}\sum_{k=1}^{N} ||r_{i,L}(k) - C_o(z,\theta_o)e_{o,L}(k)||_2^2. \tag{3.34}$$

Finally, the complete MIMO formulation of the VRFT algorithm for the cascaded structure can be summarised by presenting a pseudo-code.

Pseudo-code. *MIMO VRFT for a cascaded controller.*

1. *Compute* $u_{Li}(k) = M_i(z)u(k)$,
2. *compute* $e_{Li}(k) = M_i(z)(M_i^{-1}(z) - I)y_i(k)$,
3. *compute* $R_i = \frac{1}{N}\sum_{k=1}^{N}\zeta_{Li}(k)\otimes\varphi_{Li}(k)$,
   where $\varphi_{Li}(k) = \frac{z}{z-1}[e_{Li}^T(k)\otimes I,\ldots,e_{Li}^T(k-n_i)\otimes I]^T$
   and $\zeta_{Li}^T(k) = [u_{Li}(k-l_i)^T,\ldots,u_{Li}(k+l_i)^T]$,
4. *compute* $r^i = \frac{1}{N}\sum_{k=1}^{N}\zeta_{Li}(k)\otimes u_{Li}(k)$,
5. *compute* $\hat{\theta}_i = ({R_i}^T R_i)^{-1}({R_i}^T r^i)$,
6. *compute* $r_i(k) = C_i^{-1}(z,\hat{\theta}_i)u(k) + y_i(k)$
   (check if minimum-phase, otherwise return to 1),
7. *compute* $u_{Lo}(k) = M_o(z)r_i(k)$,
8. *compute* $e_{Lo}(k) = M_o(z)(M_o^{-1}(z) - I)y_o(k)$,
9. *compute* $R_o = \frac{1}{N}\sum_{k=1}^{N}\zeta_{Lo}(k)\otimes\varphi_{Lo}(k)$,
   where $\varphi_{Lo}(k) = \frac{z}{z-1}[e_{Lo}^T(k)\otimes I,\ldots,e_{Lo}^T(k-n_o)\otimes I]^T$
   and $\zeta_{Lo}^T(k) = [u_{Lo}(k-l_o)^T,\ldots,u_{Lo}(k+l_o)^T]$,
10. *compute* $r^o = \frac{1}{N}\sum_{k=1}^{N}\zeta_{Lo}(k)\otimes u_{Lo}(k)$,
11. *compute* $\hat{\theta}_o = ({R_o}^T R_o)^{-1}({R_o}^T r^o)$.

The inputs of the algorithm are made up of the input and output dataset $d_N = \{u(k), y_i(k), y_o(k)\}_{k=1,\ldots,N}$, the reference models for the inner and outer

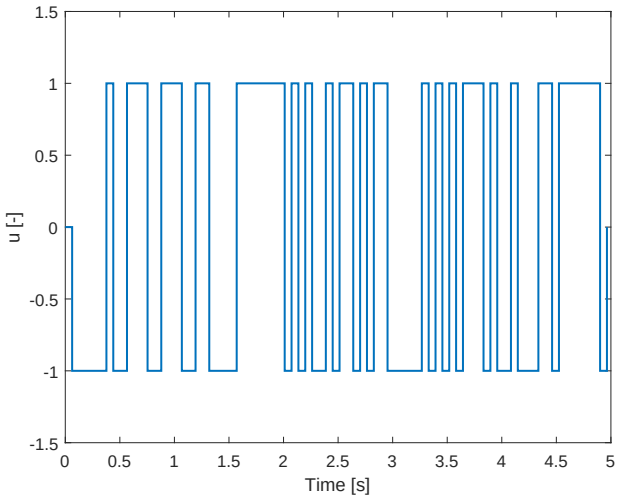


Figure 3.4: An example of a PRBS input sequence.

loops $M_i(z), M_o(z)$ and the respective instrumental variable lengths l_i, l_o. In addition, the order of the regulator has to be chosen as well for both feedback loops, n_i, n_o. The outputs of the algorithm are the parameters for the inner and outer controller $\hat{\theta}_i, \hat{\theta}_o$.

3.4 Experiment design

The excitation input for the experiment has to take into account several factors as they must provide a sufficiently informative dataset of the system dynamics, that is it should excite the system in the bandwidth of interest. This aspect is complicated by the fact that the experiments must be carried out in closed-loop, which negatively affect the outcome of the algorithm.

The experiment input is a Pseudo-Random Binary Sequence (PRBS) which is a two-state pseudo-random excitation which can be characterised in amplitude and bandwidth of excitation. The pseudo-random generation seed allows for reproducibility of the input.

There are no requirements on the input other than allowing a sufficiently informative dataset. In the experiments that have been carried out, two separated inputs have been injected, exciting only one axis at a time, in order to provide a higher signal-to-noise ratio. This, however, is not a requirement. The PRBS signal is interesting for various reasons, such as:

- ease of implementation, because as the name suggests it is a binary sequence, meaning that its possible values are either +1 or -1. This

class of inputs can also be easily automated.

- customizable, where the amplitude of the signal can be easily scaled;
- good harmonic content, which can be tailored to excite the system inside a specific bandwidth;
- reproducible, since a pseudo-random signal requires a seed which can be reused.

While the rationale of data-driven methods is to overcome the need of identifying a model of the system, such a model can be derived by using the same dataset for the given bandwidth of excitation. *Black-box* identification techniques are easily implemented for MIMO systems. Subspace Model Identification (SMI) methods are a class of *black-box* techniques which is suitable for data collected in closed-loop. An overview of SMI methods can be found in [19]. Amongst SMI methods, the Predictor-Based System Identification (PBSID) can be a viable approach, as described in [20].

More refined inputs can be conceived in order to excite specific dynamics of the system, achieving a desired power spectral density. Exciting a broad bandwidth can be a good solution for identification purposes, however, in data-driven methods, the data is not required to precisely describe the mathematical structure of the model, but to obtain a desired closed-loop behaviour. An optimal approach to input design for data-driven algorithms has been followed in [21], applied to the SISO scheme of the VRFT and CbT algorithms.

This approach was not implemented, as optimal input design was not inside the scope of the thesis, whose primary objective is the definition of a quick tuning methodology for UAV platforms, but it has been presented for completeness and provide areas for future work.

3.5 Reference models

Reference models influence heavily the result of the algorithm, in particular regarding stability, which is not guaranteed. Issues such as a non-minimum phase plant, including the presence of time delays, can become critical.

Defining reference models in this framework is not simple, as this would require knowledge of the system which data-driven methods aim to eliminate. A simple approach to model reference would require the knowledge of the system dominant dynamics, often expressed as a second-order system. The desired crossover frequency can be defined, and an approximation of the system damping ratio can be set.

As seen in the mathematical definition of model-reference control, the assumption that the optimal parameters for the given parametrised controller class lead to the desired complementary sensitivity function has to be verified. The structure of the data must be compatible with the reference model, but also with the physical limitations of the system to be controlled.

Simple reference models can be used when the required performance is conservative, such that the collected data is sufficiently informative in the objective bandwidth. However, when higher performance is required, the reference model can be enhanced by suitably introducing additional poles and zeros. The presence of time delays can also be determined to relax the requirements on the control effort, making it more compatible with the dataset.

In [22], an approach to defining an optimal reference model from data is presented and applied to the SISO VRFT method. This approach is based on a specific numerical optimisation technique called Particle Swarm Optimisation (PSO), useful when a small set of parameters have to be controlled. The number of poles and zeros is defined, and then the optimisation is made by suitably positioning them, changing their time constant. The choice of the time constants is made by defining a cost function which is augmented by terms which account for control effort, tracking error and model matching.

It might be argued that this approach is not suitable for a quick tuning methodology, but it is instead aimed at improving performance. For this reason, it was not implemented in the presented work. Once again, it was reported in order to provide an overview of potential areas to be improved to obtain higher levels of performance.

Chapter 4

Drone platforms

In this chapter, the multirotor platforms used for testing and evaluation are presented, discussing their main construction features. A detailed discussion on the existing controller structure is proposed, along with the main modifications introduced as part of the scope of this thesis.

4.1 ADAM-0

The ADAM-0 multirotor platform (see Figure 4.1), is a fixed-pitch quadrotor with the following characteristics:

- Take-Off Weight (TOW): approximately 1450 grams;
- Battery: 4S Lithium-Polymer (Li-Po) 4000 mAh;
- Flight time: 12 minutes;
- Motors: 4x Brush-less DC motors (BLDC);
- Propellers: 4x two-bladed, diameter 12";
- Frame dimensions (footprint): 500 mm (excluding rotors).

Its geometry makes it well suited for experimental research, reducing the effort required for controller testing as gyroscopic couplings are relatively small, and its behaviour is the same for both pitch and roll axes.

The medium size also makes it easy to maintain and introduce modifications. It is built with off-the-shelf components, making it affordable and reliable.

Figure 4.1: The ADAM-0 UAV platform.

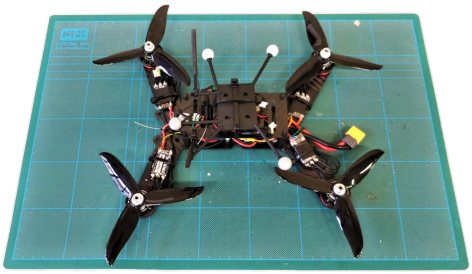


Figure 4.2: The ANT-R UAV platform.

4.2 ANT-R

The ANT-R fixed-pitch quadrotor drone is one of the latest additions to the Fly-ART laboratory. The ANT-R main features are:

- Take-Off Weight (TOW): approximately 745 grams;
- Battery: 4S Li-Po 2650 mAh;
- Maximum theoretical speed: 130 km/h;
- Flight time: 13 minutes;
- Motors: 4x racing-grade BLDC motors;
- Propellers: 4x three-bladed, diameter 5";
- Frame dimensions (footprint): 250 mm (excluding rotors).

It has been designed to provide a lightweight and high-performance machine optimised for forward flight, thus featuring a smaller cross-section.

The rotor disposition is an H configuration, which is identical to an X configuration in terms of motor numbering and sense of rotation. This arrange-

ment determines a different behaviour between the pitch and roll dynamics, due to different inertia moments.

4.3 Controller structure

Concerning the control architecture, both the ADAM-0 and ANT-R platforms adopt an attitude control scheme based on cascaded PID loops for the pitch, roll and yaw axes, running at 250 Hz. This structure has been covered in Section 2.2.1.

The MIMO implementation boils down to the same architecture, where each input is now a vector of 2 components, one for the roll axis and the other for the pitch axis. A block diagram representation is shown in Figure 4.3, where each block represents a 2×2 transfer function, which in turn is characterised by a 2×2 matrix containing the controller parameters since they share the same unitary gain transfer function. For example, the integral term transfer function can be described as:

$$C_i(z) = K_i \frac{T_s}{z-1} = \begin{bmatrix} K_{i(1,1)} & K_{i(1,2)} \\ K_{i(2,1)} & K_{i(2,2)} \end{bmatrix} \frac{T_s}{z-1}. \tag{4.1}$$

The diagonal terms affect the control action over the same axis of the measurement, for instance, pitch rate error leading to a pitch moment. The off-diagonal terms lead to control action on the other channel. The ij-th ($i \neq j$) component of the controller gives an effect on input i for an error on component j. These terms represent the decoupling action of the MIMO controller. A diagonal matrix is consequently equivalent to a set of two decoupled controllers.

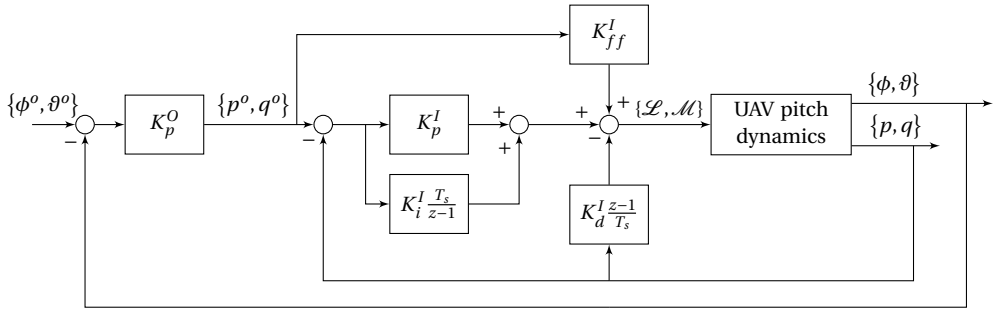


Figure 4.3: Block diagram of the MIMO control system, with a feed-forward gain and a derivative action based on angular rate feedback.

The default parameters of the controller are shown in Table 4.1, which will be used for the dataset collection experiments that follow.

Table 4.1: Default controller parameters.

$\mathbf{K_p^O}$	$\mathbf{K_{ff}^I}$	$\mathbf{K_p^I}$	$\mathbf{K_i^I}$	$\mathbf{K_d^I}$
$\begin{bmatrix} 6.5 & 0 \\ 0 & 6.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.15 & 0 \\ 0 & 0.15 \end{bmatrix}$	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$	$\begin{bmatrix} 0.003 & 0 \\ 0 & 0.003 \end{bmatrix}$

In Section 3.3.3, the regressor for the MIMO algorithm was defined for a generic PID controller fed by error, while an unconventional PID structure is employed for the considered UAV platforms. However, this type of structure can be converted into a regressor form in a straightforward step, resulting in:

$$u(k) = u(k-1) + \sum_{i=0}^{n_e} B_i e(k-i) + \sum_{j=0}^{n_y} B_j y(k-j) + \sum_{m=0}^{n_r} B_m r(k-m) \tag{4.2}$$

$$= u(k-1) + \varphi^T(k)\theta \tag{4.3}$$

A complete and formal derivation of the regressor can be found in Appendix A.1.

4.4 Common hardware

The ADAM-0 and ANT-R platforms differ mainly for their propulsion system and battery capacity; however, they share most of the off-the-shelf hardware, such as electronics and sensors. The following components are used:

- Flight Control Unit, an electronic board which runs the control laws and the Inertial Measurement Unit;
- Power Distribution Board, an integrated circuit which distributes current to the components from the battery;
- Electronic Speed Controller, a component which controls the DC motor speed;
- Telemetry module, half-duplex communication link used to send desired setpoints and recover real-time data;
- Receiver module, used for traditional radio control using an RC transmitter;
- External safety switch, used to allow the drone to be armed;
- GPS receiver, used as an additional source for position and linear velocity data;
- Magnetometer, used as an additional source for attitude data;

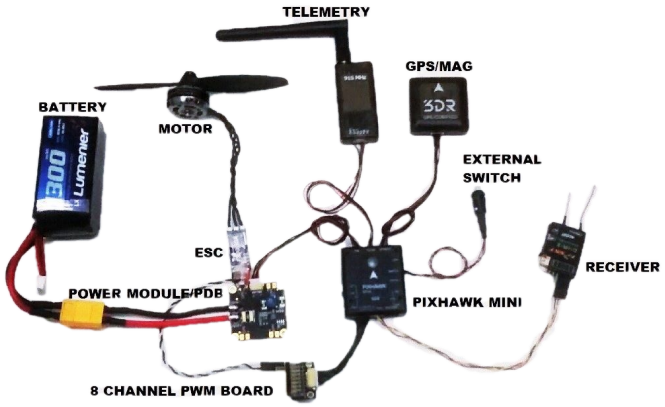


Figure 4.4: Illustration of the assembly of the various components.

- Companion board, used to publish and read data from a wireless network.

An example of the wirings and connections required for these components is shown in Figure 4.4.

Flight Control Unit Both platforms feature the use of the *Pixhawk Mini* FCU. The FCU is an integrated circuit board that host the autopilot and controllers. Its modular design allows distributing the various sensors and components, which can help to reduce intrusiveness.

The autopilot runs the attitude and position flight control laws. A Kalman Filter (KF) is present to improve the state estimation process, where information provided by an external sensor, such as a GPS receiver and magnetometer combination, is used to correct the information of the internal Inertial Measurement Unit (IMU). The IMU is made up of Micro-Electro-Mechanical Systems (MEMS) accelerometers and gyroscopes, which measure accelerations and angular rates. The attitude is reconstructed by numerical integration of the angular rates. For this reason, the Kalman filter allows for a better result which could otherwise be affected by drift due to error integration over time.

The controller setpoint is either sent via a remote controller or using a dedicated telemetry module, while the controller output is conditioned into a Pulse-Width Modulation (PWM) signal which is sent to the Electronic Speed Controllers (ESC).

Electronic Speed Controller Brush-less DC motors (BLDC), unlike Brushed DC motors, cannot be powered directly by a constant DC source.

(a) Pixhawk Mini FCU. (b) Electronic Speed Controller.

Figure 4.5: Common hardware.

They are synchronous motors, requiring the use of an inverter, which transforms DC into a switching three-phase electric signal. This component is also known as Electronic Speed Controller (ESC).

The ESC receives as input a constant DC from the Power Distribution Module (PDB) and a PWM signal from the FCU, which is then used to regulate the switching speed of the three-phase output current.

Testing facility The testing environment is the Fly-ART laboratory, which has a dedicated indoor (12 m × 6 m × 3 m) flying cage where the drones can be flown safely.

Indoor testing limits the quality of acquired GPS data, requiring a different source for the attitude, position and velocity for the long term corrections of the state estimate. An *OptiTrack* motion capture system (MOCAP) is used, with the help of reflective markers, to determine position and attitude. Once the data has been acquired, it is sent wirelessly to a companion board (*NanoPi* board) which is connected to the FCU.

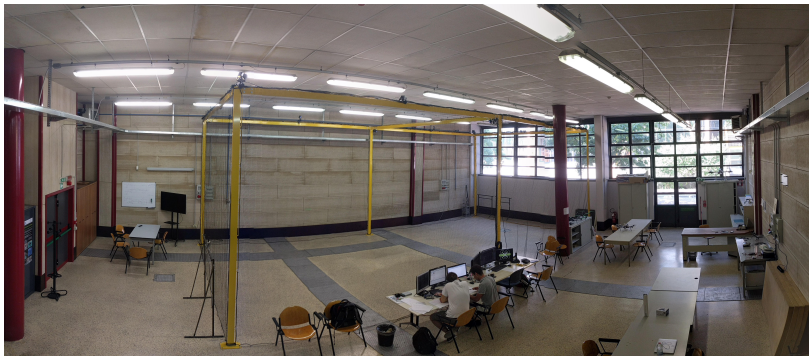


Figure 4.6: The Fly-ART laboratory.

4.5 Software and firmware

The framework of the attitude control law, shown in Section 4.3, is a *Simulink* model, which is then converted into C++ code using code auto-generation. This code is then compiled into a customised firmware. There is no need to interact with the auto-generated code, and it is only needed to specify the desired language, in this case C++, and the target hardware (ARM Cortex). The *Simulink* implementation allows for flexibility in performing modifications, as well as making it possible to validate the control law before the installation on the FCU.

The *OptiTrack* camera system requires specific software, *Motive*, used to publish the current velocity, position and attitude using ROS (Robot Operating System). ROS provides libraries, tools, hardware abstraction, device drivers, visualizers, message-passing, package management, and more to help software developers create robot applications, which include UAVs.

The companion *NanoPi* unit onboard the drone is a MAVROS node inside the ROS network. In particular, MAVROS is a ROS node that can convert between ROS topics and MAVLink messages allowing the FCU to communicate with ROS. Finally, using MAVROS, it is also possible to relay desired position setpoints or to command the injection of a control input from a ground control station. In this case, a *MATLAB* script run on the ground control station is used to publish the required topics over the ROS network.

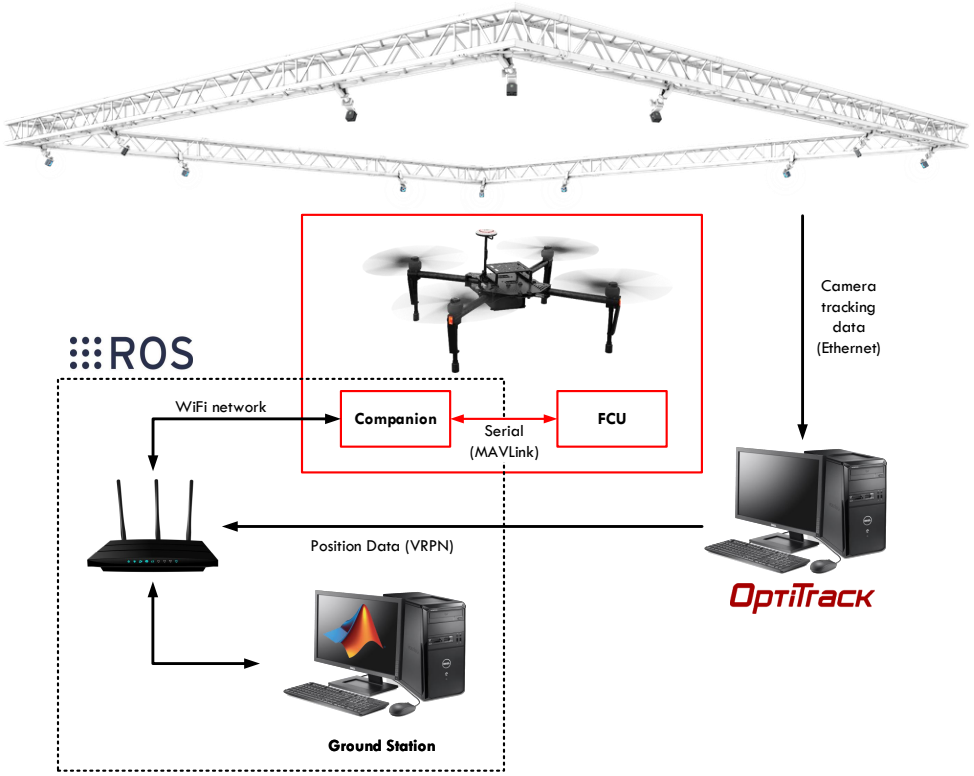


Figure 4.7: *OptiTrack* position data updates and *MATLAB* control setup using ROS nodes.

Chapter 5

Simulation

In this chapter, the algorithm results are validated and compared to other controller synthesis techniques. The main benchmarks used for validation of MIMO systems will be covered, as well as results from a nonlinear simulator of the ADAM-0 platform.

5.1 Benchmarks

5.1.1 LV100 gas turbine

One of the most common MIMO benchmarks is the LV100 Gas Turbine Engine, described in [23], which has been used for performance comparison amongst data-driven methods.

The plant is modelled as a linear two-input, two-output, five-state minimum phase system. The inputs are the fuel flow and variable area turbine nozzle. The outputs are the gas generator spool speed and temperature. The five states are, in order, the gas generator spool speed, the power output, temperature, fuel flow actuator level, and variable area turbine nozzle actuator level. A state-space continuous time model for the system is given by:

$$\dot{x} = Ax + Bu \qquad y = Cx + Du$$

$$A = \begin{bmatrix} -1.4 & -0.055 & 0 & 43.0 & 6.3 \\ 0.093 & -0.11 & 0 & 4.2 & -0.76 \\ -7.8 & -0.26 & -3.3 & 300.0 & -4.5 \\ 0 & 0 & 0 & -25.0 & 0 \\ 0 & 0 & 0 & 0 & -33.0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.1}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

A small difference is applied in [5] with respect to [23] where the temperature term contributing to the variation of the temperature, $A(3,5)$, has its order reduced by three to avoid numerical conditioning issues. Equation (5.1) represents the actual system that has been implemented.

A PRBS input is used to provide an I/O dataset for the algorithm. The two inputs have been excited separately. A total of 5000 datapoints are generated, with a frequency bandwidth that comprise 0-30 rad/s. The test has been carried out in closed loop, where the following initial controller has been employed:

$$C_0(z) = \begin{bmatrix} \frac{z-0.99}{z-1} & \frac{0.1z-0.099}{z-1} \\ \frac{-z+0.99}{z-1} & \frac{z-0.99}{z-1} \end{bmatrix}. \tag{5.2}$$

Finally, measurement noise is introduced in the test and applied in closed-loop. Noise has been modelled with a zero mean white noise with variance $0.0025I$.

For this benchmark, a PI controller fed by the tracking error is chosen. The plant dynamic is expressed in continuous time, however, for this simulation it has been discretised using Tustin's (or trapezoidal) approximation, with a sampling time of 0.1 s, corresponding to 10 Hz. A first-order reference model, with a sampling time of 0.1 s, is chosen such that:

$$M(z) = \begin{bmatrix} \frac{0.4}{z-0.6} & 0 \\ 0 & \frac{0.4}{z-0.6} \end{bmatrix}. \tag{5.3}$$

This reference model, thus, requires that the turbine spool speed is not affected by temperature variations, and vice-versa, by suitably changing the fuel flow and the turbine variable area.

Results from the multivariable extension of other data-driven methods are considered, the Iterative Feedback Tuning (IFT) [24] and Correlation-based Tuning (CbT) [12]. The time domain response to two subsequent steps is shown in Figure 5.1.

From the step response, it can be highlighted that the VRFT method is well able to decouple the dynamics, reducing substantially the effect that one input channel has over the other. The requirement, represented by the reference model, is well satisfied, as the reference model is almost indistinguishable from the VRFT result.

It is noted that the IFT is the method leading to the worst performance and it also the most demanding in terms of effort, requiring 6 iterations and results from 500 different experiments. CbT comes close in terms of model-following; however, it is still an iterative scheme, and it requires 8 different

Table 5.1: LV100 parameter results.

VRFT	$\begin{bmatrix} \frac{0.3552z-0.07165}{z-1} & \frac{0.3997z-0.2918}{z-1} \\ \frac{19.8z-19.8}{z-1} & \frac{-3.151z+2.215}{z-1} \end{bmatrix}$
CbT	$\begin{bmatrix} \frac{0.3636z-0.09866}{z-1} & \frac{0.3653z-0.2691}{z-1} \\ \frac{18.69z-18.16}{z-1} & \frac{-3.453z+2.652}{z-1} \end{bmatrix}$
IFT	$\begin{bmatrix} \frac{0.248z-0.03}{z-1} & \frac{0.38z-0.199}{z-1} \\ \frac{16.47z-15.91}{z-1} & \frac{0.063z-0.054}{z-1} \end{bmatrix}$

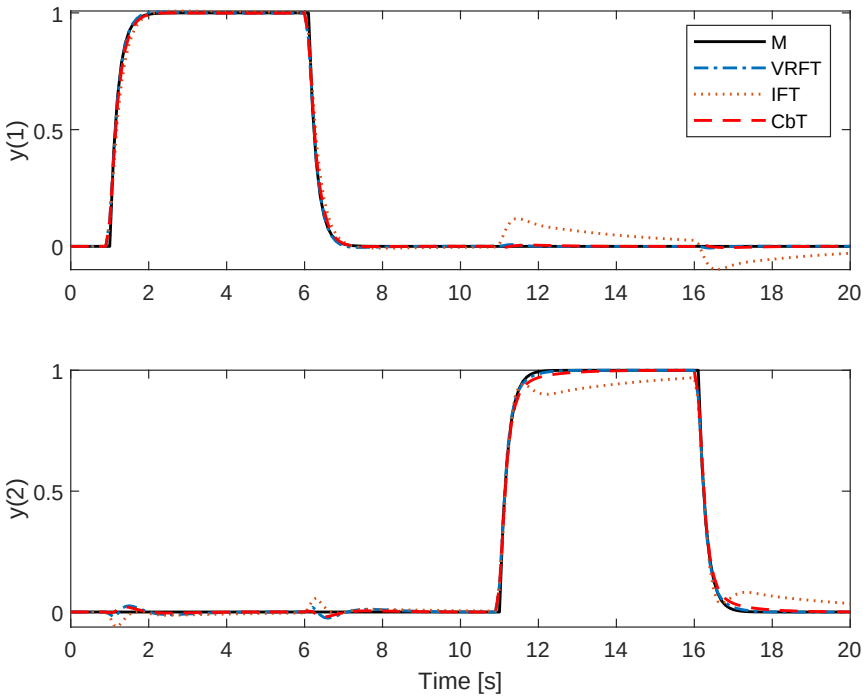


Figure 5.1: LV100: Step response comparison with different controllers.

experiments, while VRFT provides satisfactory results using a single experiment and no iterations.

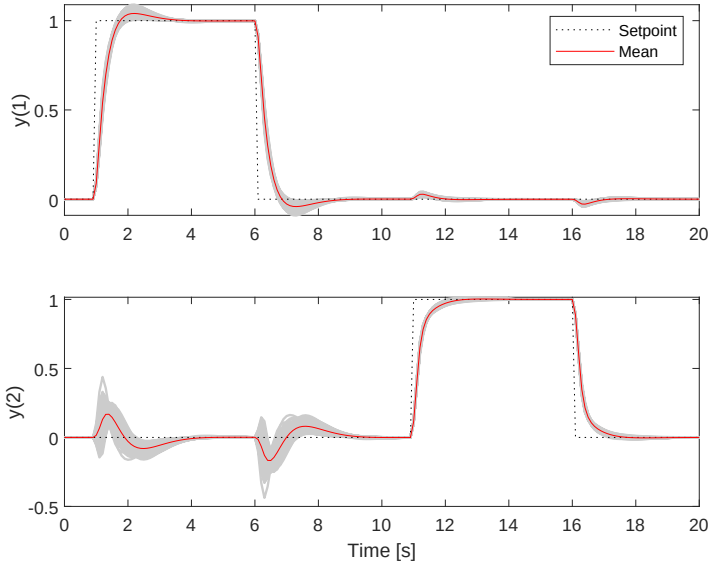
Furthermore, a statistical analysis is performed using different noise realisations to determine their effect on the parameters. The variance of the parameters will be determined under these conditions by simulating the response for different experiments. The same PRBS input will be used across the experiments on the linear plant shown in Equations (5.1). Datasets with 1000 and 10000 data-points are compared. The parameters are described in terms of mean values and standard deviation, detailed in Table 5.2. The ratio between the parameter standard deviation and value is also included.

Table 5.2: LV100: parameters statistical analysis.

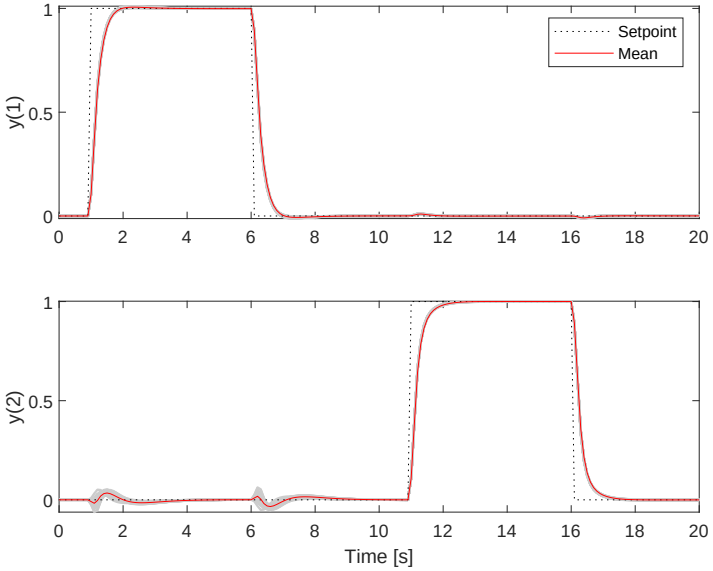
		θ_{10^3}	σ_{10^3}	$(\sigma/\theta)_{10^3}$	θ_{10^4}	σ_{10^4}	$(\sigma/\theta)_{10^4}$
$\mathbf{K_p}$	(1,1)	0.070	0.116	1.651	0.075	0.031	0.418
	(2,1)	14.037	1.172	0.083	18.698	0.925	0.049
	(1,2)	0.310	0.022	0.072	0.288	0.014	0.047
	(2,2)	1.477	0.175	0.118	2.149	0.117	0.054
$\mathbf{K_i}$	(1,1)	2.874	0.189	0.066	2.823	0.132	0.047
	(2,1)	6.449	1.006	0.156	6.238	0.443	0.071
	(1,2)	1.121	0.060	0.053	1.086	0.050	0.046
	(2,2)	8.579	0.510	0.059	9.268	0.439	0.047

The difference in the mean values can be an indicator of bias of the estimation, as this algorithm is only asymptotically unbiased for dataset length $N \to \infty$. When the larger dataset is considered, the standard deviation is decreased on all parameters, halved in most cases. The parameter $K_p(1,1)$ is particularly affected by uncertainty, which in the case of the smaller dataset is such that it leads to sign variations.

By comparing the step responses in Figure 5.2 it can be seen that for smaller dataset there is higher uncertainty, seen from the thicker grey zone. This result is coherent with the expected behaviour from the analysis of the parameter variance. Besides, it is noticeable that a dataset of 1000 data-points features more oscillations. As a result, it has inferior model-following capabilities.



(a) N = 1000.



(b) N = 10000.

Figure 5.2: LV100: Step response of 500 simulation runs (in grey) and the mean result.

5.1.2 Numerical example results

An additional MIMO benchmark is proposed in [25] and [26], where a discrete system of first order transfer functions is introduced:

$$G(z) = \begin{bmatrix} \frac{0.09516}{z-0.9048} & \frac{0.03807}{z-0.9048} \\ \frac{-0.02974}{z-0.9048} & \frac{0.04758}{z-0.9048} \end{bmatrix}, \tag{5.4}$$

with a sampling rate of 1 Hz. A first order transfer function is chosen as reference model, such that:

$$M(z) = \begin{bmatrix} \frac{0.1}{z-0.9} & 0 \\ 0 & \frac{0.1}{z-0.9} \end{bmatrix}. \tag{5.5}$$

This example is purposely built to guarantee perfect tracking with 2×2 PI controller. Performance from the MIMO algorithm and the SISO one are compared, using the same input/output dataset. The outcome of the algorithms are detailed in Table 5.3, and the step response is shown in Figure 5.3.

Table 5.3: Numerical example controller parameters.

SISO	**MIMO**
$\begin{bmatrix} \frac{1.082z-1.023}{z-1} & 0 \\ 0 & \frac{2.223z-2.057}{z-1} \end{bmatrix}$	$\begin{bmatrix} \frac{0.8406z-0.7606}{z-1} & \frac{-0.6726z+0.6086}{z-1} \\ \frac{0.5254z-0.4754}{z-1} & \frac{1.681z-1.521}{z-1} \end{bmatrix}$

Since significant coupling is present, the SISO algorithm is not able to completely satisfy tracking requirements. It is noted that even if the decoupling regulators, namely the off-diagonal terms, are not present, the effect is cancelled over time as a load disturbance. The disturbance rejection property of the controller depends on numerous factors, including the controller complementary sensitivity.

Thus, systems affected by significant coupling effects benefit from the synthesis of a full MIMO regulator. It can be seen from Figure 5.3 that the MIMO controller completely overlaps the requirement.

5.2 ADAM-0 simulator

A complete *Simulink* ADAM-0 nonlinear simulator has been used to validate the results of the algorithm. The simulator uses the equations of motion shown in Chapter 2, using some assumptions on the aerodynamic modelling

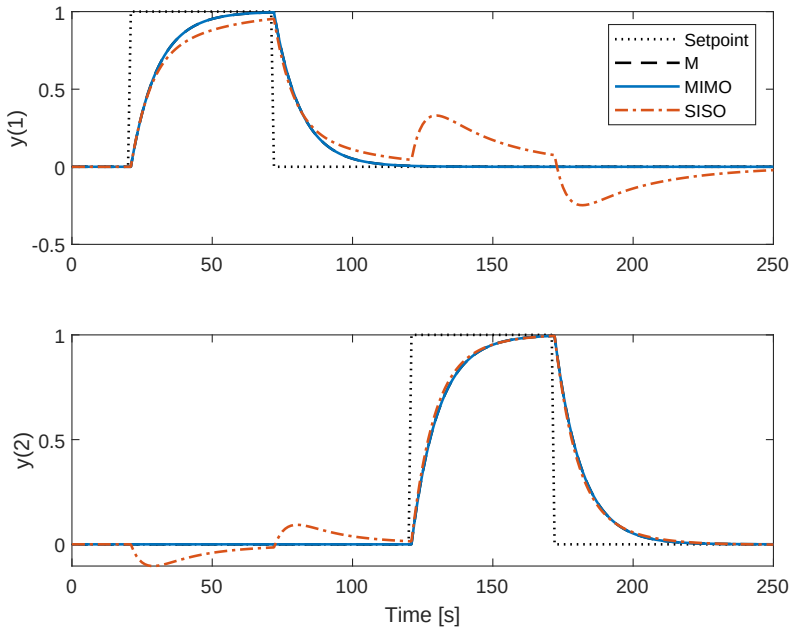


Figure 5.3: Numerical example step response.

of the damping, also know as stability derivatives. It is built specifically for preliminary testing of control laws before compiling it in the drone firmware.

The simulator can replicate the attitude dynamics under feedback control on all axes. An artificial inertial coupling has been introduced, where the off-diagonal terms represent 10% of the diagonal terms. These terms will introduce gyroscopic effects since the pitch and roll axis are no longer principal axes of inertia.

Simulation data is collected in closed-loop in order to create the input and output dataset required for the VRFT algorithm. Two PRBS excitation signals, one for the pitch moment and one for the roll moment, are applied consecutively. The input $\bar{u}(t) = \{\bar{\mathscr{L}}, \bar{\mathscr{M}}\}^T$ is injected in the system as shown in Figure 3.2. The two signals are different, but they share the same PRBS parameters (signal amplitude and min/max switching interval). In this case, for each axis, a total excitation time of 20 s has been used, with a frequency of 50 rad/s and an amplitude of 0.15. The latter is a dimensionless amplitude, referred to the maximum moment that can be applied.

As illustrated in Section 3.3.2, an initial controller $C_0(z)$ that stabilizes the system must be available, with parameters collected in Table 4.1

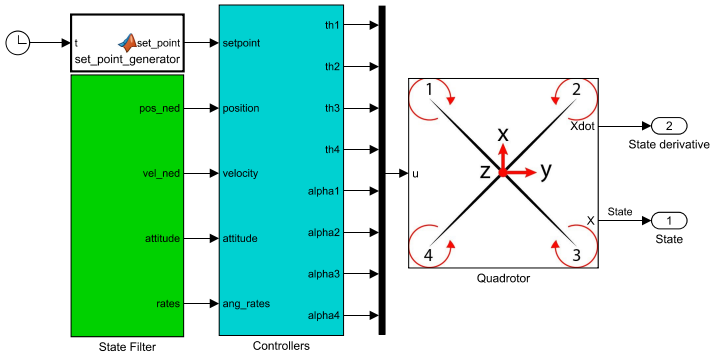


Figure 5.4: ADAM-0 *Simulink* model.

Reference models

For both the pitch and roll inner loops, the reference model is a second-order system, with a desired bandwidth and damping ratio of 20 rad/s and 0.4 respectively:

$$M_i(z) = \frac{0.003131z + 0.003065}{z^2 - 1.932z + 0.9380}. \tag{5.6}$$

In this specific case no filtering action was required for the SISO algorithm, thus the weighting function has been defined as $W_i(z) = 1$. Therefore, considering the MIMO case, the reference models are 2×2 matrices of transfer functions, with the transfer function $M_i(z)$ on the main diagonal and zeros on the secondary diagonal, requiring full decoupling.

Similarly, requirements have been set for the outer loop, once again a second-order model, and a slower response. The desired bandwidth is 10 rad/s with a damping ratio of 0.7:

$$M_o(z) = \frac{0.0007852z + 0.0007706}{z^2 - 1.9440z + 0.9455}. \tag{5.7}$$

Controller parameters comparison

Results from the SISO algorithm applied to both pitch and roll data are then compared to the controller obtained from the MIMO formulation shown in Section 3.3.3 for the given set of reference models for the inner and outer dynamics. Noise has been introduced in the system, modelled as white noise with standard deviation obtained from hovering endurance tests to account for the uncertainty of the state estimate.

In order to compare the results, a doublet benchmark has been considered, that is a quick consecutive variation of the attitude that has a zero

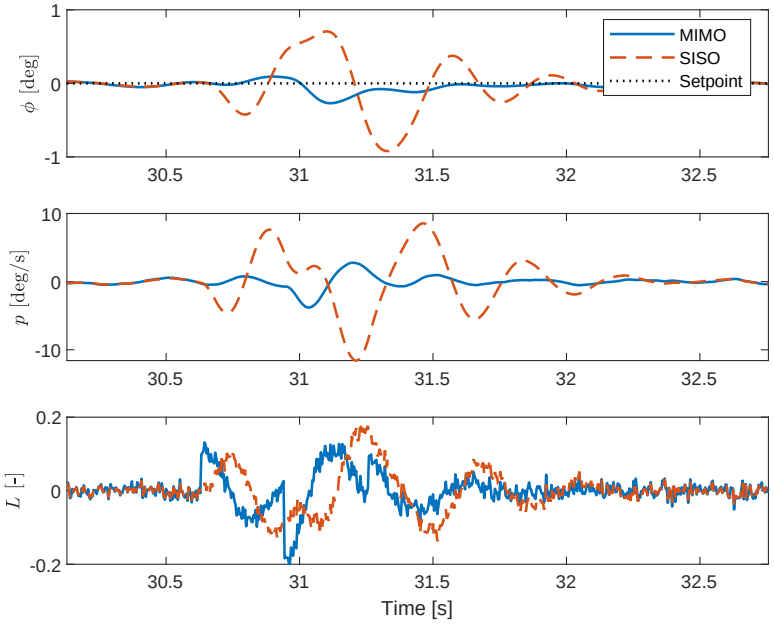
mean. The doublet period $T = 0.42$ s and amplitude $A = 22.5$ deg is held constant between tests. The simulated pitch doublet is shown in Figure 5.5, where it is highlighted that the full MIMO controller can significantly reduce the coupling effects. Couplings are reduced in both loops, almost cancelled for the MIMO regulator, which is also able to provide a faster response in the attitude dynamics.

The parameters of both algorithms are indicated side-by-side in Table 5.4. The diagonal terms are almost the same, with the exception of the integrator gains in the inner loop, which are significantly higher than the ones obtained from the SISO method.

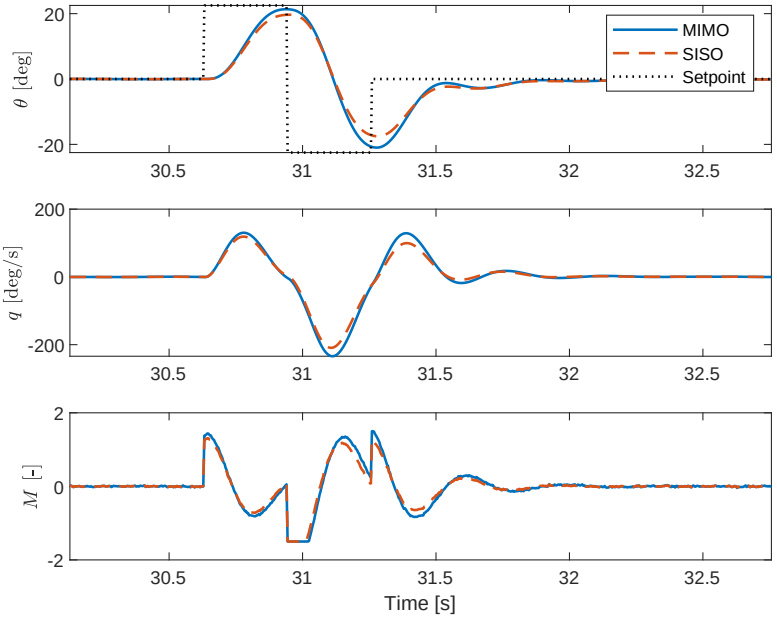
The outer loop gains of the P controller feature an almost identical term on the diagonal. Conversely, the off-diagonal terms are smaller by two orders of magnitude. This size means that the inner loop controller is able to decouple the dynamics effectively from the attitude rate and making them similar.

Table 5.4: ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop simulation data.

	SISO	**MIMO**
$\mathbf{K_p^O}$	$\begin{bmatrix} 4.6650 & 0 \\ 0 & 4.6772 \end{bmatrix}$	$\begin{bmatrix} 5.2090 & 0.0104 \\ 0.0114 & 5.1874 \end{bmatrix}$
$\mathbf{K_p^I}$	$\begin{bmatrix} 0.1490 & 0 \\ 0 & 0.1300 \end{bmatrix}$	$\begin{bmatrix} 0.1464 & 0.0094 \\ 0.0094 & 0.1275 \end{bmatrix}$
$\mathbf{K_i^I}$	$\begin{bmatrix} 0.1765 & 0 \\ 0 & 0.1807 \end{bmatrix}$	$\begin{bmatrix} 0.2630 & 0.0038 \\ 0.0038 & 0.2555 \end{bmatrix}$
$\mathbf{K_d^I}$	$\begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix}$	$\begin{bmatrix} 0.0005 & 0 \\ 0 & 0.0004 \end{bmatrix}$



(a) Roll response



(b) Pitch response

Figure 5.5: Simulation of a pitch angle doublet.

Chapter 6

Experimental testing and results

This chapter contains the experimental data that has been collected as part of the proposed algorithm validation seen in Chapter 3. The two aforementioned UAV platforms, ADAM-0 and ANT-R, will be employed in a series of experiments to test the tracking performance of the synthesised regulator using the SISO and the extended MIMO implementation of the VRFT algorithm.

For each platform, a data acquisition test is carried out in order to provide the I/O dataset required by the method. A pitch and roll moment PRBS input is injected in the control action, as described in Section 5.2. The collected dataset comprise the control actions $u(t) = \{\mathscr{L}(t), \mathscr{M}(t)\}^T$, the excitation input $\bar{u}(t) = \{\bar{\mathscr{L}}(t), \bar{\mathscr{M}}(t)\}^T$, the attitude angle $y_o(t) = \{\phi(t), \vartheta(t)\}^T$ and, finally, the angular rates $y_i(t) = \{p(t), q(t)\}^T$. A reference model is selected based on the desired requirements, by making suitable assumptions on the achievable performance.

Once the parameters have been obtained, a benchmark is chosen to evaluate performance indicators and compare the results. For this purpose, a doublet benchmark is adopted, with parameters changing based on expected behaviour and safety considerations.

6.1 ADAM-0

In this section, the tests carried out on the ADAM-0 platform are described, detailing the methodologies and specific test configuration. In particular, a validation test based on the original architecture is performed and it will be compared to traditional single-axis VRFT.

A second experiment feature the introduction of inertial coupling, obtained by positioning a concentrated mass close to motor 3, in a way that the

drone pitch and roll axis are no longer principal axes of inertia.

6.1.1 Nominal configuration

The experimental data is collected in the same way as presented for the simulation results, described in Section 5.2. The default controller parameters are used as the initial regulator, illustrated in Table 4.1.

Figure 6.1a and Figure 6.1b show the involved signal in the data-driven tuning procedure. These signals share the same specifications of the simulated experiment in Section 5.2. For the sake of clarity, the signals are represented in two figures, but were carried out subsequently during the same flight. Applying the inputs separately is not a requirement of the method, but it can increase the signal-to-noise ratio.

Reference models

The chosen reference models are second-order systems with the addition of a delay. The choice of the reference model in data-driven methods can affect the stability of the regulator. Thus, it might need adjustments between tests and lead to slightly different results, due to parameter uncertainty. In this case, the reference model chosen for the SISO algorithm is different from the one used for the simulation and MIMO formulation.

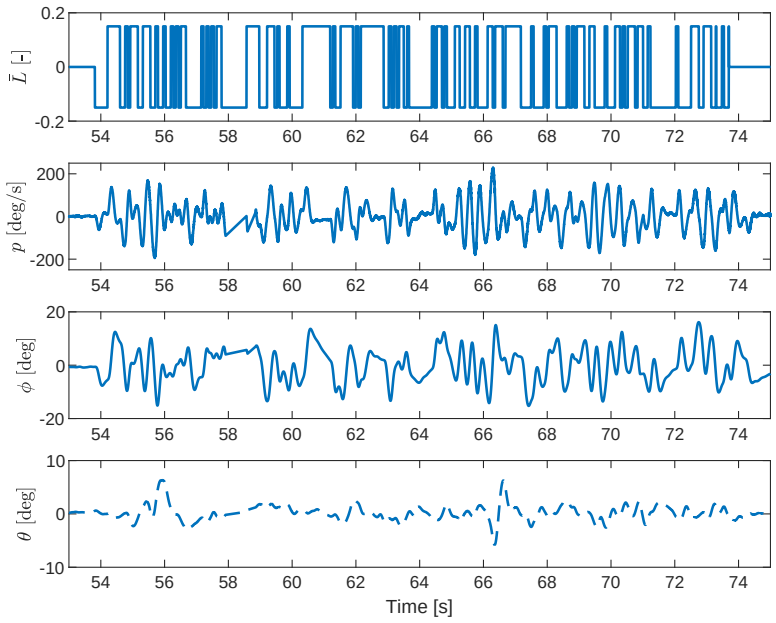
Table 6.1: Experimental data reference models for the inner and outer loops.

		ω **[rad/s]**	ζ	**Delay**
MIMO	Inner loop	18	0.3	1
	Outer loop	10	0.8	1
SISO	Inner loop	20	0.4	3
	Outer loop	10	0.7	3

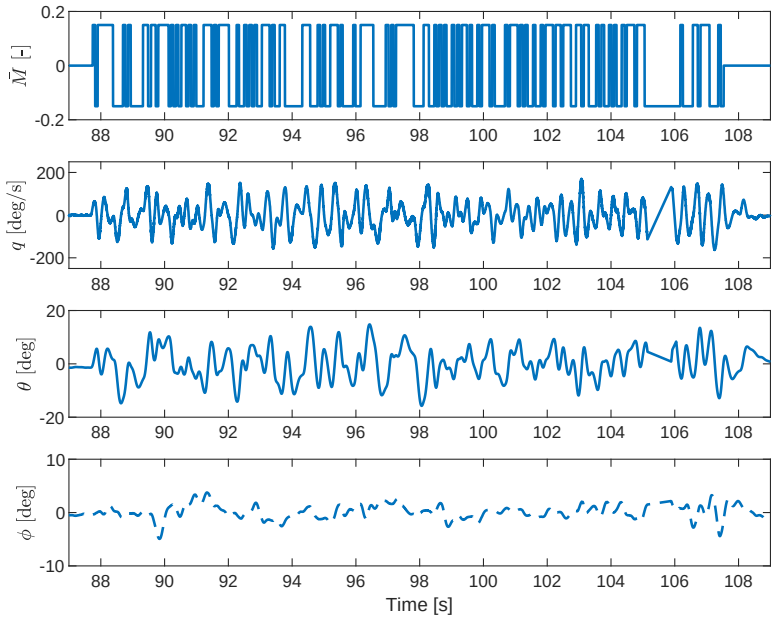
Controller parameter comparison

Given the reference models and closed-loop experimental dataset, the VRFT method leads to the parameter values reported in Table 6.2.

Since the data-collecting experiments are conducted in near-hovering conditions, the secondary diagonal of the parameters of the MIMO regulator in Table 6.2 is always one or more orders of magnitude smaller than the primary terms. This confirms an almost decoupled dynamics between the degrees of freedom in the quadrotor platform. The symmetry of the build is



(a) Roll excitation.



(b) Pitch excitation

Figure 6.1: ADAM-0: closed-loop experimental dataset.

Table 6.2: ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data.

	SISO	**MIMO**
$\mathbf{K_p^O}$	$\begin{bmatrix} 4.2505 & 0 \\ 0 & 4.2061 \end{bmatrix}$	$\begin{bmatrix} 4.4484 & -0.1287 \\ 0.2364 & 5.2220 \end{bmatrix}$
$\mathbf{K_p^I}$	$\begin{bmatrix} 0.1381 & 0 \\ 0 & 0.1495 \end{bmatrix}$	$\begin{bmatrix} 0.1187 & -0.0004 \\ 0.0044 & 0.1252 \end{bmatrix}$
$\mathbf{K_i^I}$	$\begin{bmatrix} 0.1039 & 0 \\ 0 & 0.3039 \end{bmatrix}$	$\begin{bmatrix} 0.1749 & 0.0023 \\ 0.0090 & 0.1164 \end{bmatrix}$
$\mathbf{K_d^I}$	$\begin{bmatrix} 0.0015 & 0 \\ 0 & 0.0027 \end{bmatrix}$	$\begin{bmatrix} 0.0007 & 0.0001 \\ 0.0001 & 0.0011 \end{bmatrix}$

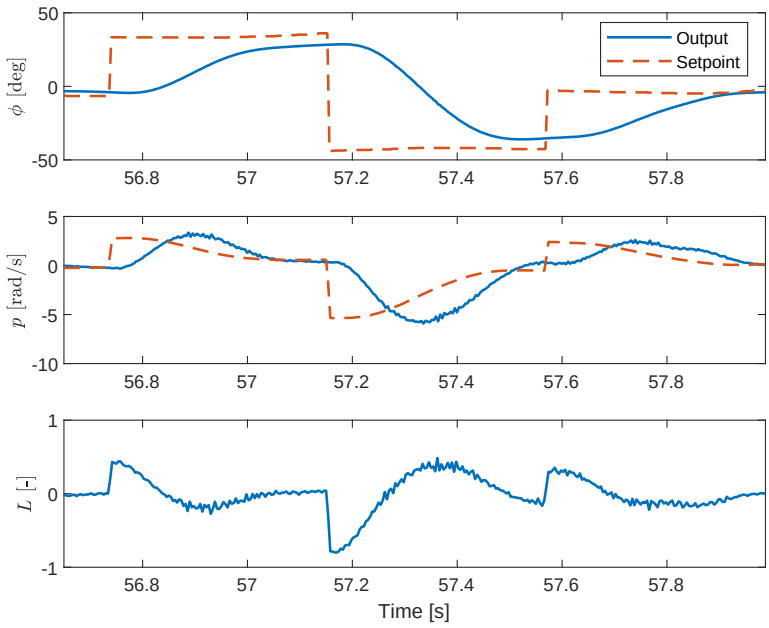
more evident when the SISO results are analysed, since pitch and roll parameters are very similar.

The benchmark for the performance comparison is a doublet, with period $T = 0.45$ s and amplitude $A = 40$ deg. Since the doublet experiment requires the position and velocity outer feedback loops to be disabled, experiments have been carried out by a pilot, leading to difficulties in replicating the exact input and conditions. For the sake of brevity, only one of the excitations is shown, as in this platform the results are identical.

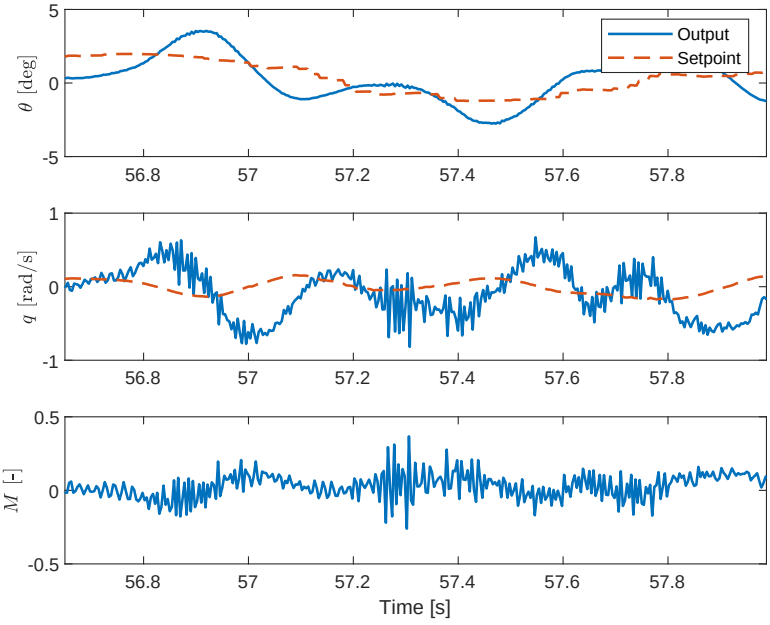
Preliminary tests are shown in Figures 6.2 and 6.3, where the roll doublet response shows a coupling, which in both cases is very limited, as expected. The pitch angle variations are about $\pm$ 3 deg; however, they are smoother for the MIMO solution. Note that the regulator obtained from the MIMO algorithm leads to a quicker response.

The MIMO controller features a quick suppression of the oscillations in the pitch rate loop, which can follow the rate setpoint given from the attitude feedback loop. Finally, the control effort on the pitch axis is stabilised and slightly reduced with respect to the SISO controller. To better evaluate the performance on the roll axis, the reference models have also been simulated for the measured attitude setpoint and compared to the responses of the roll rate and roll angle, shown in Figure 6.4.

Two cost functions can be defined in order to compare the results, one for the setpoint tracking error and one for the model tracking error, for which a qualitative result, based on the time domain response, has been described. The cost functions are defined for the roll performance, both for the angular

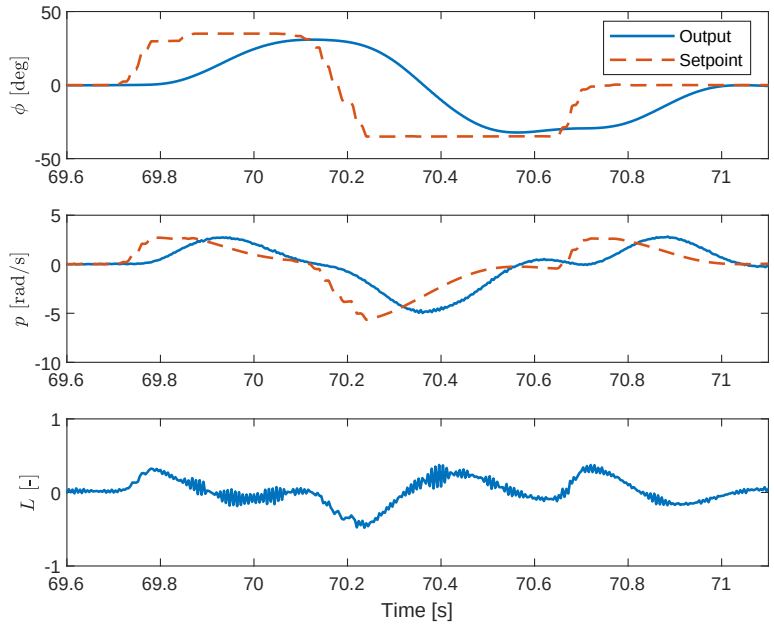


(a) Roll response

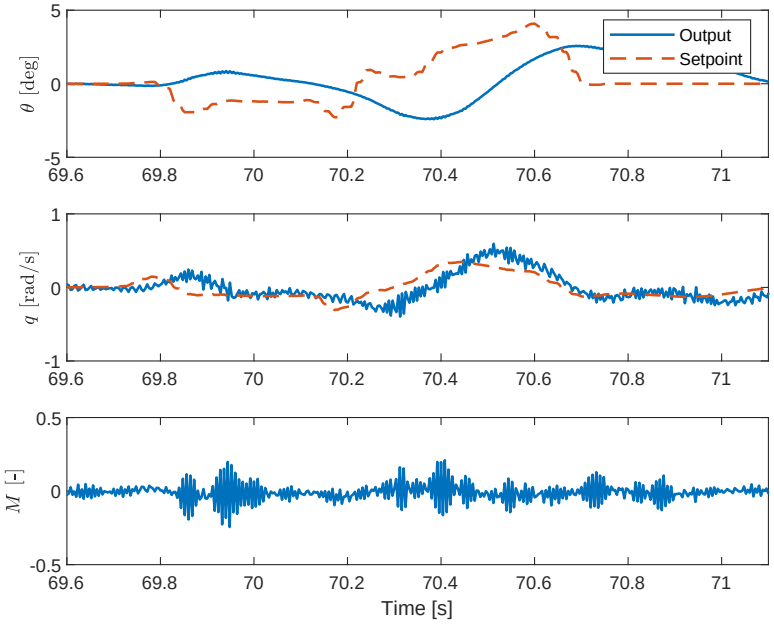


(b) Pitch response

Figure 6.2: ADAM-0: Roll doublet experiment with SISO method parameters.



(a) Roll response



(b) Pitch response

Figure 6.3: ADAM-0: Roll doublet experiment with MIMO method parameters.

rate and attitude:

$$J_{M,o} = \frac{1}{N}\sum_{k=k_0}^{k_0+N}\left(\phi(k) - M_o(z)\phi^o(k)\right)^2 \tag{6.1}$$

$$J_{r,o} = \frac{1}{N}\sum_{k=k_0}^{k_0+N}\left(\phi(k) - \phi^o(k)\right)^2 \tag{6.2}$$

$$J_{M,i} = \frac{1}{N}\sum_{k=k_0}^{k_0+N}\left(p(k) - M_i(z)\,p^o(k)\right)^2 \tag{6.3}$$

$$J_{r,i} = \frac{1}{N}\sum_{k=k_0}^{k_0+N}\left(p(k) - p^o(k)\right)^2, \tag{6.4}$$

where k_0 indicates the starting point of the doublet, while N is the number of samples of the doublet. $M_i(z)$ and $M_o(z)$ are respectively the reference model for the inner and outer loops. The same structure can also be applied to the pitch axis, by substituting the measured pitch angle ϑ, the angle setpoint ϑ^o, the measured pitch rate q and setpoint q^o. The subscript M indicates the model tracking cost, while the subscript r the setpoint tracking cost.

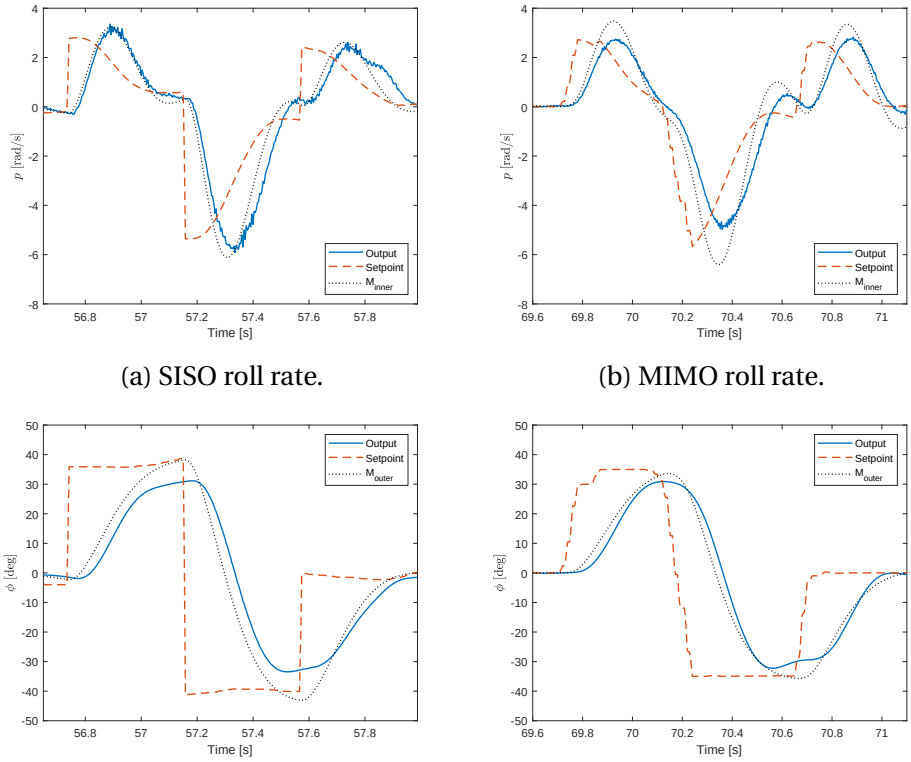
The results for the nominal configuration of the ADAM-0 platform are collected in Table 6.3. While the two inputs are not exactly replicated, some considerations can be made about the response shown in Figure 6.4. The MIMO controller performs better in the outer loop, both in terms of setpoint and model tracking. The SISO algorithm achieves higher performance in the inner controller model-tracking test, while setpoint tracking capabilities are inferior to those of the MIMO controller, as seen from the cost $J_{r,i}$.

Table 6.3: ADAM-0: roll performance cost functions from experimental data.

	$\mathbf{J_{M,o}}$	$\mathbf{J_{M,i}}$	$\mathbf{J_{r,o}}$	$\mathbf{J_{r,i}}$
MIMO	0.0033	0.5542	0.1612	2.1705
SISO	0.0122	0.3145	0.2645	3.3345

6.1.2 Perturbed configuration

A gyroscopic coupling is artificially introduced in the system to evaluate the decoupling capabilities of the MIMO algorithm on a real application. this has been obtained by inserting a localised mass under one of the rotors. The size of the frame permits the introduction of such effect with relatively small masses due to the large arm between the rotor and centre of mass of the



(a) SISO roll rate.

(b) MIMO roll rate.

(c) SISO roll angle.

(d) MIMO roll angle.

Figure 6.4: ADAM-0: Roll axis performance of the nominal configuration.

drone. In this case, a mass of approximately 160 g is located close to motor number 3, at a distance of approximately 20 cm from the centre.

Following the same procedure, a PRBS signal is applied to the input in order to collect the required output data to run the algorithm. The same PRBS parameters of the nominal configuration have been used. The collected dataset is shown in Figure 6.6. It can be seen that coupling effects are still limited, even after the introduction of the mass on motor 3. The main effect that can be observed is the PWM signal sent to the motors, where motor 3 sees a greater speed request in order to provide the additional thrust, shown in Figure 6.7. As a consequence, motor number 2 has a lower angular speed to avoid roll imbalance, which leaves motors 1 and 4 almost constant.

Reference models

To reduce control effort due to the additional payload, the inner dynamics reference model has been relaxed and had its natural frequency reduced to

Figure 6.5: ADAM-0: perturbed configuration.

15 rad/s. Damping ratio has been slightly increased to $\zeta = 0.33$. Due to the slower dynamics, the effect of the time delay has not been considered, obtaining the following transfer function:

$$M_i(z) = \frac{0.001776z + 0.001753}{z^2 - 1.958z + 0.9612}. \tag{6.5}$$

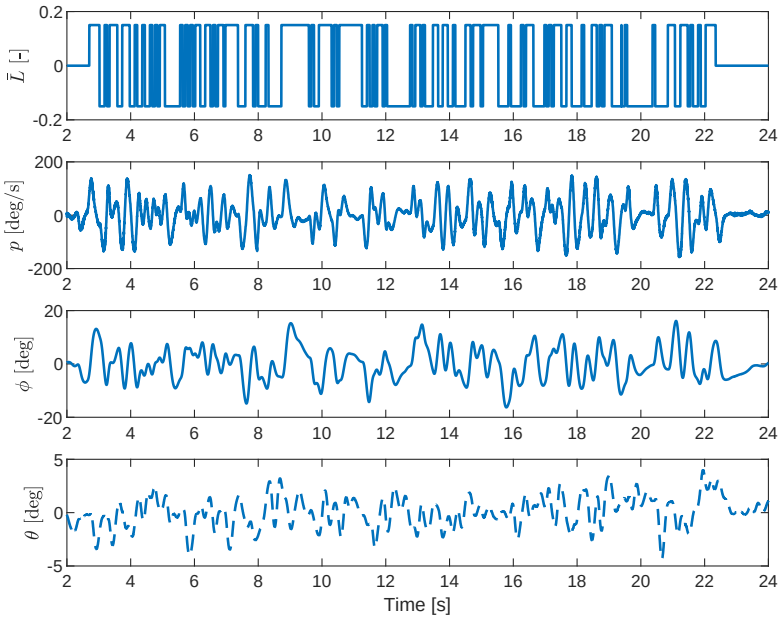
The outer loop (attitude) reference model is once again a second-order system with a natural frequency of 10 rad/s and a damping ratio of 0.8. The discrete transfer function that describes this dynamic is:

$$M_o(z) = \frac{0.0007831z + 0.0007666}{z^2 - 1.936z + 0.938}. \tag{6.6}$$

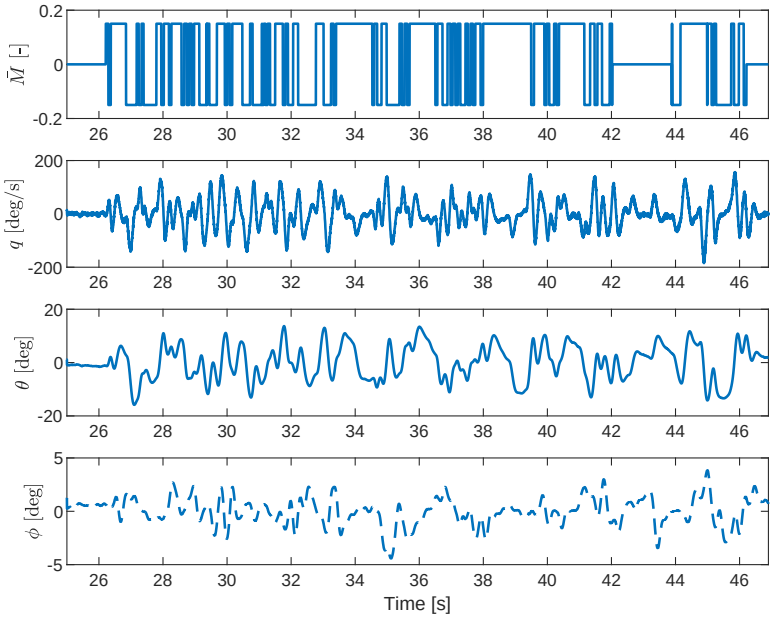
Results

By providing the obtained dataset from the PRBS excitation and the reference models above, the VRFT algorithm provides the parameter estimates shown in Table 6.4. It is noted that the SISO algorithm is more susceptible to the dataset and reference model, since using the collected dataset did not lead to compatible parameters for the controller. For this reason, the regulator obtained from the MIMO derivation will be compared to the controller obtained from the SISO algorithm tuned with data collected from the nominal configuration experiment. The same reference model of Equations (6.5) and (6.6) is used.

For safety reasons, the doublet parameters have been lowered to reduce potential risks related to actuator saturation, since the added mass has increased the control effort in hovering conditions. The doublet magnitude has been set to 15 deg, while the period has been increased to 0.8 s.



(a) Roll excitation.



(b) Pitch excitation

Figure 6.6: ADAM-0: closed-loop experimental dataset of the perturbed configuration.

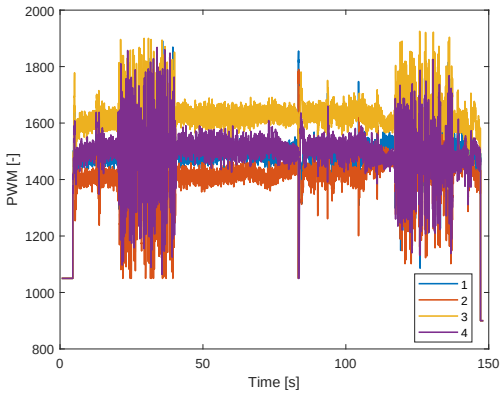


Figure 6.7: PWM input during the identification test flight.

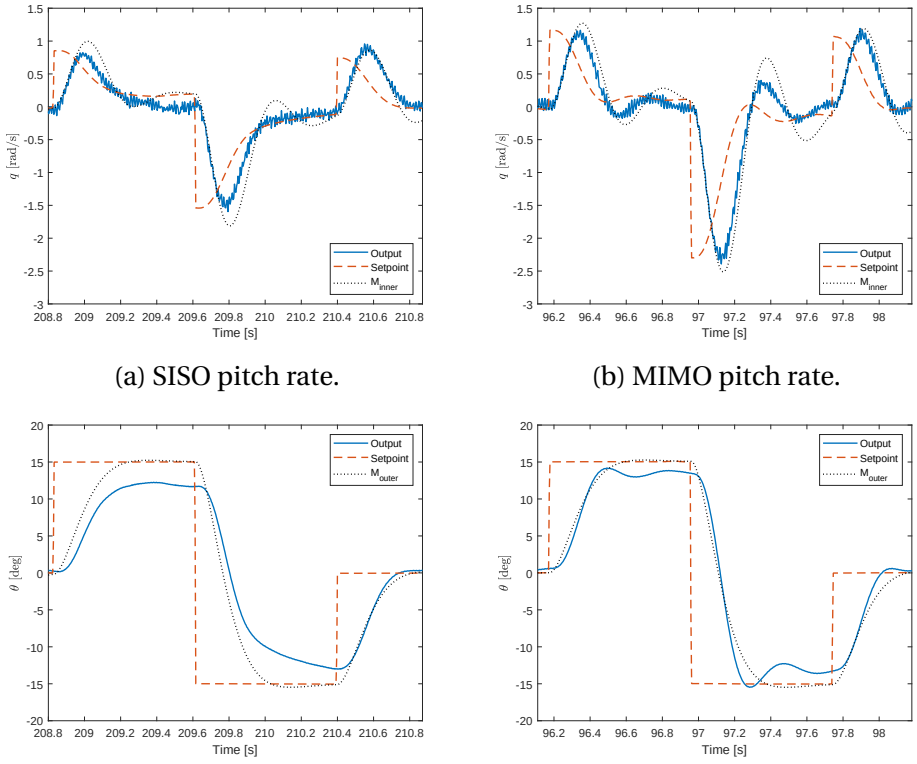
Table 6.4: ADAM-0: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data of the perturbed configuration.

	SISO	**MIMO**
$\mathbf{K_p^O}$	$\begin{bmatrix} 3.3546 & 0 \\ 0 & 3.3075 \end{bmatrix}$	$\begin{bmatrix} 4.0081 & -0.0805 \\ 0.1154 & 4.6357 \end{bmatrix}$
$\mathbf{K_p^I}$	$\begin{bmatrix} 0.0984 & 0 \\ 0 & 0.1028 \end{bmatrix}$	$\begin{bmatrix} 0.0964 & -0.0065 \\ -0.0076 & 0.0942 \end{bmatrix}$
$\mathbf{K_i^I}$	$\begin{bmatrix} 0.2213 & 0 \\ 0 & 0.3496 \end{bmatrix}$	$\begin{bmatrix} 0.2286 & 0.1025 \\ 0.0034 & 0.2380 \end{bmatrix}$
$\mathbf{K_d^I}$	$\begin{bmatrix} 0.0007 & 0 \\ 0 & 0.0015 \end{bmatrix}$	$\begin{bmatrix} 0.0005 & 0.0001 \\ -0.0001 & 0.0005 \end{bmatrix}$

From the doublet responses, shown from Figure 6.9 to 6.12, it can be observed that the coupling effects are comparable on both plants, even for this attempt at introducing them. The main visible effect is an offset of the control action, required to balance the effect of the mass positioned on one side. Once again, the full MIMO regulator allows for a quicker setpoint tracking on the attitude angle.

Analogously to the previous analysis of the nominal configuration, the given reference models are simulated to evaluate reference tracking and model tracking capabilities of the synthesised controllers. The numerical results are shown in Table 6.5, while the time domain response is shown in Figure 6.8.

The cost functions are consistent with the response shown in Figure 6.8,



(a) SISO pitch rate.

(b) MIMO pitch rate.

(c) SISO pitch angle.

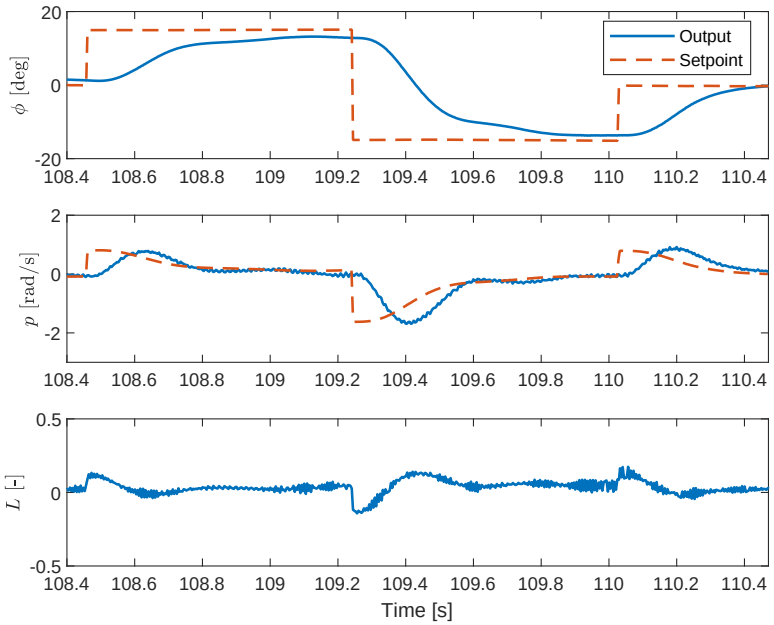
(d) MIMO pitch angle.

Figure 6.8: ADAM-0: Pitch axis performance of the perturbed configuration.

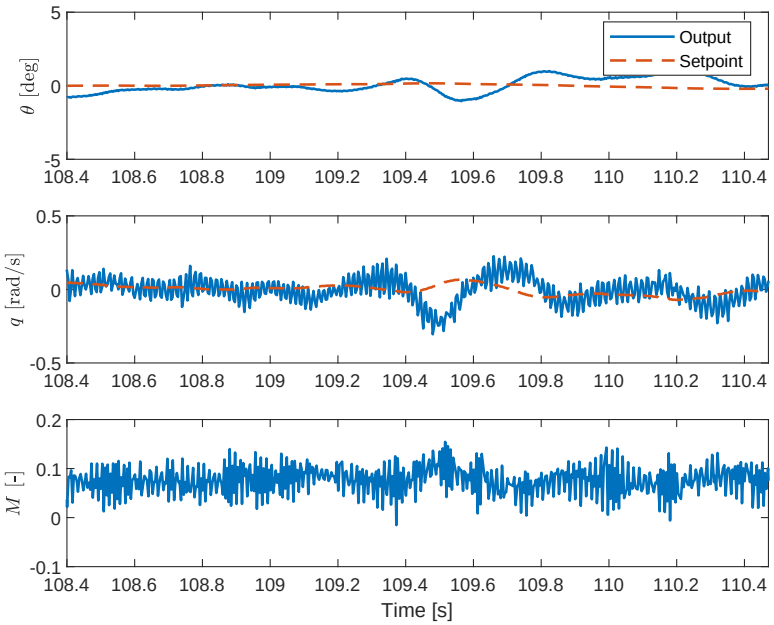
in particular for the outer loop performance. The performance inside the inner loop is well captured in terms of model tracking for the MIMO regulator, which is also slightly better in following the given setpoint, as confirmed by the cost function. In the outer loop, the behaviour of the SISO controller is much slower than the original request, whereas the MIMO algorithm is quicker. Although it features some oscillations, it is more performant overall.

Table 6.5: ADAM-0: pitch performance cost functions from experimental data of the perturbed configuration.

	$J_{M,o}$	$J_{M,i}$	$J_{r,o}$	$J_{r,i}$
MIMO	0.0006	0.0452	0.0158	0.2649
SISO	0.0457	0.3839	0.0581	0.3059

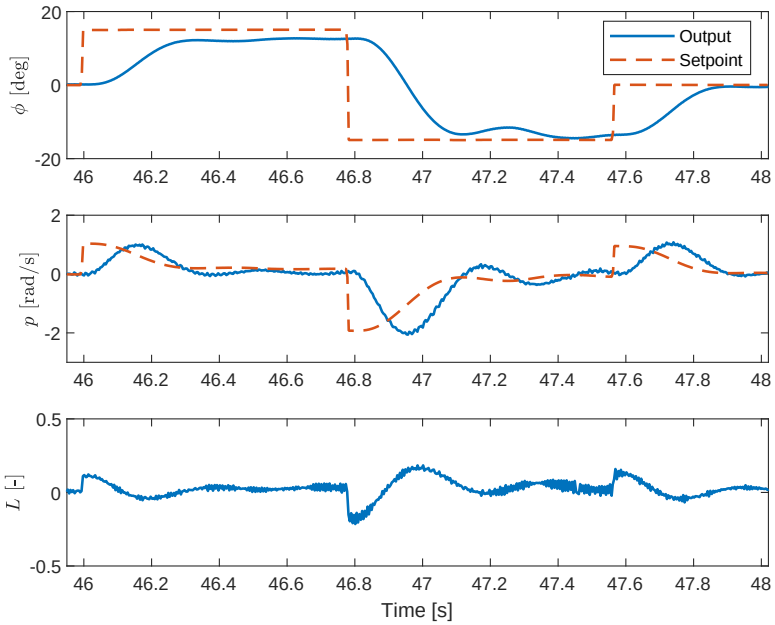


(a) Roll response

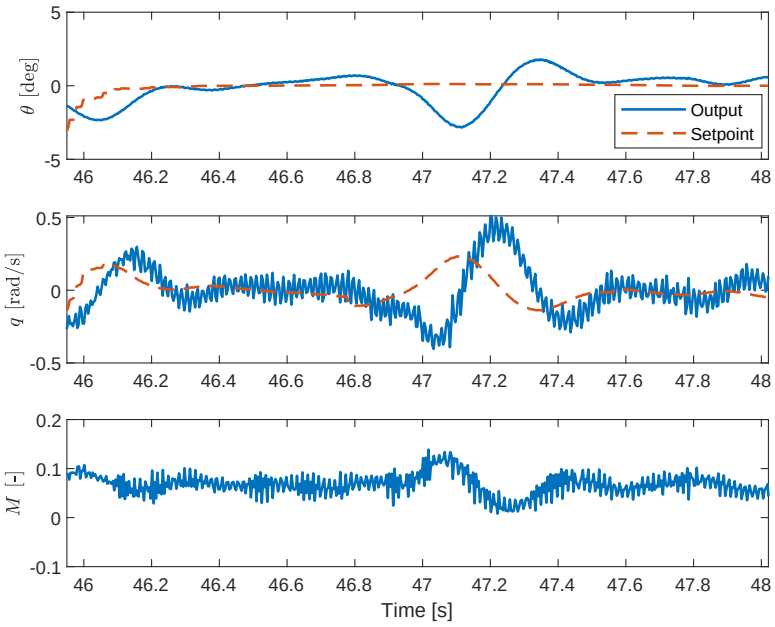


(b) Pitch response

Figure 6.9: ADAM-0: Roll doublet experiment with SISO method parameters on the perturbed configuration.

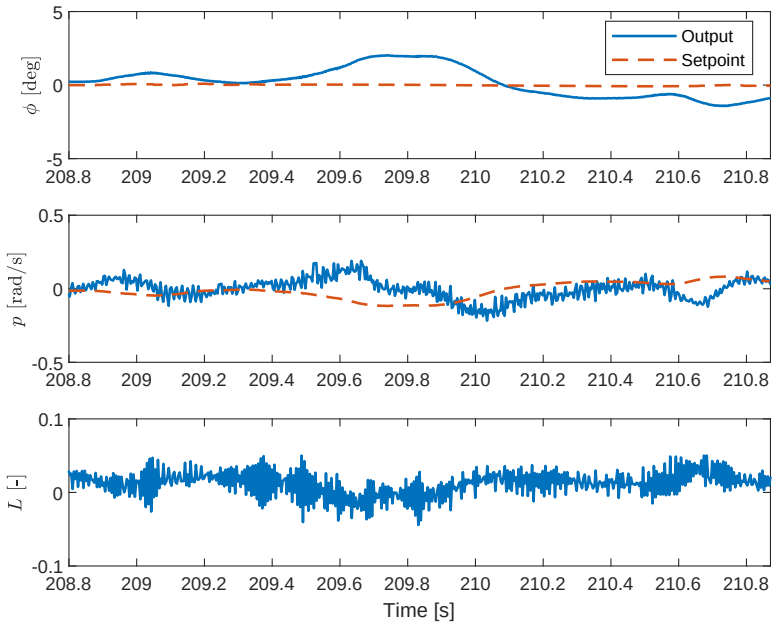


(a) Roll response

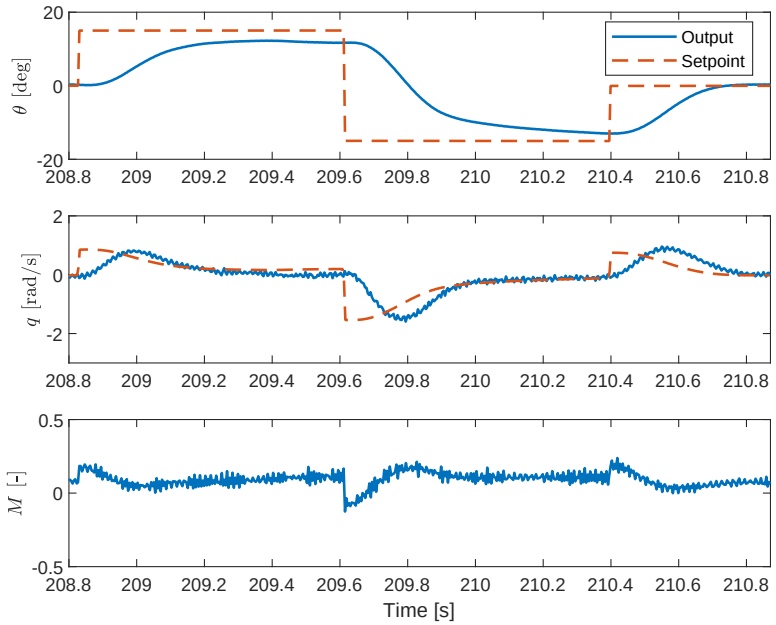


(b) Pitch response

Figure 6.10: ADAM-0: Roll doublet experiment with MIMO method parameters on the perturbed configuration.

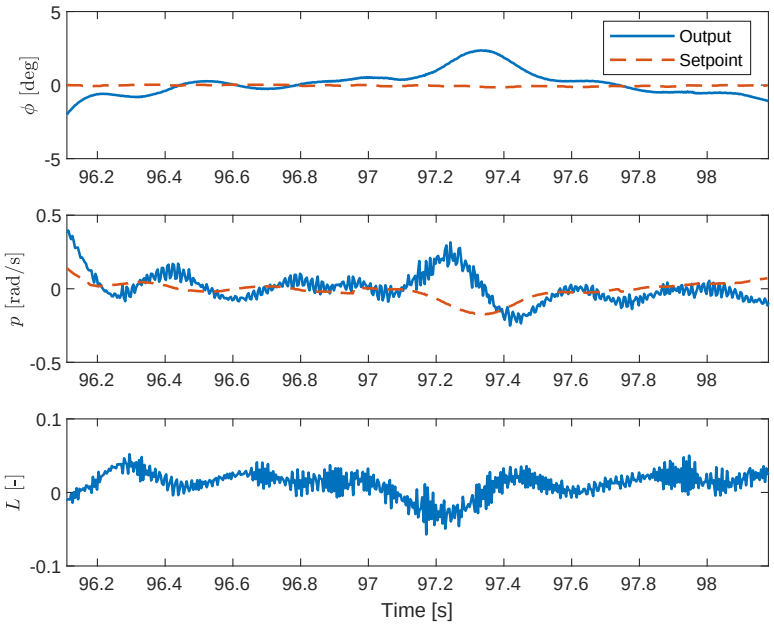


(a) Roll response

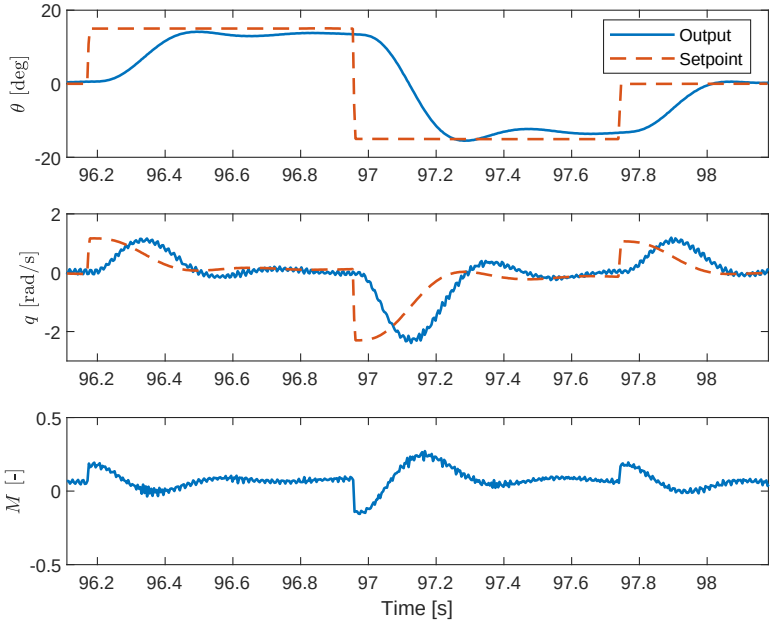


(b) Pitch response

Figure 6.11: ADAM-0: Pitch doublet experiment with SISO method parameters on the perturbed configuration.



(a) Roll response



(b) Pitch response

Figure 6.12: ADAM-0: Pitch doublet experiment with MIMO method parameters on the perturbed configuration.

6.2 ANT-R

In this section, the ANT-R platform is tested to validate the algorithm on a different machine. As seen in Section 4, ANT-R feature asymmetric inertia properties for the pitch and roll axes. In particular, a higher inertia characterises the pitch axis. Results from the SISO and MIMO algorithm will be compared, with a focus on the discrepancy between the actual performance and the desired one.

For the I/O dataset, a PRBS input has been used as well, with a frequency range of interest between 0-50 rad/s, with a normalised moment of 0.1, that is an injection of 10% of the maximum applicable moment, for a duration of 15 s per channel. Experimental data is shown in Figure 6.13. The initial controller for the plant has been slightly modified from the default values to improve stability. The parameters are shown in Table 6.6.

Table 6.6: ANT-R: initial controller parameters.

$\mathbf{K_p^O}$	$\mathbf{K_{ff}^I}$	$\mathbf{K_p^I}$	$\mathbf{K_i^I}$	$\mathbf{K_d^I}$
$\begin{bmatrix} 6.5 & 0 \\ 0 & 6.0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0.07 \end{bmatrix}$	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$	$\begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$

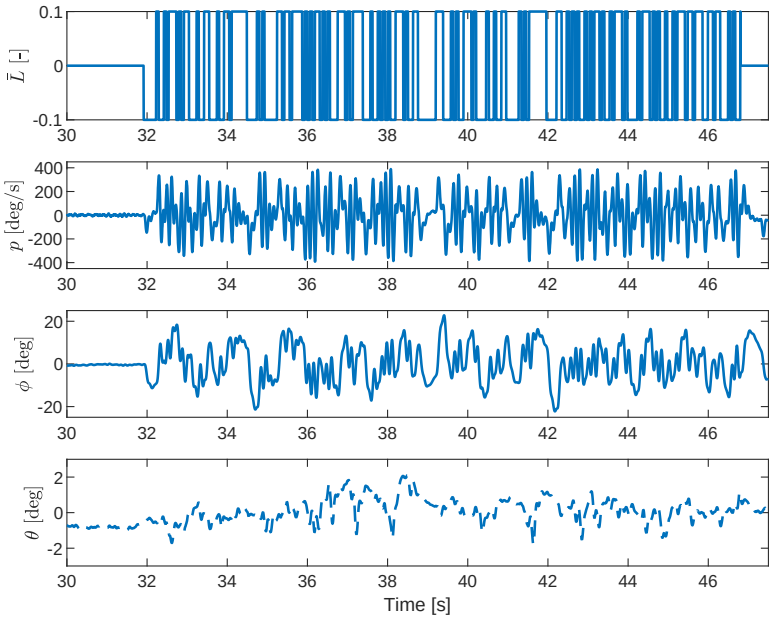
Reference models

A second-order system has been chosen as the reference model, similarly to the ADAM-0 platform. Since this drone is much lighter and features higher performance motors, quicker dynamics have been set. A natural frequency of 30 rad/s and a damping ratio of 0.3 determine the angular rate desired transfer function. The off-diagonal terms are set to zero, as coupling between the roll and pitch rate dynamics is undesirable. The sampling rate of the controller is the same as ADAM-0 (250 Hz), for which we obtain the following transfer function:

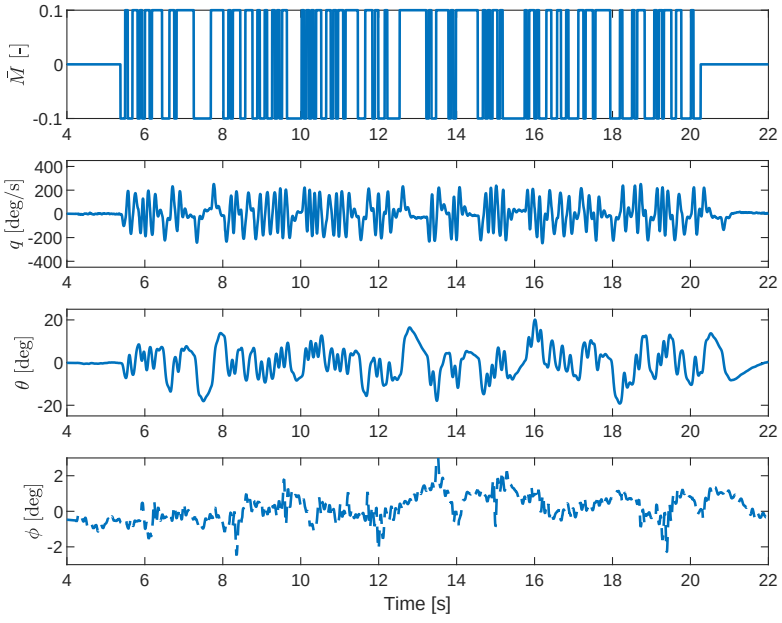
$$M_i(z) = z^{-1} \frac{0.007022z + 0.006855}{z^2 - 1.917z + 0.9305}. \tag{6.7}$$

Similarly, for the outer attitude dynamics, a second-order model is selected, with a natural frequency of 20 rad/s and a damping ratio of 0.8, since overshoots of the setpoint are undesirable for the attitude angles. The transfer function can be obtained:

$$M_o(z) = z^{-1} \frac{0.003066z + 0.002938}{z^2 - 1.874z + 0.8799}. \tag{6.8}$$



(a) Roll excitation.



(b) Pitch excitation

Figure 6.13: ANT-R: closed-loop experimental dataset used by MIMO data-driven method.

6.2.1 Parameter results comparison

The reference models are the same for both algorithms, which provide results summarised in Table 6.7.

Table 6.7: ANT-R: controller parameters for outer and inner controllers considering the VRFT method with closed-loop experimental data.

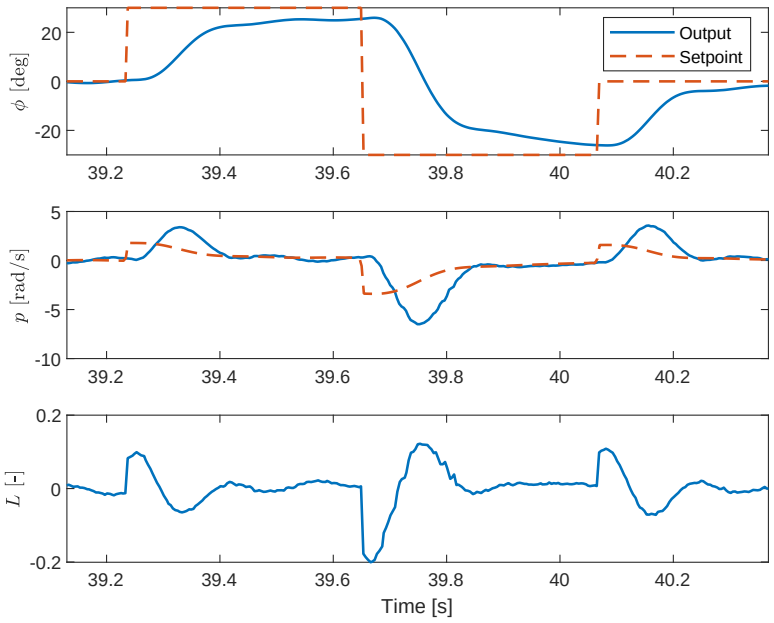
	SISO	MIMO
$\mathbf{K_p^O}$	$\begin{bmatrix} 6.1029 & 0 \\ 0 & 6.2546 \end{bmatrix}$	$\begin{bmatrix} 8.4068 & 0.1209 \\ -0.0166 & 8.9678 \end{bmatrix}$
$\mathbf{K_p^I}$	$\begin{bmatrix} 0.0300 & 0 \\ 0 & 0.0469 \end{bmatrix}$	$\begin{bmatrix} 0.0230 & -0.0001 \\ 0 & 0.0396 \end{bmatrix}$
$\mathbf{K_i^I}$	$\begin{bmatrix} 0.2157 & 0 \\ 0 & 0.2323 \end{bmatrix}$	$\begin{bmatrix} 0.0481 & 0.0152 \\ 0.0138 & 0.1265 \end{bmatrix}$
$\mathbf{K_d^I}$	$\begin{bmatrix} 0.0004 & 0 \\ 0 & 0.0003 \end{bmatrix}$	$\begin{bmatrix} 0.0002 & 0 \\ 0 & 0.0002 \end{bmatrix}$

The controllers are then tested on the drone, and their performance is evaluated based on a doublet benchmark. The roll excitation is presented in Figure 6.14, showcasing the response of the SISO controller, and the same in Figure 6.15 for the MIMO controller.

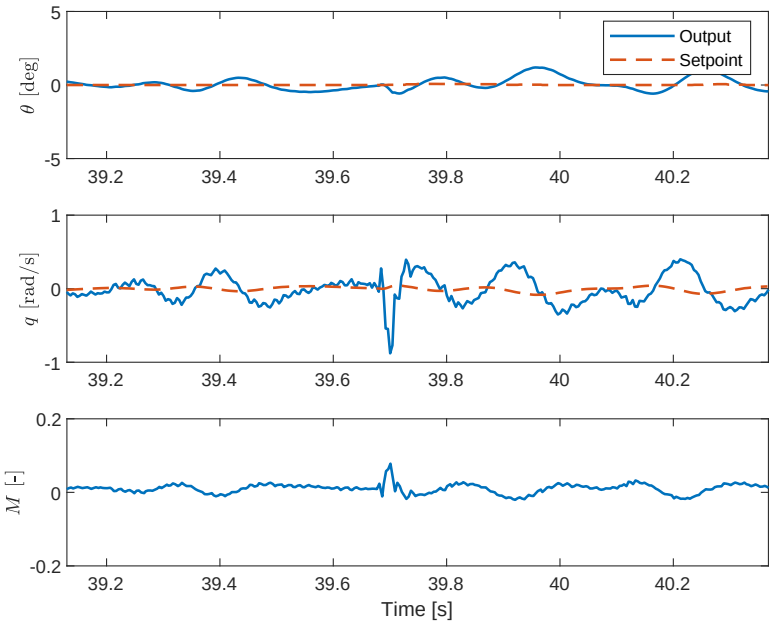
It can be seen that the SISO controller is much slower in the angle dynamics than the MIMO regulator, especially after the change of sign of the setpoint, when a variation of 30 deg is applied. The slower response of the outer loop depends on the gains, as it is only a P controller: the gains for the SISO implementation are smaller by at least 25% with respect to the diagonal terms of the MIMO outer loop controller. An additional consideration can be made regarding the control effort, as the MIMO controller uses a comparable input while achieving substantially better performance in terms of attitude tracking.

Both regulators lead to small oscillations around zero on the pitch axis when a roll doublet is executed. These oscillations are observed in both the angle and angular rate, however, it may be considered well decoupled, as expected. This result can be seen in the MIMO controller gains reported in Table 6.7, wherein the off-diagonal terms are lower by one or more orders of magnitude.

Finally, a comparison between the desired roll response and actual response, as previously carried out, is shown in Figure 6.16. The cost functions, defined in Equations (6.2) to (6.4), are evaluated and analysed. The results are shown in Table 6.8.

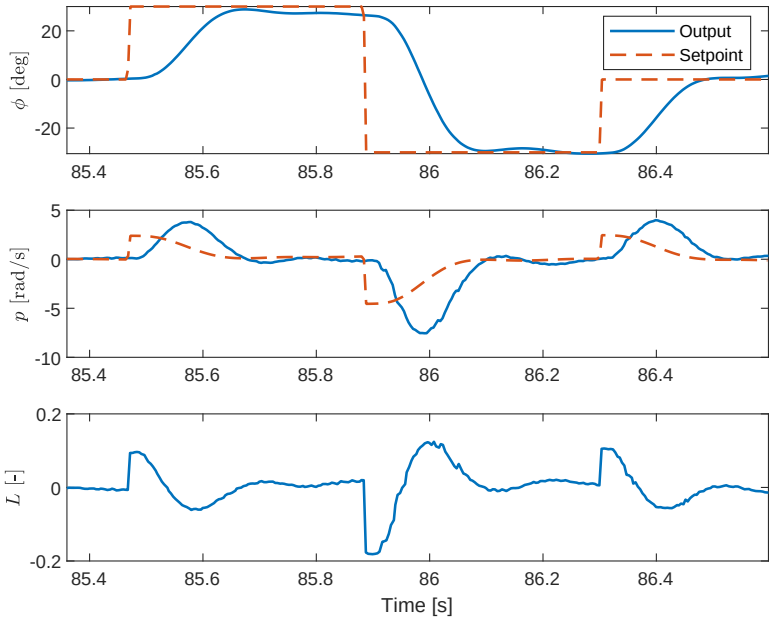


(a) Roll response

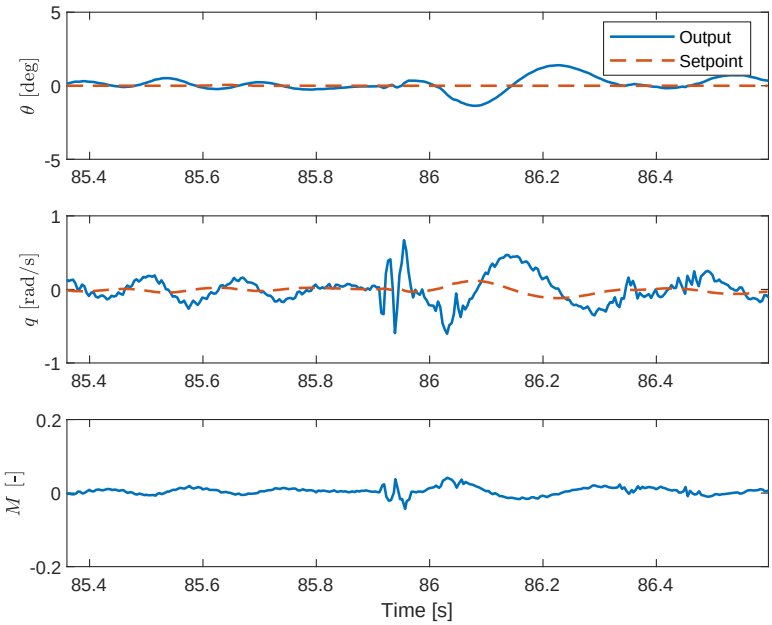


(b) Pitch response

Figure 6.14: ANT-R: Roll doublet experiment with SISO method parameters.

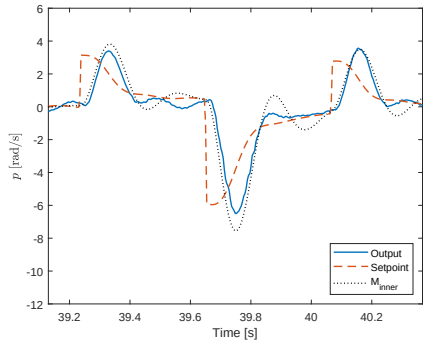


(a) Roll response

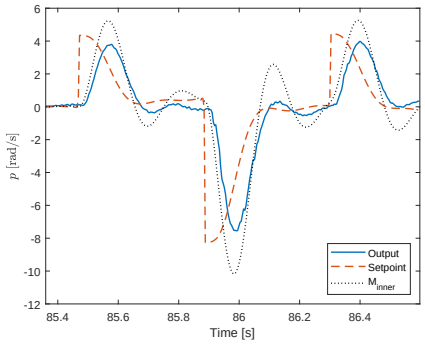


(b) Pitch response

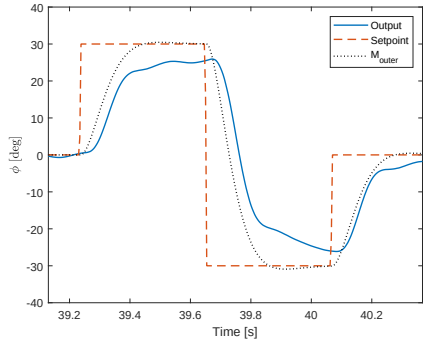
Figure 6.15: ANT-R: Roll doublet experiment with MIMO method parameters.



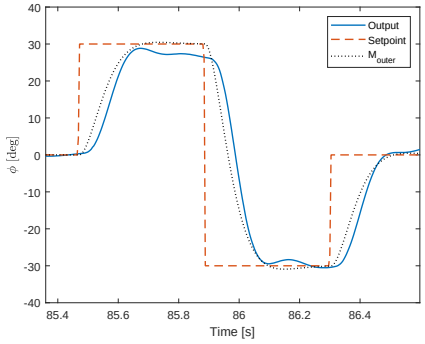
(a) SISO roll rate.



(b) MIMO roll rate.



(c) SISO roll angle.



(d) MIMO roll angle.

Figure 6.16: ANT-R: Roll axis performance.

In both cases, in the angular rate dynamics, there are evident oscillations around the setpoint. These are also present in the desired reference model response as the damping ratio is small. It is noted that in this case, the SISO controller has a better model-following capability for the attitude rate loop. Conversely, the MIMO controller is much quicker in following the requirement for the attitude loop. This result is not limited to the model tracking capabilities, but also in terms of setpoint tracking, where the SISO algorithm is not able to reach the imposed reference angle. The response is coherent with the cost function evaluations.

Table 6.8: ANT-R: roll performance cost functions from experimental data.

	$J_{M,o}$	$J_{M,i}$	$J_{r,o}$	$J_{r,i}$
MIMO	0.0047	1.4315	0.1024	4.8303
SISO	0.0130	0.3773	0.1103	2.5215

Chapter 7

Conclusions

The problem of data-driven design of the attitude control law for a multi-rotor UAV has been considered. The VRFT method has been extended to a more general class of controllers and by allowing the closed-loop execution of data-collection experiments on the system. The class of PRBS input signals has been selected which can be easily reproduced and lead to a sufficiently informative dataset in terms of attitude dynamics. Experimental results show that the in-flight tests can be conducted safely and that a satisfactory level of performance can be achieved by using 20 seconds datasets.

The obtained regulators have been compared for both platforms, distinguishing between reference tracking and model following capabilities as part of the framework of model reference control. From the experiments that have been carried out, the couplings between inputs were limited, leading to almost diagonal MIMO regulators for the nominal configurations of ADAM-0 and ANT-R, with a comparable off-axis response. Significant differences have been observed when the channel response is considered: in the attitude loop the SISO regulator always failed to match the requirements, while achieving a comparable, but inconsistent, result in the angular rate loop. The ultimate goal of the attitude control law is to regulate the attitude dynamics, meaning that in this sense the MIMO formulation was able to provide a better overall result with comparable control effort consistently.

In the perturbed configuration of the ADAM-0 platform, an artificial coupling was introduced, but it was bounded for safety concerns due to the limited payload capabilities of the platform. This lead one off-diagonal component of the integral gain to have a magnitude comparable to the main terms on the diagonal, even if from simulations it was expected to affect both components. However, couplings can arise from a number of dynamic and aerodynamic effects, not only from gyroscopic terms. In general, such influence is difficult to model accurately, as most of the causes are bound to nonlinear

behaviour. These effects make this class of synthesis methods very appealing, since almost no assumptions on the system are made. For plants that feature high levels of coupling, such as the ones covered in the simulation chapter, the MIMO algorithm can provide a satisfactory solution in terms of performance, being able to significantly separate the dynamics.

As with other data-driven methods, no stability constraint is enforced on the algorithm, making the solution of the method reliant on the choice of a suitable reference model. The reference model is critical as it must be compatible with both the dataset and the physical system limitations. This means that a build-up approach to experimental testing is suggested where even limited prior knowledge is unavailable.

The main advantages over the SISO formulation are that the instrumental variable parameters (model order and past/future windowing of PBSID) are not needed, reducing the number of algorithm tunable variables. Furthermore, for one possible choice of the instrumental variable for the SISO algorithm, the initial controller must be known, while in the MIMO formulation it is no longer necessary.

To summarise the main findings, this method allows a more general approach to the attitude control problem, reducing the number of tuning parameters and removing the hypothesis of symmetrical builds and decoupled dynamics, obtaining the same information as the result of the algorithm itself, while achieving good levels of performance.

Future developments

The derived method could be applied to an existing drone platform, where a suitable reference model is chosen and proven, in a way that a quick experiment can allow for the fast, or even automatic, retuning of the controller parameters. For instance, it could be applied after an operating condition change, *e.g.* a payload introduction, or to account for the ageing of components, which cause a degradation of performance over time.

Future developments of this work could focus on the complete formulation of the control laws of the UAV platform, thus including the simultaneous tuning of the position and velocity feedback loops, which has been carried out in [16] for the SISO VRFT algorithm.

Finally, the proposed approach has been developed in a multivariable framework, which might be of little practical interest in UAVs due to inherent symmetry. However, it constitutes an essential intermediate step towards the application in helicopters attitude control design, for which closed-loop testing in near hovering conditions could be feasible. This method is still

lacking a way to ensure closed-loop stability prior to the actual implementation, which might be unacceptable for crewed aircraft or even expensive UAV platforms. The introduction of a constraint on stability should be investigated.

Bibliography

[1] S. Formentin, K. Van Heusden, and A. Karimi, "A comparison of model-based and data-driven controller tuning," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 882–897, 2014.

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

[3] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.

[4] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*, vol. 2. Instrument society of America Research Triangle Park, NC, 1995.

[5] S. Formentin, S. Savaresi, and L. Del Re, "Non-iterative direct data-driven controller tuning for multivariable systems: theory and application," *IET control theory & applications*, vol. 6, no. 9, pp. 1250–1257, 2012.

[6] S. Formentin, M. C. Campi, and S. M. Savaresi, "Virtual reference feedback tuning for industrial pid controllers," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11275–11280, 2014.

[7] F. Previdi, T. Schauer, S. M. Savaresi, and K. J. Hunt, "Data-driven control design for neuroprotheses: a virtual reference feedback tuning (vrft) approach," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 176–182, 2004.

[8] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE control systems magazine*, vol. 18, no. 4, pp. 26–41, 1998.

[9] H. Hjalmarsson, "Iterative feedback tuning—an overview," *International journal of adaptive control and signal processing*, vol. 16, no. 5, pp. 373–395, 2002.

[10] K. Van Heusden, A. Karimi, and D. Bonvin, "Data-driven model reference control with asymptotically guaranteed stability," *International Journal of Adaptive Control and Signal Processing*, vol. 25, no. 4, pp. 331–351, 2011.

[11] A. Karimi, K. Van Heusden, and D. Bonvin, "Non-iterative data-driven controller tuning using the correlation approach," in *2007 European Control Conference (ECC)*, pp. 5189–5195, IEEE, 2007.

[12] L. Miskovic, A. Karimi, D. Bonvin, and M. Gevers, "Correlation-based tuning of linear multivariable decoupling controllers," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 7144–7149, IEEE, 2005.

[13] G. O. Guardabassi and S. M. Savaresi, "Virtual reference direct design method: an off-line approach to data-based control system design," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 954–959, 2000.

[14] M. C. Campi, A. Lecchini, and S. M. Savaresi, "Virtual reference feedback tuning: a direct method for the design of feedback controllers," *Automatica*, vol. 38, no. 8, pp. 1337–1346, 2002.

[15] T. Söderström and P. Stoica, "Instrumental variable methods for system identification," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 1–9, 2002.

[16] S. Capocchiano, "Data-driven attitude and position control design for micro aerial vehicles," 2017.

[17] S. Capocchiano, P. Panizza, D. Invernizzi, and M. Lovera, "Closed-loop data-driven attitude control design for a multirotor uav," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 153–158, IEEE, 2018.

[18] S. Formentin, A. Cologni, D. Belloli, F. Previdi, and S. M. Savaresi, "Fast tuning of cascade control systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10243–10248, 2011.

[19] G. Van der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, "Closed-loop subspace identification methods: an overview," *IET Control Theory & Applications*, vol. 7, no. 10, pp. 1339–1358, 2013.

[20] A. Chiuso, "The role of vector autoregressive modeling in predictor-based subspace identification," *Automatica*, vol. 43, no. 6, pp. 1034–1048, 2007.

[21] S. Formentin, A. Karimi, and S. M. Savaresi, "Optimal input design for direct data-driven tuning of model-reference controllers," *Automatica*, vol. 49, no. 6, pp. 1874–1882, 2013.

[22] D. Selvi, D. Piga, and A. Bemporad, "Towards direct data-driven model-free design of optimal controllers," in *2018 European Control Conference (ECC)*, pp. 2836–2841, IEEE, 2018.

[23] M. Yeddanapudi and A. Potvin, *Nonlinear control design blockset: User's guide*. 1997.

[24] H. Hjalmarsson, "Efficient tuning of linear multivariable controllers using iterative feedback tuning," *International journal of adaptive control and signal processing*, vol. 13, no. 7, pp. 553–572, 1999.

[25] S. Formentin and S. M. Savaresi, "Noniterative data-driven design of multivariable controllers," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 5106–5111, IEEE, 2011.

[26] L. Campestrini, D. Eckhard, L. A. Chía, and E. Boeira, "Unbiased mimo vrft with application to process control," *Journal of Process Control*, vol. 39, pp. 35–49, 2016.

Appendix A

Kronecker Product

If A is a $m \times n$ matrix and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}. \tag{A.1}$$

This product, unlike regular matrix product, can be always done, with no requirements on the order of the two matrices, allowing the operation also for vectors.

Important properties

1. Bi-linearity and associativity

$$A \otimes (B + C) = A \otimes B + A \otimes B \tag{A.2}$$
$$(A + B) \otimes C = A \otimes C + B \otimes C \tag{A.3}$$
$$(kA) \otimes B = A \otimes (kB) = k(A \otimes B) \tag{A.4}$$
$$(A \otimes B) \otimes C = A \otimes (B \otimes C). \tag{A.5}$$

2. Non-commutative: $A \otimes B$ and $B \otimes A$ are in general different matrices, however they are permutation equivalent, meaning that there exist matrices P and Q such that

$$A \otimes B = P(B \otimes A)Q. \tag{A.6}$$

   If A and B are square matrices, then $P = Q^T$.

3. Mixed product

$$(A \otimes B)(C \otimes D) = (AB) \otimes (BD). \tag{A.7}$$

4. Transpose

$$(A \otimes B)^T = A^T \otimes B^T. \tag{A.8}$$

5. Spectrum: for square matrices A and B, respectively of order n and m, it follows that

$$\operatorname{tr}(A \otimes B) = \operatorname{tr}A \operatorname{tr}B \tag{A.9}$$

$$\det(A \otimes B) = \det(A)^m \det(B)^n. \tag{A.10}$$

   Notice that the exponent of $\det A$ is the order of B and vice-versa.

6. Inverse: $A \otimes B$ is invertible if and only if A and B are invertible, then

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{A.11}$$

7. Equivalence: Given $AXB = C$, with suitable orders to allow for matrix multiplication, it can be rewritten in the following form

$$(B^T \otimes A) \operatorname{vec}(X) = \operatorname{vec}(AXB) = \operatorname{vec}(C). \tag{A.12}$$

   In the case of two matrices, it can be showed that:

$$\operatorname{vec}(AB) = \operatorname{vec}(IAB) = (B^T \otimes I) \operatorname{vec}(A). \tag{A.13}$$

   This property can be exploited to transform a product between a matrix and a vector. Given matrix A and vector b, such that multiplication is allowed, then

$$\operatorname{vec}(Ab) = Ab = (b^T \otimes I) \operatorname{vec}(A). \tag{A.14}$$

A.1 PID output in linearised form

It is possible to write the output at time t, $u(t) = u(kT_s)$ where T_s will be omitted, of the MIMO discrete PID controller of order n_u in a linearised form, based on the current and previous errors $e(k), e(k-1), \cdots$, which represent the input of the controller.

$$u(k) = u(k-1) + \sum_{i=0}^{n} B_i e(k-i) \tag{A.15}$$

$$= u(k-1) + B_0 e(k) + B_1 e(k-1) + \cdots + B_n e(k-n). \tag{A.16}$$

The Kronecker transformation equivalence property (A.14) will be used to transform the product $B_i e^{(k-i)}$:

$$B_i e(k-i) = \text{vec}\left(B_i e(k-i)\right) = \left(e(k-i)^T \otimes I\right) \text{vec}(B_i). \tag{A.17}$$

It is possible to write the summation that define the controller output:

$$\begin{aligned}\sum_{i=0}^{n} B_i e(k-i) = \left[e(k)^T \otimes I, \cdots, e(k-n)^T \otimes I\right] \text{vec}\big([B_0, \cdots, B_n]\big) \\ = \varphi(k)^T \theta,\end{aligned} \tag{A.18}$$

where:

$$\theta = \text{vec}\big([B_0, \cdots B_n]\big) \tag{A.19}$$

$$\varphi(k) = \left[e(k)^T \otimes I, \cdots, e(k-n)^T \otimes I\right]^T. \tag{A.20}$$

The structure of the B_i matrices can be specialized for a given PID structure and discretisation form. The case of the parallel MIMO PID discretised using Backward Euler's integration will be carried out:

$$U(s) = \left(K_p + K_i \frac{1}{s} + K_d s\right) E(s) \tag{A.21}$$

$$u(k) = \left(K_p + K_i \frac{zT_s}{z-1} + K_d \frac{z-1}{zT_s}\right) e(k). \tag{A.22}$$

After manipulating the above equation, it yields:

$$u(k) = u(k-1) + \left(K_p + K_i T_s + K_d \frac{1}{T_s}\right) e(k) \tag{A.23}$$

$$- \left(K_p + K_d \frac{2}{T_s}\right) e(k-1) \tag{A.24}$$

$$+ K_d \frac{1}{Ts} e(k-2). \tag{A.25}$$

By comparing Equation (A.15) and the above, the structure of B_i can be obtained, as in Equation (A.19). This, in turn, allows the computation of the regressor $\varphi(k)$, used for the MIMO implementation of the VRFT algorithm, as shown in Chapter 3.

$$B_0 = \left(K_p + K_i T_s + K_d \frac{1}{T_s}\right) \tag{A.26}$$

$$B_1 = -\left(K_p + K_d \frac{2}{T_s}\right) \tag{A.27}$$

$$B_2 = K_d \frac{1}{T_s}. \tag{A.28}$$

The obtained structure can be further extended for an unconventional PID configuration, such that the derivative action is based on the output dynamic, instead of the error, and a feed-forward action of the setpoint is present, as seen in Section 2.2.1:

$$u(k) = u(k-1) + \sum_{i=0}^{n} B_i e(k-i) + \sum_{j=0}^{m} M_j y(k-j) + \sum_{p=0}^{q} N_p r(k-p). \tag{A.29}$$

The regressor form of this architecture is now derived:

$$u(k) = \left[K_p + K_i \frac{zT_s}{z-1} \right] e(k) + K_d \frac{z-1}{zT_s} y(k) + K_{ff} r(k). \tag{A.30}$$

An intermediate step is carried out to simplify the notation. The integral contribution term is split by defining a fictitious input $e_F(t)$:

$$u_i(k) = K_i \frac{zT_s}{z-1} e(k) = K_i T_s \frac{z}{z-1} e(k) = K_i T_s e_F(k). \tag{A.31}$$

It follows that:

$$u(k) = K_p e(k) + K_i T_s e_F(k) + K_d \frac{z-1}{zT_s} y(k) + K_{ff} r(k) \tag{A.32}$$

$$u(k+1) = K_p e(k+1) + K_i T_s e_F(k+1) + K_d \frac{1}{T_s} (z-1) y(k) + K_{ff} r(k+1). \tag{A.33}$$

By exploiting the shift invariance and by subtracting the input at the previous tep $u(k-1)$, it yields:

$$\begin{aligned} u(k) = u(k-1) &+ K_p \big[e(k) - e(k-1) \big] + && \text{(A.34)} \\ &+ K_i T_s \big[e_F(k) - e_F(k-1) \big] + && \text{(A.35)} \\ &+ K_d \frac{1}{T_s} \big[y(k) - 2y(k-1) + y(k-2) \big] + && \text{(A.36)} \\ &+ K_{ff} \big[r(k) - r(k-1) \big]. && \text{(A.37)} \end{aligned}$$

This form is convenient as the final regressor form can be manipulated to directly obtain the parameters of interest:

$$\begin{aligned} \varphi(k)^T = \big[& (e(k) - e(k-1))^T \otimes I, && \text{(A.38)} \\ & (e_F(k) - e_F(k-1))^T \otimes I, \\ & \big(y(k) - 2y(k-1) + y(k-2) \big)^T \otimes I, \\ & (r(k) - r(k-1))^T \otimes I \big] \\ \theta = \operatorname{vec} & \big(\big[K_p, K_i T_s, K_d / T_s, K_{ff} \big] \big). && \text{(A.39)} \end{aligned}$$

One final step is to scale back the estimates for K_i and K_d by multiplying or diving for the sample time T_s.