



POLITECNICO DI MILANO
DIPARTIMENTO DI SCIENZE E TECNOLOGIE AEROSPAZIALI
DOCTORAL PROGRAMME IN AEROSPACE ENGINEERING

DATA-DRIVEN CONTROL SYSTEM DESIGN FOR
MULTIROTOR SYSTEMS

Doctoral Dissertation of:
Pietro Panizza

Supervisor:

Prof. Marco Lovera

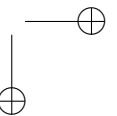
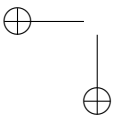
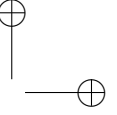
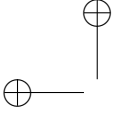
Tutor:

Prof. Carlo Luigi Bottasso

The Chair of the Doctoral Program:

Prof. Luigi Vigevano

2017 – Cycle XXX



Abstract

The definition of flight control laws is one of the most demanding activities in the process of designing remote-controlled multirotor systems. Usually, the controller is synthesised by means of techniques that rely on a prior modelling activity on the real plant, which however frequently neglects some significant dynamic effects. In order to overcome all these limitations, the data-driven control approach emerges as a valuable solution to obtain the specific controller. Since the main feature of these methods is the ability to obtain or tune a controller directly from experimental input-output plant data, they have been proposed to avoid the problem of under-modeling and to facilitate the design of fixed-order controllers. Different linear data-driven methods available in literature have been compared and applied. The Virtual Reference Feedback Tuning (VRFT) method is first adopted because it represents the most promising algorithm. In order to overcome the problem of VRFT with noisy data, the Correlation based Tuning (CbT) method is considered. The lack of an accurate plant model makes it impossible to guarantee the stability of the closed-loop system before implementing and testing the controller on the real system. In order to overcome this critical limitation of data-driven methods, an interesting controller design technique, called Controller Unfalsification, is implemented. Different extensions has been proposed in this thesis to apply the data-driven methods

to tune the controllers of multirotor systems. These extensions consider closed-loop data obtained in flight, cascade control systems and controller architecture based on advanced PID implementation. In this work, all the data-driven algorithms and their extensions are applied to tune the attitude and the position controllers of three multirotor systems. The considered Unmanned Aerial Vehicles (UAVs) are all quadrotors and they differ from size, actuation and control architectures. Exploiting the different features of each UAV and the chance to execute specific tests, different analysis have been performed. The experimental tests have shown that data-driven methods can be successfully applied to tune the considered controllers and they provide performance comparable to model-based methods.

Contents

1	Introduction	1
1.1	Model reference control	5
1.2	Structure of the thesis	7
2	Overview of data-driven tuning methods	9
2.1	Virtual Reference Feedback Tuning	11
2.1.1	Cascade control systems	15
2.2	Correlation based Tuning	18
2.3	Controller unfalsification	21
2.3.1	Problem formulation	22
2.3.2	Optimisation criterion	24
2.3.3	Stability constraint and implementation	26
3	UAV control architectures	31
3.1	Overall control architecture	32
3.2	SISO pitch attitude controller	35
3.3	MISO pitch attitude controller	36
3.4	MISO longitudinal position controller	38
4	Data-driven algorithms for multirotor systems	41

Contents

4.1	Closed Loop Experiments	42
4.1.1	The PBSID algorithm	45
4.2	Controller with feed-forward and action on plant output . . .	50
4.3	Correlation based Tuning for cascade control systems	56
5	Considered multirotor platforms	59
5.1	Aermatica P2-A1	60
5.2	Proto-1	63
5.3	ANT-1	66
6	Results	69
6.1	Aermatica P2-A1	70
6.1.1	VRFT setpoint tracking and load disturbance evaluation	70
6.1.2	CbT and sensitivity to SNR analysis	78
6.2	Proto-1	83
6.2.1	Setpoint tracking evaluation	85
6.2.2	Load disturbance rejection evaluation	87
6.3	Ant-1	88
6.3.1	Pitch attitude controller with open-loop experiments .	89
6.3.2	Pitch attitude controller with closed-loop experiments	112
6.3.3	Pitch-roll attitude controllers with closed-loop exper- iments	119
6.3.4	Position controller	123
7	Conclusions	133
	Glossary	147
	Bibliography	151

CHAPTER *1*

Introduction

ONE of the main objectives of control theory is to design a specific controller that drives the output of a plant to track a defined set-point signal or to satisfy a design target.

In the *model-based* approach a mathematical model of the plant is required in order to obtain the specific controller. Modelling the plant is necessary for this type of methods and it represents one of the most delicate and difficult steps in model-based methods. As described in [42,46], model identification can be adopted to obtain the plant model exploiting measured data from experimental tests on the true system. Different identification techniques can be used to get the model of the plant: in the *black-box* framework the model is obtained directly and solely from the measured input-output data, whereas exploiting *grey-box* algorithms a physically-motivated model is first derived from first principle considerations and then the model

Chapter 1. Introduction

parameters are calibrated with the measured experimental data. However, even if the most advanced identification method is employed, the model always represents an approximation of the real system and some errors are inevitable. Consequently, since the model-based approach is based on the assumption that the plant model represents the true system, these methods are inherently less safe and less robust due to the unmodeled dynamics. *Robust control theory* was born to deal with this kind of problems including additive and multiplicative descriptions and the assumption of bounds on noise or model uncertainties. Furthermore, even if the model is accurate but the assumptions on the system are not correct, the results on the stability and robustness of the closed-loop system are not always valuable.

Since in this work control theory is applied to UAVs, the state of the art of system identification for UAVs must be considered. System identification is now a well established approach for the development of control oriented models in the rotorcraft field (see, *e.g.*, [24, 29, 49] and the references therein). Though the application to full scale rotorcraft is by now fairly mature, less experience has been gathered on small-scale vehicles. In particular, it is apparent from the literature that mathematical models for UAV dynamics are easy to establish as far the kinematics and dynamics of linear and angular motion are concerned, so that a large portion of the available works dealing with UAV control is based on such models. Unfortunately, characterising aerodynamic effects and additional dynamics such as, *e.g.*, due to actuators and sensors, is far from trivial, and has led to an increasing interest in the experimental characterisation of the dynamic response of UAVs.

Usually, trying to characterise difficult effects produces a complex model that cannot be used for controller design. Indeed, a model with a high order or a high level of nonlinearity leads to controllers with high order and high nonlinearity. Thus, a controller reduction procedure is inevitable since controllers that are too complex could be difficult or costly to design, use and maintain. This step is generally problematic since any stability guarantees that were formulated for the full-order controller may not transfer to the reduced-order controller. Furthermore, whilst the optimality of the

full-order controller can be guaranteed, that is not the case for the reduced-controller.

In many applications the structure of the controller is predetermined. Many industrial processes, for example, use predefined PID controllers and the procedure is limited to tuning the PID gains. Tuning only the controller gains starting from a full-order controller is far from trivial. For this reason, the full-order controller cannot be employed and structured model-based control techniques have been developed.

Typical linear control system design methodologies that belong to the model-based approaches are: LQR design [8] and robust control [44]. For non linear system the most employed methods are: Lyapunov-based design methodologies, backstepping method [9] and dynamic inversion [16].

In order to overcome all these limitations, the *data-driven* control approach emerges as a valuable solution to obtain the specific controller. Since the main feature of these methods is the ability to obtain or tune a controller directly from experimental input-output plant data, they have been proposed to avoid the problem of under-modeling and to facilitate the design of fixed-order controllers. Data-driven algorithms skip the modelling phase almost entirely and instead reformulate the controller identification procedure as a parameter optimisation problem in which the optimisation is carried out directly on the controller parameters. Furthermore, the achieved performance of the controllers is not linked to the techniques used to model the plant or the order of the identified plant model. It emerges that the main difference between model-based and data-driven approaches is whether the plant model is involved in the controller design. From this point of view, the data-driven class includes also methods that are not strictly related to the control community such as: neural network based control methods or fuzzy control methods (see [27]). Several data-driven control design algorithms have been proposed recently. Compared to the work in [1] which was focused on PID controllers and exploited simple empirical rules, the new data-driven methods are based on a rigorous mathematical analysis and under, certain reasonable assumptions, they can guarantee also the stability of the closed-loop system. The data-driven algorithms considered

Chapter 1. Introduction

in this work are also computational efficient: this allows a fast re-tuning of the controller when the plant performance is reduced (*e.g.*, components ageing) or when the operating conditions change (*e.g.*, different payload or environment).

As other control strategies, data-driven methods are not omnipotent. Certain assumptions must be made before applying these algorithms and, considering the data-driven methods employed in this work, some of these assumptions involve the system to control (such as, *e.g.*, achievable closed-loop bandwidth, dominant dynamics, presence of time-delays). Without this information, obtaining a satisfactory tuning can be challenging, as the choice of an unattainable closed-loop reference model can lead to poor performance (not unlike erroneous structure selection in model identification problems). The reader must not be surprised by this statement. Indeed, the amount of required plant information is less than in the model-based framework and, as will be explained in the next section and in the following chapters, this information is usually available from the plant manufacturer or can be obtained with simple open-loop or closed-loop tests. Furthermore, some new definitions must be coined for these methods such as *robustness*. Indeed, since these algorithms do not involve directly the plant model and neither the unmodeled dynamics, the traditional definition of robustness is no longer valid.

At this point where the data-driven framework is introduced, the reader could ask if data-driven methods perform better than model-based methods. Recalling the results in [21], if the evaluation criterion is the variance of the controller parameters then the model-based approach achieves better results since it has been shown that an approach based on two optimisation steps is statistically efficient (see again [21] and the references therein). Nevertheless, the previous criterion represents only an intermediate step toward the evaluation of the methods. As it will be explained in the next section, where *model reference control* and the cost criterion are presented, if the control cost achieved by the designed controller is taken into account, the following considerations are valid:

- if the model structure is perfectly known and the model order is low,

1.1. Model reference control

the model-based approach is theoretically always the best approach.

- If the model structure is not completely known and/or a high order model is identified, the data-driven approach can statistically outperform the model-based solution in terms of the control cost, even if the variance of the parameters remains larger.
- Because in the real world the model structure is never perfectly known and under-modeling can not be avoided with a low-order model, the data-driven approach may give better results in many real applications.

The previous considerations are the conclusion of [21] and they are here reported for the sake of clarity. Furthermore, in order to achieve a statistically efficient estimate, the model-based approach requires both the system and the noise model to be correctly parameterized. Finally, the data-driven approach requires a convex optimisation problem if the controller is linearly parameterized whereas the model-based approach requires that both the controller and the plant model are linearly parameterized.

The applicability of data-driven methods to the tuning of control laws for multirotor UAVs has been verified with reference to three different platforms which cover a wide range of Take-Off Weights (TOWs) corresponding to:

- a *large* platform (5 kg TOW), representative of a multirotor for professional industrial applications (see, *e.g.*, the DJI Matrice 200 platform);
- a *medium* platform (1.5 kg TOW), representative of a multirotor for recreational personal use (see, *e.g.*, the DJI Phantom 4 platform);
- a *small* platform (200 g TOW), representative of the class of "harmless" multirotors according to the Federal Aviation Authority (< 250 grams, see [2]) or the Ente Nazionale per l'Aviazione Civile - ENAC (< 300 grams, see [18]).

1.1 Model reference control

Usually, the requirements on the closed-loop behaviour of the system are expressed as simple conditions on *e.g.*, the bandwidth of the closed-loop

Chapter 1. Introduction

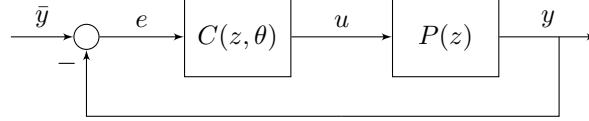


Figure 1.1: *The control system.*

system or its disturbance rejection properties. In addition some robustness requirements may be considered such as requiring a certain gain and phase margin.

The *model reference* approach represents a different way to define the design target. It differs from traditional methods in how the requirements for the controller are specified: instead of providing explicit limits on overshoot, bandwidth or response time, the requirements are provided in the form of a reference model for the closed-loop behaviour of the system. The objective is to design a controller such that the difference between the reference model and the actual closed-loop behaviour of the system is as small as possible.

Consider the closed-loop system shown in Figure 1.1 with the *unknown* stable linear Single Input Single Output (SISO) plant $P(z)$ and the controller $C(z; \theta)$ where θ is the n -dimension vector of controller parameters. The objective of minimising the difference between the reference model and actual closed-loop transfer functions is formulated with the control cost criterion in the following:

$$J_{MR}(\theta) = \left\| \left(\frac{P(z)C(z, \theta)}{1 + P(z)C(z, \theta)} - M(z) \right) W(z) \right\|_2^2 \quad (1.1)$$

where $M(z)$ is the closed-loop reference model and $W(z)$ is a weighting function chosen by the user to focus the model matching problem in the desired frequency range. The optimal controller that minimises the control cost in (1.1) exists and is given by:

$$\bar{C}(z) = \frac{M(z)}{P(z)(1 - M(z))}. \quad (1.2)$$

1.2. Structure of the thesis

In order to obtain the optimal controller in (1.2), two approaches can be adopted:

The model-based approach assumes that a detailed and reliable model of the plant $P(z)$ is available in order to directly compute the ideal controller.

The data-driven approach attempts to minimise the control objective in (1.1) by solving a parameter optimisation problem without first estimating a model of the plant.

The model-based approach solves the model reference control problem assuming that a model of the plant is available. As explained in the first part of this chapter, this model may be derived exploiting different identification techniques but unmodeled dynamics always exist. Note that the choice of a high order model is not the solution. Indeed, the order of the ideal controller in (1.2) depends on the order of the plant model.

As declared before, even if data-driven methods do not require accurate knowledge of the process, at least some prior information on the plant is required. Considering the model reference control problem, this information is employed to select a proper reference models $M(z)$.

1.2 Structure of the thesis

This thesis is organised as follows:

- Chapter 2 provides a general overview of data-driven methods. A first classification of these methods is presented and the most promising algorithms are illustrated in detail.
- In Chapter 3 the control architecture to tune is presented. In this chapter the reader understands how a multicopter UAV works and which are the control architectures. For each control architecture the unknown controller parameters to tune are described.
- Chapter 4 provides all the extensions of the methods presented in Chapter 2 to deal with the control problem illustrated in Chapter 3.

Chapter 1. Introduction

In particular, the main extension involves the possibility to tune the controller with experimental data obtained with flight tests.

- In Chapter 5 all the multirotor platforms employed in this work are presented in detail. They are all quadrotors and they differ from size, actuation architecture and control structure.
- All the results are displayed in Chapter 6 where different analysis are illustrated, exploiting the different features of each UAV and the possibility to execute specific tests.
- In Chapter 7 some conclusions on the results presented in this work are drawn.

CHAPTER 2

Overview of data-driven tuning methods

SINCE data-driven methods are still in their infancy, different names are used in the literature to describe this kind of algorithms: data-driven, data-based, model-free,... The term *data-driven* was first proposed in computer science but recently it entered also the vocabulary of several researchers in the control community. As was presented in Chapter 1, these methods were born to overcome the limitations of model-based methods and this goal can be accomplished in different ways.

The first classification of the data-driven methods considers the structure of the controller: some data-driven algorithms are able to tune only the unknown parameters of a fixed-structured controller, others involve the plant model structure implicitly and lead to a controller structure that *a-priori* is unfixed (see [27]). The first class of data-driven methods is considered in this work since the main goal is to tune controllers with a fixed struc-

Chapter 2. Overview of data-driven tuning methods

ture. Indeed, in most practical cases, changing the controller structure is not feasible (*e.g.*, the controller source code is not available). Furthermore, in all the UAVs considered in this work, the controllers were already tuned with model-based or manual methods and the new data-driven tuning will be compared with the existing tuning without changing the controller architecture. The determination of the control structure goes beyond the scope of this work and the controller architecture was already implemented on all the considered UAVs.

The data-driven algorithms can be classified also on how they obtain the optimal tuning. Some methods employ an *iterative* procedure. The Iterative Feedback Tuning (IFT) method, that was first proposed in [25, 26] belongs to this class and it was considered at the beginning of this work. It involves an iterative optimisation of the parameters of the structured controller according to an estimated gradient of a control performance criterion. It is comparatively slow and requires several experiments on the plant at each iteration. Moreover it can only guarantee that the result is close to the local minimum of the cost function. The IFT method was not considered in this thesis due to these drawbacks and because the required experiments cannot be performed on the first UAV that is employed in this work.

Beside iterative algorithms, the class of *non-iterative* methods proposes a more attractive perspective to tune the controller parameters. Instead of performing multiple experiment on the process, the non-iterative methods are computationally efficient: they can be called also *one-shot* algorithms in the sense that a single batch of experimental data is used to solve the optimisation problem. Thus they allow also a fast re-tuning of the controller when the plant performance is reduced (*e.g.*, components ageing) or operating conditions change (*e.g.*, different payloads or environment). VRFT and CbT belong to this class and they are presented in detail in the next sections. Furthermore, in [19], VRFT was already extended to tune a cascade control system with data from a single experiment and this makes VRFT the best candidate to solve the tuning problem since all UAVs exploit cascade control architectures, as will be illustrated in Chapter 3.

Since the data-driven methods do not explicitly involve the plant model,

2.1. Virtual Reference Feedback Tuning

it is far from straightforward to ensure stability constraints on the closed-loop system. Indeed, the VRFT method considers only the performance of the closed-loop system, minimising the discrepancy between the desired and the actual input-output behaviour. CbT, as presented in [52], allows to consider also the stability constraint exploiting the small-gain theorem. The stability condition in the CbT algorithm involves the discrepancy between the actual controller and a stabilising controller previously defined (see again [52] for more details). Usually this controller is used to collect the data in a closed-loop experiment. With stable but non-minimum phase plant, this approach provides only a refinement of the already available controller and the initial stabilising controller must be known. To overcome this limitation, a recent method called *controller unfalsification* has been proposed (see [3]). It is a non-iterative data-driven control design approach that incorporates stability tests originally introduced for the unfalsified control framework (see again [3]). This method is applied in this work to show its capability to deal with a real tuning problem and to compare it with the other data-driven methods.

2.1 Virtual Reference Feedback Tuning

In this section the VRFT is introduced. This method was presented in [10, 23] and, as the method name suggests, it exploits the idea of a *virtual reference* signal. The key concept underlying VRFT is that if the input and the output of the controller are known then the model matching problem in (1.1) can be reformulated as a parameter identification problem on the controller.

The main features of VRFT are that the model-reference problem (1.1) is solved without any knowledge of the system and using only a set of available open-loop measurements $D_N = \{u(t), y(t)\}_{t=1..N}$, where N is the length of the dataset. The only requirement for this experiment is that it must excite the system over the entirety of the frequency range of interest.

Consider the reference signal $r(t)$ that would feed the system in closed-loop operation when the closed-loop model is $M(z)$ and the output is the measured $y(t)$. Such a signal is called *virtual reference* because it is not

Chapter 2. Overview of data-driven tuning methods

used to generate $y(t)$ and it can be computed from the output data as

$$r(t) = M(z)^{-1}y(t).$$

The signal $r(t)$ can be computed offline and it represents the setpoint that generates the output $y(t)$ when the closed-loop is optimal, that is the closed-loop transfer function is $M(z)$. Starting from the signal $r(t)$, the input of the controller can be computed as

$$e(t) = r(t) - y(t).$$

Since the input and the output of the controller are now known, the model matching problem can be considered as an identification problem: a good controller (making the closed-loop as close as possible to $M(z)$) is then the one that produces the input sequence of the experiment $u(t)$ when it is fed by the error signal $e(t)$. The information about the reference closed-loop model is embedded in the input signal $e(t)$.

Formally, the cost criterion minimised by the VRFT algorithm is the following:

$$J_{VR}^N(\theta) = \frac{1}{N} \sum_{t=1}^N (u_L(t) - C(z, \theta)e_L(t))^2, \quad (2.1)$$

where $u_L(t)$ and $e_L(t)$ are suitably filtered versions of $u(t)$ and $e(t)$ such that the cost function (2.1) is a local approximation of the criterion (1.1) in the neighbourhood of the minimum point. As explained in [10], the optimal choice of the filter $L(z)$ is

$$|L|^2 = |1 - M|^2 |M|^2 |W|^2 \frac{1}{\Phi_u}, \quad (2.2)$$

where Φ_u is the spectral density of $u(t)$. $L(z)$ is not needed if the considered controller class contains the optimal controller which exactly solves the

2.1. Virtual Reference Feedback Tuning

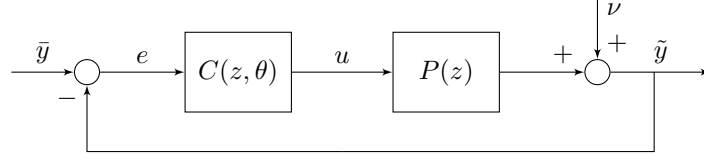


Figure 2.1: The control system with measurement noise.

model matching problem (this controller is defined in (1.2)).

Note that if the user considers a linearly parameterised controller class $\mathcal{C}(\theta) = \{C(z, \theta) = \beta^T(z)\theta, \theta \in R^n\}$, the criterion in (2.1) can be rewritten as

$$J_{VR}^N(\theta) = \frac{1}{N} \sum_{t=1}^N (u_L(t) - \varphi_L^T(t)\theta)^2, \quad (2.3)$$

where $\varphi_L(t) = \beta(z)e_L(t)$ and the optimal parameters are:

$$\hat{\theta}_N = \arg \min_{\theta} J_{VR}^N(\theta). \quad (2.4)$$

The closed form solution of the problem in (2.4) exists and it is:

$$\hat{\theta}_N = \left[\sum_{t=1}^N \varphi_L(t)\varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \varphi_L(t)u_L(t). \quad (2.5)$$

Obviously the input of the plant has to be a persistently exciting signal in order to apply (2.5) as described in [36].

Since VRFT exploits a Prediction Error Method (PEM) identification procedure to tune the controller, it has to deal with the problem related to these class of methods. In particular, suppose that the output of the plant is affected by an additive noise $\nu(t)$ (Figure 2.1):

$$\tilde{y}(t) = P(z)u(t) + \nu(t),$$

with the assumption that $u(t)$ and $\nu(t)$ are uncorrelated. In this case the

Chapter 2. Overview of data-driven tuning methods

PEM procedure is not adequate for this problem because the input of the controller is affected by the noise $\nu(t)$ and this results in a biased parameter vector estimate. As described in [10], an instrumental variable method can be employed to counteract the effect of noise. The instrumental variable can be built in different ways and it must be correlated with the regression variable and uncorrelated with the noise $\nu(t)$. To satisfy these requirements, the instrumental variable can be chosen in two ways:

Repeated experiments. Assuming that different realisations of the noise affect different experiments, the user has to perform a second experiment with the same input $u(t)$ obtaining a new noisy output signal $\tilde{y}'(t)$. Building the instrumental variable as

$$\zeta(t) = \beta(z)L(z) (M(z)^{-1} - 1) \tilde{y}'(t),$$

VRFT leads to the same results as in the noiseless case.

Plant identification. In some situations a second experiment with the same input signal can not be performed. Thus, a way to build the instrumental variable passes through the identification of the plant in order to get a model $\hat{P}(z)$. The model can be exploited to build the noiseless output as:

$$\hat{y}(t) = \hat{P}(z)u(t)$$

and the instrumental variable is

$$\zeta(t) = \beta(z)L(z) (M(z)^{-1} - 1) \hat{y}(t).$$

This approach guarantees a consistent estimate but its variance depends on the quality of the model $\hat{P}(z)$. Furthermore, the plant identification procedure clashes with the data-driven idea of the VRFT method. Nevertheless the reader should notice that $\hat{P}(z)$ is not directly involved in the design of the controller but it is employed only

2.1. Virtual Reference Feedback Tuning

Algorithm 1 The VRFT algorithm.

- 1: Compute $U(z)$ such that $|U(e^{j\omega})|^2 = \Phi_u(\omega)$.
 - 2: Compute $L(z) = (1 - M(z))M(z)W(z)U^{-1}(z)$.
 - 3: Compute $u_L(t)$ as $u_L(t) = L(z)u(t)$.
 - 4: Compute $\varphi(t) = \beta(z)L(z)(M(z)^{-1} - 1)\hat{y}(t)$.
 - 5: Identify the plant model $\hat{P}(z)$.
 - 6: Compute $\hat{y}(t) = \hat{P}(z)u(t)$.
 - 7: Compute the instrumental variable $\zeta(t) = \beta(z)L(z)(M(z)^{-1} - 1)\hat{y}(t)$.
 - 8: Compute $\hat{\theta}_N^{IV} = \left[\sum_{t=1}^N \zeta(t)\varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \zeta(t)u_L(t)$.
-

in the creation of the instrumental variable.

When the instrumental variable is selected, it can be used to solve the problem in (2.3) and the optimal solution is:

$$\hat{\theta}_N^{IV} = \left[\sum_{t=1}^N \zeta(t)\varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \zeta(t)u_L(t). \quad (2.6)$$

The VRFT procedure is summarised in Algorithm 1.

Originally in [10], the VRFT method was presented to design a SISO control system. The reader should notice that Algorithm 1 does not limit this data-driven method to SISO control system. Indeed, in [41] VRFT is applied to a Multiple Input Multiple Output (MIMO) process control system following the same steps defined in Algorithm 1. Although, in this work, VRFT will be mainly applied to SISO controllers, an application to a MIMO control system is presented in Section 6.3.3.

2.1.1 Cascade control systems

It has been shown in [19] that the VRFT rationale can be extended to multiple nested loops, by still relying on a single experiment.

Consider the cascade control scheme in Figure 2.2 where only two loops are shown without loss of generality and a noiseless environment is considered (see [19] for a deeper presentation of the methodology). Given two reference models $M_i(z)$ and $M_o(z)$, for the inner loop and the outer loop respectively, consider two families of linear proper controllers $\mathcal{C}_i(\theta_i) =$

Chapter 2. Overview of data-driven tuning methods

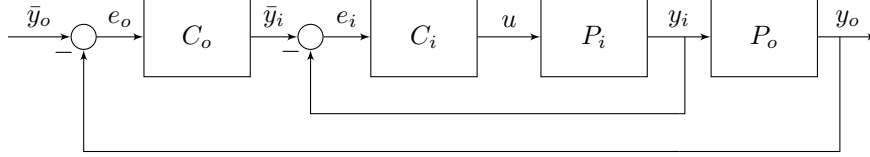


Figure 2.2: Cascade control scheme with two nested loops.

$\{C_i(z, \theta_i), \theta_i \in R_i^n\}$ and $C_o(\theta_o) = \{C_o(z, \theta_o), \theta_o \in R_o^n\}$ and the set of data $D_N = \{u(t), y_i(t), y_o(t)\}_{t=1, \dots, N}$ being $u(t)$ the control variable, $y_i(t)$ the output of the inner loop, $y_o(t)$ the output of the outer loop. The inner controller can be tuned by applying the standard VRFT as presented in Section 2.1. For the outer controller, on the other hand, the approach needs to be different, as the input of the system to control is the reference $\bar{y}_i(t)$ (see again Figure 2.2), that is not available in the dataset, since measurements are collected during an open-loop experiment.

Nevertheless, in [19] it has been shown that the reference signal $\bar{y}_i(t)$ can be derived from the available data by exploiting the fact that the inner controller is designed independently of the outer one. In detail, once $C_i(z, \theta_i)$ is fixed, the input of the inner loop can be calculated as

$$\bar{y}_i(t) = e_i(t) + y_i(t),$$

where the tracking error comes from the result of the inner design as

$$e_i(t) = C_i^{-1}(z, \hat{\theta}_i)u(t),$$

where $\hat{\theta}_i$ are the optimal parameters of the inner loop. With such a choice, $\bar{y}_i(t)$ is exactly the signal that would feed the inner loop in closed-loop working conditions when the output is $y_i(t)$. Then, the outer controller can be easily found as result of VRFT synthesis, by using the set of I/O data $D_N^o = \{\bar{y}_i(t), y_o(t)\}_{t=1, \dots, N}$. More specifically, θ_o comes as the minimizer

2.1. Virtual Reference Feedback Tuning

Algorithm 2 The VRFT method for two nested cascade control loops with a single set of experimental data.

- 1: Compute $U(z)$ such that $|U(e^{j\omega})|^2 = \Phi_u(\omega)$.
- 2: **repeat**
- 3: Compute $L_i(z) = (1 - M_i(z)) M_i(z) W_i(z) U^{-1}(z)$.
- 4: Compute $u_L(t)$ as $u_L(t) = L_i(z) u(t)$.
- 5: Compute $e_{iL}(t)$ as $e_{iL}(t) = L_i(z) (M_i^{-1}(z) - 1) y_i(t)$.
- 6: Compute $\hat{\theta}_i = \arg \min_{\theta_i} \frac{1}{N} \sum_{t=1}^N (u_L(t) - C_i(z, \theta_i) e_{iL}(t))^2$.
- 7: **until** $C_i(z, \hat{\theta}_i)$ is a minimum phase system, otherwise change $M_i(z)$.
- 8: Compute $\bar{y}_i(t) = C_i^{-1}(z, \hat{\theta}_i) u(t) + y_i(t)$.
- 9: Compute $U_o(z)$ such that $|U_o(e^{j\omega})|^2 = \Phi_{\bar{y}_i}(\omega)$ where $\Phi_{\bar{y}_i}(\omega)$ is the spectral density of $\bar{y}_i(t)$.
- 10: Compute $L_o(z) = (1 - M_o(z)) M_o(z) W_o(z) U_o^{-1}(z)$.
- 11: Compute $\bar{y}_{iL}(t)$ as $\bar{y}_{iL}(t) = L_o(z) \bar{y}_i(t)$.
- 12: Compute $e_{oL}(t)$ as $e_{oL}(t) = L_o(z) (M_o^{-1}(z) - 1) y_o(t)$.
- 13: Compute $\hat{\theta}_o = \arg \min_{\theta_o} \frac{1}{N} \sum_{t=1}^N (\bar{y}_{iL}(t) - C_o(z, \theta_o) e_{oL}(t))^2$.

of

$$J_{VR}(\theta_o) = \frac{1}{N} \sum_{t=1}^N (\bar{y}_{iL}(t) - C_o(z, \theta_o) e_{oL}(t))^2 \quad (2.7)$$

where $\bar{y}_{iL}(t)$ and $e_{oL}(t)$ are suitably filtered versions of $\bar{y}_i(t)$ and $e_o(t)$, the latter being the virtual error of the outer loop:

$$e_o(t) = (M_o^{-1}(z) - 1) y_o(t).$$

The optimal filters for the inner and outer loop are discussed in [19], following the rationale of [10].

The VRFT method for two nested cascade control loops with a single set of experimental data is summarised in Algorithm 2.

Chapter 2. Overview of data-driven tuning methods

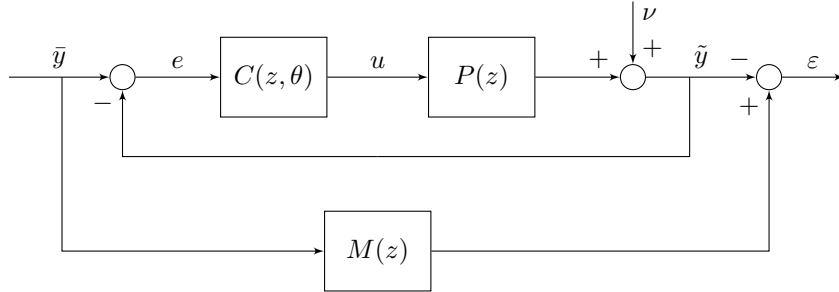


Figure 2.3: *The tuning scheme for the CbT method.*

2.2 Correlation based Tuning

As explained in Section 2.1, VRFT leads to a specific identification problem and when experimental data is affected by noise, the PEM procedure is not consistent to solve this problem. In order to overcome this limitation the CbT method is considered in this work (see [30, 51, 52]). It is a non-iterative controller tuning algorithm that employs the correlation approach to deal with noisy data. As in the VRFT method, the performance specification is provided in terms of a closed-loop reference model and VRFT and CbT share the same cost criterion to minimise as defined in (1.1). Furthermore, CbT incorporates a stability constraint for the closed-loop system: it is implemented as a set of convex constraints leading the minimisation problem in (1.1) to a constrained optimisation problem. CbT is also able to deal with nonminimum-phase or unstable plants (see again [52]).

In the CbT approach, the optimal controller is computed exploiting the error $\varepsilon(t, \theta)$ as depicted in Figure 2.3. Indeed, the minimisation of (1.1) is equivalent to minimisation of the norm of the system given $\varepsilon(t)$ as an output when it is fed by a flat-spectrum input $\bar{y}(t)$.

For a linearly parameterised controller, an approximation of the cost criterion defined in (1.1) can be made in order to make it convex. In particular the approximation considers the sensitivity function: the ideal sensitivity function is given by

$$\bar{S}(z) = \frac{1}{1 + \bar{C}(z)P(z)} = 1 - M(z)$$

2.2. Correlation based Tuning

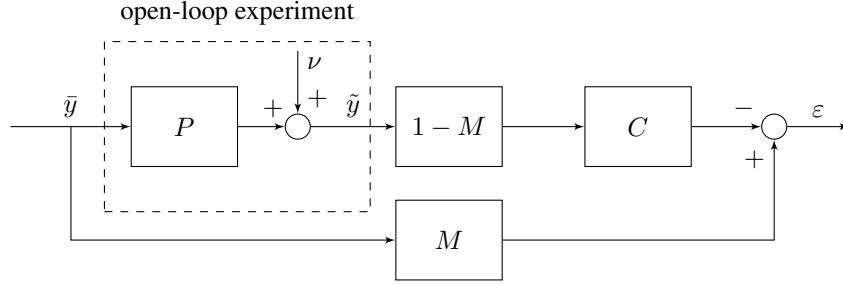


Figure 2.4: The approximate tuning scheme for CbT method.

where $\bar{C}(z)$ is the optimal controller that realises $M(z)$ as defined in (1.2). Assuming that the actual sensitivity function is equal to the ideal sensitivity function, that is

$$S(z, \theta) = \frac{1}{1 + C(z, \theta)P(z)} = \bar{S}(z),$$

the model matching problem in (1.1) becomes convex and the tuning scheme for the CbT method can be redrawn as in Figure 2.4 (see [52] for a discussion on the approximation). Note that the same approximation is also exploited in the VRFT algorithm to solve a convex optimisation problem (see [10]).

Considering the Figure 2.4 and the data $D_N = \{\bar{y}(t), \tilde{y}(t)\}_{t=1..N}$ from an open-loop test, where N is the length of the dataset. The error $\varepsilon(t, \theta)$ depends on the exogenous signals $\bar{y}(t)$ and $\nu(t)$:

$$\begin{aligned} \varepsilon(t, \theta) &= M(z)\bar{y}(t) - C(z, \theta)(1 - M(z))\tilde{y}(t) \\ &= (M(z) - C(z, \theta)(1 - M(z))G(z))\bar{y}(t) \\ &\quad - C(z, \theta)(1 - M(z))\nu(t). \end{aligned} \quad (2.8)$$

If the optimal controller defined in (1.2) is considered, the error in (2.8) becomes:

$$\varepsilon(t, \theta) = -\bar{C}(z)(1 - M(z))\nu(t) \quad (2.9)$$

Chapter 2. Overview of data-driven tuning methods

and since $\nu(t)$ is not correlated with the reference $\bar{y}(t)$, the error computed with the ideal controller is not correlated with $\bar{y}(t)$. Thus, the goal is to find the optimal controller parameter θ such that the error $\varepsilon(t, \theta)$ is uncorrelated with $\bar{y}(t)$.

The implementation of the CbT method is now briefly illustrated (see [52] for more details). To decorrelate $\varepsilon(t, \theta)$ and $\bar{y}(t)$, an extended instrumental variable $\varsigma(t)$ correlated only with $\bar{y}(t)$ is introduced:

$$\varsigma(t) = \left[\bar{y}_F(t+l) \quad \dots \quad \bar{y}_F(t) \quad \dots \quad \bar{y}_F(t-l) \right]^T \quad (2.10)$$

where l is a sufficiently large integer and

$$\bar{y}_F(t) = F(z)\bar{y}(t).$$

The filter $F(z)$ assumes here the same role of the filter $L(z)$ in the VRFT method and the optimal choice of $F(z)$ is such that

$$|F|^2 = |1 - M|^2 |W|^2 \frac{1}{\Phi_{\bar{y}}}$$

where $\Phi_{\bar{y}}$ is the spectral density of the reference signal $\bar{y}(t)$. See [52] for discussion on the optimal filter. The correlation function is defined as

$$f_{N,l}(\theta) = \frac{1}{N} \sum_{t=1}^N \varsigma(t) \varepsilon(t, \theta)$$

and the correlation criterion to minimise is

$$J_{N,l}(\theta) = f_{N,l}^T(\theta) f_{N,l}(\theta).$$

Thus, the optimal controller parameters are

$$\hat{\theta}_N = \arg \min_{\theta} J_{N,l}(\theta). \quad (2.11)$$

2.3. Controller unfalsification

Algorithm 3 The CbT algorithm.

- 1: Compute $\bar{Y}(z)$ such that $|\bar{Y}(e^{j\omega})|^2 = \Phi_{\bar{y}}(\omega)$.
 - 2: Compute $F(z) = (1 - M(z))W(z)\bar{Y}^{-1}(z)$.
 - 3: Compute $\bar{y}_F(t)$ as $\bar{y}_F(t) = F(z)\bar{y}(t)$.
 - 4: Choose l close to the impulse response of $M(z)$.
 - 5: Compute $\varsigma(t) = [\bar{y}_F(t+l) \ \dots \ \bar{y}_F(t) \ \dots \ \bar{y}_F(t-l)]^T$.
 - 6: Compute the error $\varepsilon(t, \theta) = M(z)\bar{y}(t) - C(z, \theta)(1 - M(z))\bar{y}(t)$.
 - 7: Compute $f_{N,l}(\theta) = \frac{1}{N} \sum_{t=1}^N \varsigma(t)\varepsilon(t, \theta)$.
 - 8: Compute $J_{N,l}(\theta) = f_{N,l}^T(\theta)f_{N,l}(\theta)$.
 - 9: Compute $\hat{\theta}_N = \arg \min_{\theta} J_{N,l}(\theta)$.
-

The optimal parameters in (2.11) asymptotically converge to the optimiser of (1.1) (the proof is provided in [52]).

The choice of the parameter l in (2.10) is now discussed. This parameter represents a trade-off between accuracy and bias: l must be large enough to minimise (1.1) using (2.11) but the bias due to noise increases with l (see [52]). As proposed in [21], selecting l close to the impulse response of the closed-loop reference model $M(z)$ represents a good trade-off. Finally, if the Signal to Noise Ratio (SNR) in the experimental data is high, different choices of l can be made leading to better performance as will be illustrated in Section 6.1. In particular, in this specific case, increasing the value of l leads to a better minimisation of (1.1) and the bias due to noise is negligible due to the high SNR.

The CbT method is summarised in Algorithm 3.

2.3 Controller unfalsification

As already discussed, data-driven methods are based on the assumption that the model of the plant is not known. Typically, they only require the identification of basic plant properties, generally inadequate for control design purposes but essential to define an achievable closed-loop dynamic model. The lack of an accurate plant model makes it impossible to guarantee the stability of the closed-loop system before implementing and testing the controller on the real plant. In order to overcome this critical limitation of data-driven methods, an interesting controller design technique, called

Chapter 2. Overview of data-driven tuning methods

controller unfalsification, has been presented in [3]. The proposed method is based on the unfalsified control theory. As suggested in [3], by exploiting the concept of fictitious reference, one can define cost functions in terms of discrepancy between desired input-output behaviour which allows the derivation of a data-driven controller tuning procedure that includes an effective stability constraint.

2.3.1 Problem formulation

As in many other data-driven methods, the goal is the tuning of a controller parameter vector θ based on an available dataset, properly gathered by means of an experimental test. The controller unfalsification method theoretically guarantees closed-loop stability, as well as providing adequate output performance. According to the method proposed in [3], such a result can be obtained by forcing input and complementary sensitivity functions to be as close as possible to *a-priori* defined reference models (respectively $Q(z)$ and $M(z)$). By minimising the discrepancy between the complementary sensitivity function and the reference model $M(z)$, one forces the output behaviour to approximate the desired one. However, there is no guarantee that the obtained controller actually stabilises the plant, even if the plant is stable and non-minimum phase. Indeed, in such cases, instability can occur if the reference model is not achievable or the dataset is very limited or strongly affected by noise. For this reason, it is necessary to define a second reference model $Q(z)$, which is the desired control sensitivity function. As it will be shown below, one can ensure internal stability of the closed-loop by minimising the discrepancy between control sensitivity and a properly defined $Q(z)$. Such controller tuning procedure leads to a multi-objective minimisation problem and it is convenient to reformulate the problem exploiting the concept of *Pareto optimal solution*, as suggested in [3]. This allows one to define a single objective problem based on the minimisation of a cost function of the form

$$J(\theta) = (1 - \delta)J_n(\theta) + \delta J_v(\theta), \quad (2.12)$$

2.3. Controller unfalsification

where J_n and J_v are related to input and output discrepancies respectively, while $\delta \in [0, 1]$ is a weighting factor which establishes a trade-off between closed-loop stability and output performance.

Controller structure

The unfalsified control, as well as VRFT and CbT approaches, requires that the controller structure is defined *a priori* by the designer. A smart choice is to consider a parametric controller family of the form

$$C(z, \theta) = \frac{\bar{N}(z, \theta) N^*(z)}{\bar{D}(z, \theta) D^*(z)}, \quad (2.13)$$

where $D^*(z)$ and $N^*(z)$ are fixed polynomials with all unstable roots. Typically, $D^*(z)$ is associated with the presence of the controller integral term. Instead, $\bar{N}(z, \theta)$ and $\bar{D}(z, \theta)$ are polynomials whose coefficients form the unknown parameter vector θ , $\bar{N}(z, \theta) = n_0 + n_1 z + \dots + n_i z^i$ and $\bar{D}(z, \theta) = 1 + d_1 z + \dots + d_j z^j$, that is, $\theta = [n_0 \ n_1 \ \dots \ n_i \ d_1 \ \dots \ d_j]$.

Reference models

Reference models $Q(z)$ and $M(z)$ have to be consistent with the prescribed controller family in order to achieve a well-defined design problem. As suggested in [3], the following conditions should be met:

- $Q(z)$ factorised as $Q(z) = N^*(z)\bar{Q}(z)$ with $\bar{Q}(z)$ stable and minimum phase.
- Roots of $N^*(z)$ are zeros of $M(z)$.
- Roots of $D^*(z)$ are zeros of $1 - M(z)$.

Although the definition of the closed-loop reference model $M(z)$ required few information (see Chapter 1), the choice of the desired control sensitivity function $Q(z)$ is not trivial. Ideally, it should be defined as $Q(z) = M(z)/P(z)$ to guarantee the transfer functions to be consistent

Chapter 2. Overview of data-driven tuning methods

with each other, but due to the lack of an accurate plant model, this relationship is only imposed in static conditions, *i.e.*, $Q(1) = M(1)/\hat{P}(1) = 1/\hat{P}(1)$, so that

$$Q(z) = \frac{1}{\hat{P}(1)} \frac{N_Q(z)}{D_Q(z)}. \quad (2.14)$$

Regarding the choice of poles and zeros of $Q(z)$, one must rely on the estimated transfer function of the plant to roughly identify its frequency response. More specifically, focus should be placed on the frequency intervals where the magnitude of the plant is higher, so that $Q(z)\hat{P}(z) \cong 1$ within the desired bandwidth and $\ll 1$ beyond this frequency.

2.3.2 Optimisation criterion

In the following subsection, the optimisation criterion will be explained in detail.

First, it is essential to introduce the notion of fictitious reference r_θ , which consists of defining the setpoint signal by inverting the controller transfer function as follows:

$$r_\theta(t) = C^{-1}(z, \theta)u(t) + \tilde{y}(t), \quad t = 0, 1, \dots, N. \quad (2.15)$$

Care should be taken not to confuse the virtual reference concept introduced by the VRFT approach with the aforementioned fictitious reference. As a matter of fact, the VRFT virtual reference is obtained by inverting the reference model $M(z)$ rather than the potential controller $C(z, \theta)$. Therefore, the approach outlined here is much more similar to the alternative version of the VRFT suggested in [48], which is based on the inverse controller identification.

The fictitious reference $r_\theta(t)$ can be adopted as the input of the reference

2.3. Controller unfalsification

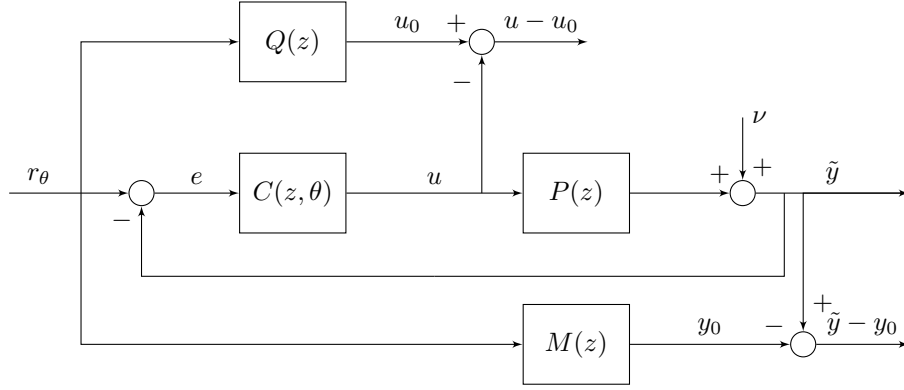


Figure 2.5: The tuning scheme for the controller unfalsification method.

models to obtain the ideal control variable and the ideal output:

$$u_0(t) = Q(z)r_\theta(t) \quad (2.16)$$

$$y_0(t) = M(z)r_\theta(t). \quad (2.17)$$

Basically, the optimisation criterion consists in selecting the parameter vector θ to minimise the difference between the set of measured input-output experimental data $\{u(t), \tilde{y}(t)\}$ and the signals $u_0(t)$ and $y_0(t)$, as schematically shown in Figure 2.5. By replacing the definition of fictitious reference $r_\theta(t)$ in equations (2.16) and (2.17), the input-output discrepancies can be written as

$$u(t) - u_0(t) = u(t) - Q(z)\tilde{y}(t) - C^{-1}(z, \theta)Q(z)u(t) \quad (2.18)$$

$$\tilde{y}(t) - y_0(t) = (1 - M(z))\tilde{y}(t) - C^{-1}(z, \theta)M(z)u(t). \quad (2.19)$$

In general, the measured output $\tilde{y}(t)$ is affected by noise, so it can be expressed as $\tilde{y}(t) = y(t) + \nu(t) = P(z)u(t) + \nu(t)$, where $y(t)$ is the noise-free component of the output. It follows that in (2.18) and (2.19) a term related to the measurement noise appears. Nevertheless, under the assumption that the input signal $u(t)$ and the output disturbance $\nu(t)$ are uncorrelated and the number of samples tends to infinity, the solution that

Chapter 2. Overview of data-driven tuning methods

minimises a cost function based on the discrepancies tends to the noise-free one.

Hence, the two cost functions can be respectively defined as the squared Euclidean norm of (2.18) and (2.19):

$$J_n = \|(u - u_0)\|^2 \quad (2.20)$$

$$J_v = \|(\tilde{y} - y_0)\|^2. \quad (2.21)$$

In conclusion, to obtain a single objective minimisation problem, these two cost functions have to be combined together, as previously disclosed in (2.12), resulting in the following design criterion:

$$\hat{\theta}(\delta) = \arg \min_{\theta \in \Theta_S} \{(1 - \delta)J_n(\theta) + \delta J_v(\theta)\} \quad (2.22)$$

where $\hat{\theta}$ is the optimal solution associated with the controller that simultaneously guarantees internal stability and the best possible closed-loop performance, and Θ_S is the set of all the parameter vectors θ such that $C^{-1}(z)Q(z)$ is stable.

2.3.3 Stability constraint and implementation

In general, a controller that exactly achieves $M(z)$ stabilises the plant if and only if the unstable zeros of $P(z)$ and $M(z)$ are the same. Since the plant model is unknown, the choice of an appropriate reference model $M(z)$ is not trivial. As already mentioned, the cost function J_n associated with the control sensitivity reference model $Q(z)$ is critical for considering the stability requirement. Indeed, if the plant is stable, the controller $C_Q(z)$ that exactly achieves the reference model

$$Q(z) = \frac{C_Q(z)}{1 + P(z)C_Q(z)}, \quad (2.23)$$

2.3. Controller unfalsification

guarantees stability of the closed-loop system. Consequently, when $\delta = 0$ the minimisation of criterion (2.22) ensures that the stability requirement is met. However, the stability is preserved also in a neighbourhood of δ equal to zero, in fact, the following theorem holds.

Theorem 1. *Let $\hat{\theta}(0)$ be the solution of the minimisation process of J_n , so that the controller $C(z, \hat{\theta}(0))$ stabilises the plant. Then, there exists $\bar{\delta} > 0$, such that for any $\delta < \bar{\delta}$ the controller $C(z, \hat{\theta}(\delta))$ is stabilising.*

Thanks to Theorem 1, it is possible to derive an effective tuning strategy that takes into account the stability requirement, but before introducing the practical implementation of the algorithm it is useful to specify what the stability test consists of. The input discrepancy can be rewritten as function of the inverse of $C_Q(z)$ and $C(z)$ as suggested by Proposition 1.

Proposition 1. *The input discrepancy can be written as*

$$u(t) - u_0(t) = \Delta_Q(z, \theta)Q(z)u(t) - Q(z)n(t), \quad (2.24)$$

where

$$\Delta_Q = C_Q^{-1}(z) - C^{-1}(z, \theta). \quad (2.25)$$

By using Proposition 1 and relying on the small-gain theory, Theorem 2 can be proven, thus obtaining the stability test. In addition, Proposition 1 clarifies the existing relationship between input discrepancy and closed-loop stability. Proofs of both theorems are widely discussed in [3].

Theorem 2. *Let θ be the controller parameter vector. Then, if*

$$\|Q(z)\Delta_Q(z, \theta)\|_\infty < 1, \quad (2.26)$$

the controller $C(z, \theta)$ stabilises the unknown plant $P(z)$.

Chapter 2. Overview of data-driven tuning methods

Algorithm 4 Control unfalsification tuning method.

- 1: **repeat**
 - 2: Compute $\hat{\theta}(\delta) = \arg \min_{\theta \in \Theta_S} \{(1 - \delta)J_n(\theta) + \delta J_v(\theta)\}$, with $\delta = 0$.
 - 3: Compute $u - u_0$ and estimate $\|Q(z)\Delta_Q(z, \hat{\theta}(0))\|_\infty$.
 - 4: **until** $\|Q(z)\Delta_Q(z, \hat{\theta}(0))\|_\infty < 1 - \alpha$ otherwise change $M(z)$.
 - 5: **repeat**
 - 6: Set $\delta = 1/2^{iter-1}$.
 - 7: Compute $\hat{\theta}(\delta) = \arg \min_{\theta \in \Theta_S} \{(1 - \delta)J_n(\theta) + \delta J_v(\theta)\}$, with $\hat{\theta}(0)$ as initial guess;
 - 8: Compute $u - u_0$ and estimate $\|Q(z)\Delta_Q(z, \hat{\theta}(\delta))\|_\infty$.
 - 9: **until** $\|Q(z)\Delta_Q(z, \hat{\theta}(\delta))\|_\infty < 1 - \alpha$.
-

Finally, as suggested in [3], the H_∞ norm can be effectively estimated relying only on the gathered dataset and input discrepancy. So, it can be rewritten as

$$\|Q(z)\Delta_Q(z, \theta)\|_\infty \simeq \sup \frac{|\hat{u}(\omega) - \hat{u}_0(\omega)|}{|\hat{u}(\omega)|}, \quad (2.27)$$

where, $\hat{u}(\omega)$ and $\hat{u}_0(\omega)$ are the discrete Fourier transforms of $u(t)$ and $u_0(t)$, respectively.

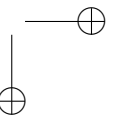
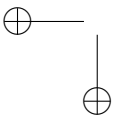
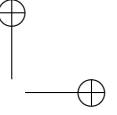
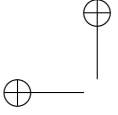
Algorithm implementation

As already mentioned, if the experimental data-test is sufficiently informative and the reference models are appropriately chosen, then minimisation (2.22) leads to a stabilising controller when $\delta = 0$. This solution can be used as a first guess for a second minimisation process, but this time with the aim of looking for the largest δ that guarantees stability. In fact, thanks to Theorem 1, starting from $\delta = 1$ and then gradually decreasing it, we certainly find a value of δ , more or less close to zero, such that the minimisation (2.22) leads to a stabilising controller. From a practical point of view, the stability test is passed only if the estimate of $\|Q(z)\Delta_Q(z, \theta)\|_\infty$ is less than $1 - \alpha$. The scalar α is required for taking into account the estimating

2.3. Controller unfalsification

error related to a finite dataset and the presence of the disturbance.

To sum up, the tuning procedure developed in [3] is shown the Algorithm 4 in a synthetic way.



CHAPTER 3

UAV control architectures

IN this chapter, the control architectures of the multirotor systems employed in this work are presented. In particular two UAVs share the same architecture whereas the third UAV exploits a more advanced control structure. For each architecture the unknown controller parameters are also illustrated. These parameters will be tuned with the methods presented in Chapter 2 and Chapter 4 and the results are provided in Chapter 6.

The chapter is organised as follows. Section 3.1 displays the overall control architecture that includes both the position and the attitude controllers. The first control structure is then illustrated in Section 3.2. It contains SISO PID controllers and for this type of architecture only the attitude controller is provided. For the second control structure both the attitude and the position controllers are presented in Section 3.3 and Section 3.4 respectively.

Chapter 3. UAV control architectures

3.1 Overall control architecture

The control of UAVs and in particular of multirotors is a challenging problem mainly for two reasons: their dynamics are characterized by nonlinearities and they are underactuated with respect to the six rigid body Degrees of Freedom (DoFs). Quadrotor control synthesis has been studied extensively in the literature. A classical PID architecture for both attitude and position control remains one of the most common choices, see for example [43], against the more modern LQ technique, as discussed in *e.g.*, [8]. Indeed, while for aggressive manoeuvring flight nonlinear models and nonlinear control design methods are needed (see, *e.g.*, [39] for a recent survey), if one is mainly concerned with applications such as inspection, surveillance, mapping, video and photography (which, incidentally, cover most of the actual market for this type of vehicles) then linear modelling and control design methods are more suitable, while on the other hand, the expected performance level is significantly higher. The state-of-the-art in linear control for small scale helicopters is given by approaches such as, *e.g.*, [31], in which modern robust control design is coupled to identification of linear rotorcraft models. A nonlinear control method with proven effectiveness for underactuated system is the backstepping, see for example [9], where it is compared with sliding-mode technique. More general approaches, on the other hand, consider nonlinear trajectory planning and tracking techniques. Many methods have been proposed, covering, *e.g.*, control on nonlinear manifolds [33], adaptive control [11,38], dynamic inversion [17] and feedback linearization [37]. Of particular interest are methods for planning and tracking based on the flatness property of quadrotor dynamics (see, *e.g.*, [20,40]) as well as procedures based on smoothing of a given trajectory using motion primitives (see for example [7]).

Regarding multirotor platforms, the most popular architecture adopts variable rotor angular rate as control input (with fixed rotors blade pitch): this choice is primarily due to rotors hub mechanical simplicity and weight considerations, but, as it will be explained in Section 5.1, one UAV considered in this work adopts an alternative architecture with variable rotor blade

3.1. Overall control architecture

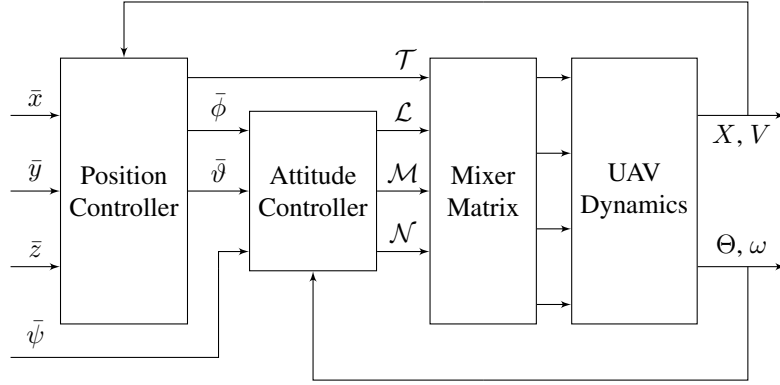


Figure 3.1: The overall control architecture for multirotor system.

collective pitch and fixed rotors angular rates.

The overall control architecture of all the UAVs exploited in this work is the same: it is a hierarchical control approach with an outer loop that controls the position of the UAV and generates the attitude setpoint for an inner loop that control the attitude of the quadrotor (see Figure 3.1). The position and the attitude controllers are not independent because the system is underactuated and *e.g.*, if the user want to move the UAV forward he has to tilt it. Furthermore, the position and the attitude measurements come from a Kalman-based estimator already implemented in the Flight Control Unit (FCU).

The position controller is fed with the position reference signals ($\bar{x}(t)$, $\bar{y}(t)$ and $\bar{z}(t)$) and the measurements that comes from the estimator, in particular the information about the position

$$X(t) = [x(t) \quad y(t) \quad z(t)]^T$$

where $x(t)$, $y(t)$ and $z(t)$ are the longitudinal, lateral and vertical positions respectively. The controller exploits also the information about the UAV linear velocity

$$V(t) = [v_x(t) \quad v_y(t) \quad v_z(t)]^T$$

Chapter 3. UAV control architectures

where $v_x(t)$, $v_y(t)$ and $v_z(t)$ are the longitudinal, lateral and vertical linear velocities respectively. The position controller generates the total thrust $\mathcal{T}(t)$ and the reference signal for the roll ($\bar{\phi}(t)$) and pitch ($\bar{\vartheta}(t)$) angles. The yaw angle setpoint $\bar{\psi}(t)$ is provided by the user. Exploiting the attitude setpoints, the inner attitude controller generates the required roll ($\mathcal{L}(t)$), pitch ($\mathcal{M}(t)$) and yaw ($\mathcal{N}(t)$) moments. The inner controller employs also the information about the attitude

$$\Theta(t) = \begin{bmatrix} \phi(t) & \vartheta(t) & \psi(t) \end{bmatrix}^T$$

where $\phi(t)$, $\vartheta(t)$ and $\psi(t)$ are the roll, pitch and yaw angles respectively and the measured angular rates

$$\omega(t) = \begin{bmatrix} p(t) & q(t) & r(t) \end{bmatrix}^T$$

where $p(t)$, $q(t)$ and $r(t)$ are the roll, pitch and yaw angular rates respectively. The measured attitude angles and angular rates come from the UAV state estimator.

The total thrust and the moments generated by position and attitude controllers feed the *mixer matrix* block. This matrix relates the required thrust and moments around each axis to the control inputs of the UAV. The number and the type of outputs of these block depend on the type of UAV: in a quadrotor there are four control inputs and, if it adopts variable rotor angular rates, it is related to the speed of the propellers. Whereas, if the UAV exploits variable rotor blade collective pitch to generate the required thrust and moments, the control inputs are the pitch angles of the rotor blades.

The control architecture was already implemented in the FCU of each UAV. The structure of the controllers was designed assuming decoupled dynamics between the Degree of Freedoms (DoFs). This assumption is valid only if the UAV is considered in near-hovering conditions, and falls progressively as the forward velocity increases. This means that the longitudinal/pitch, lateral/roll, vertical and yaw attitude DoFs are controlled independently. As was discussed previously, the longitudinal and pitch

3.2. SISO pitch attitude controller

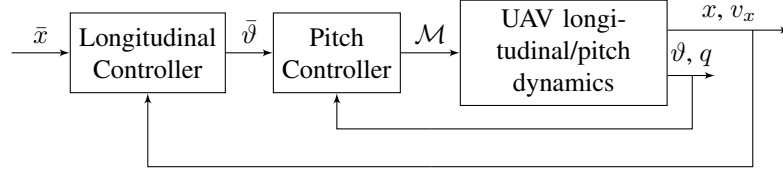


Figure 3.2: The controller architecture for the longitudinal/pitch DoFs.

DoFs and the lateral and roll DoFs are coupled because, in order to obtain a longitudinal movement, the quadrotor must have a pitch angle different from zero and in the same way a roll angle different from zero implies a lateral movement. All the data-driven methods are applied to the longitudinal/pitch DoFs (see Figure 3.2 where the mixer matrix block is omitted for the sake of simplicity). This must not be considered a simplification of tuning the entire control architecture presented in Figure 3.1. Indeed, the same results can be applied for lateral/roll DoFs, considering the geometrical symmetry of the quadrotor, and can be extended to yaw and vertical dynamics since also these controllers have a similar structure. In the following, only the scheme in Figure 3.2 is considered.

3.2 SISO pitch attitude controller

Two of the quadrotors considered in this work share the same pitch attitude control architecture. It is based on cascaded SISO PID loops and it is displayed in Figure 3.3. In detail, the outer regulator $C_o(z)$ is a PD con-

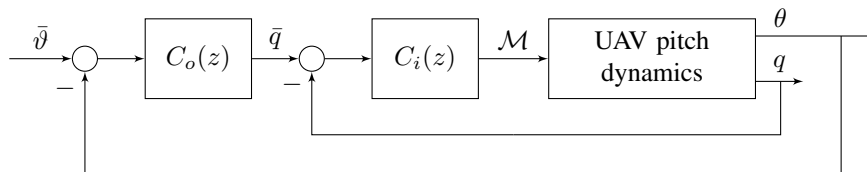


Figure 3.3: The pitch attitude controller based on SISO PID architecture.

troller based on attitude feedback (estimated angle $\vartheta(t)$, set-point $\bar{\vartheta}(t)$) and the inner regulator $C_i(z)$ is a PID controller based on angular rate feedback (measured angular velocity $q(t)$, set-point $\bar{q}(t)$). The inner controller presents an integral term to reject load disturbances and to achieve zero

Chapter 3. UAV control architectures

steady state tracking error whereas the outer loop does not provide an integral action because the disturbances are already rejected by the inner loop and, in most of applications, as in our case, the outer variable is simply the integral of the inner one, multiplied by a gain. Thus, considering the outer loop, an integral term is already included in the loop transfer function. The transfer functions of the outer and the inner controllers are:

$$C_o(z) = K_P^O + K_D^O \frac{z-1}{zT_s}$$

$$C_i(z) = K_P^I + K_I^I \frac{T_s}{z-1} + K_D^I \frac{z-1}{zT_s}$$

where T_s is the sampling time. The unknown controller parameters that will be tuned with the data-driven approach are:

$$\theta = \left[K_P^O \quad K_D^O \quad K_P^I \quad K_D^I \quad K_I^I \right]^T.$$

The control variable considered as output of the inner regulator is the pitching moment $\mathcal{M}(t)$.

3.3 MISO pitch attitude controller

The previous section presents the pitch attitude controller that is implemented on two of the three different UAVs considered in this work. As will be explained in Chapter 5, a third UAV is considered and it employs a different control architecture. Instead of changing the control structure in order to match the architecture presented in Section 3.2, the new structure is analysed and the data-driven methods are extended in Section 4.2 to deal with this new controller. The new control architecture is now illustrated.

As the control structure presented in Section 3.2, the new control architecture is based on two cascaded PID loops. The outer loop on the pitch attitude feedback is a classical proportional controller but the inner regulator, based on the pitch angular rate, presents a different architecture respect to the classical PID structure. It has a *feed-forward* gain directly computed

3.3. MISO pitch attitude controller

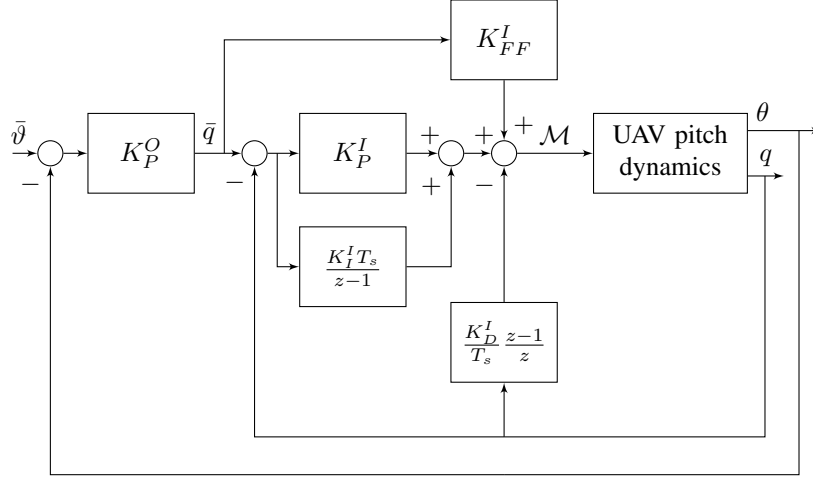


Figure 3.4: The pitch attitude controller with feed-forward gain and derivative action on plant measurements.

on the pitch angle setpoint and the derivative action is computed starting from the plant output $q(t)$ and not from the pitch angular rate error (see Figure 3.4). This last expedient avoids to generate an impulse on the control action $\mathcal{M}(t)$ when there is a step in the reference signal $\bar{q}(t)$ of the inner loop.

The output of the outer controller can be computed as:

$$\bar{q}(t) = K_P^O (\bar{\vartheta}(t) - \vartheta(t))$$

while the control input is computed by the inner controller as:

$$\mathcal{M}(t) = K_{FF}^I \bar{q}(t) + \left(K_P^I + \frac{K_I^I T_s}{z-1} \right) (\bar{q}(t) - q(t)) - \frac{K_D^I}{T_s} \frac{z-1}{z} q(t).$$

The unknown controller parameters that will be tuned with the data-driven approach are:

$$\theta = \left[K_P^O \quad K_{FF}^I \quad K_P^I \quad K_D^I \quad K_I^I \right]^T. \quad (3.1)$$

Chapter 3. UAV control architectures

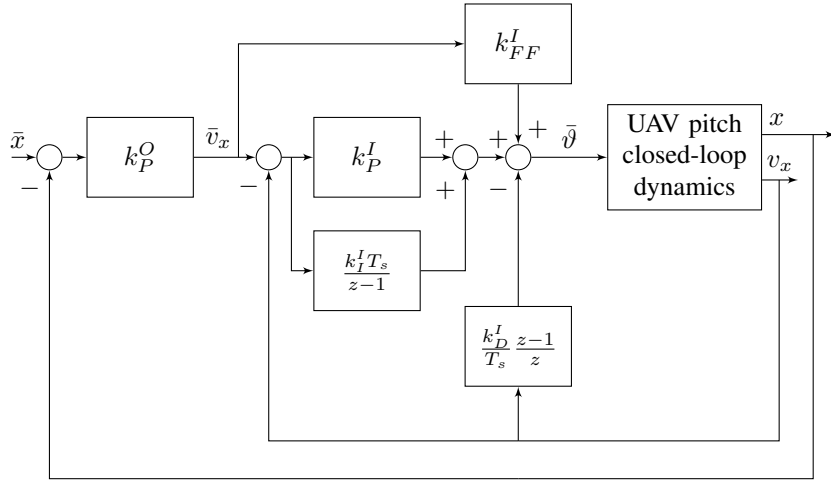


Figure 3.5: The longitudinal position controller with feed-forward and derivative action based on the plant measurements.

3.4 MISO longitudinal position controller

As will be explained in Section 5.3, where the third UAV is introduced, the size of this quadrotor and the high flexibility and customisation of its FCU allow to perform in-flight identification experiments. This feature opens the data-driven approach to a different type of controllers: the position regulator. As displayed in Figure 3.1 the position controller is located in an higher level than the attitude controller. It generates the reference signal for the attitude regulator starting from position setpoint and position measurements. In this work only the longitudinal position controller is considered since it is coupled with the pitch attitude regulator and it is independent from the controller of other DoFs.

The control architecture follows the structure of the attitude controller shown in Figure 3.4: in this case there is a proportional controller on the outer loop based on the longitudinal position that receives the position error as input and returns longitudinal linear velocity reference as output. The inner loop is associated with the longitudinal velocity and presents a derivative term computed on the plant output $v_x(t)$ and the proportional and integral terms calculated on the velocity error (see Figure 3.5). The inner controller computes the pitch angle setpoint that feeds the scheme in

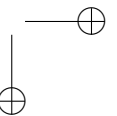
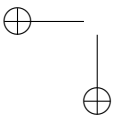
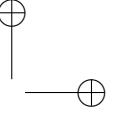
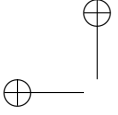
3.4. MISO longitudinal position controller

Figure 3.4.

Since the control architecture is the same as for the attitude regulator in Section 3.3, the number of the unknown controller parameters are the same and they are:

$$\theta = \left[k_P^O \quad k_{FF}^I \quad k_P^I \quad k_D^I \quad k_I^I \right]^T.$$

The unknown controller parameters are expressed with lower case letters in order to remark the difference with the unknown parameters in (3.1) although the position and the attitude controllers share the same structure.



CHAPTER 4

Data-driven algorithms for multirotor systems

IN this chapter the extensions of the data-driven methods presented in Chapter 2 are listed. In particular, the VRFT and the CbT are extended to solve specific tuning problems related to UAVs and related to the controller architectures depicted in Chapter 3.

This chapter is organised as follows. In Section 4.1 the VRFT method considers experimental data from closed-loop flight tests. Subsequently, Section 4.2 provides an extension to the standard VRFT method to deal with the controller architecture with a feed-forward term and a derivative action related to the measured output. Finally, in order to compare the VRFT and the CbT algorithms, the CbT method is extended in Section 4.3 to tune a cascade control system with a single experimental dataset.

Chapter 4. Data-driven algorithms for multirotor systems

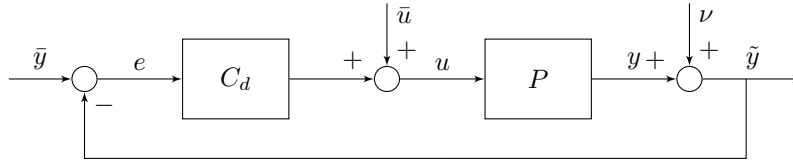


Figure 4.1: VRFT experiment in closed-loop operation.

4.1 Closed Loop Experiments

The standard VRFT algorithm presented in Section 2.1 exploits experimental data that come from an open-loop test. Performing such a test is not always possible. Some of the applications are unstable by nature and have to operate in closed-loop. Sometimes also a stable system must operate in closed-loop during the experiment to satisfy some conditions related to the experiment or due to safety reasons. In this work the VRFT method is employed to tune the controllers of UAVs. When the test to collect data is performed in flight, the UAV could collide on some obstacles or exceed the test area. In all these situations the data must be collected in closed-loop allowing the user to control the system also when the experiment is been conducting. Furthermore, closed-loop tests allow us to perform the experiment to collect the data without exploiting a test-bed and without performing changes on the system to control.

Obviously, in order to perform the closed-loop test, an initial controller that stabilise the system must be available. The discussion on how this controller is obtained goes beyond the scope of this work but, if this initial controller is employed only to collect the closed-loop data, the reader should consider the fast way to obtain the controller since it has not to reach some closed-loop performance goal and it has only to stabilise the system.

As illustrated in Figure 4.1, the excitation signal $\bar{u}(t)$ is added to the output of the controller $C_d(z)$. $C_d(z)$ is the initial controller that stabilises the system. The user can act on $\bar{y}(t)$ to control the behaviour of the system also during the experiment.

The standard VRFT method, as described in Section 2.1, cannot be applied to obtain a new controller exploiting the measurements $D_N =$

4.1. Closed Loop Experiments

$\{u(t), \tilde{y}(t)\}_{t=1..N}$: specific problems arise when the instrumental variable is created because $u(t)$ and $\nu(t)$ are now correlated. Indeed, the user cannot directly act on the input of the plant as in the standard VRFT but he can operate on the signals $\bar{y}(t)$ and $\bar{u}(t)$ and the input of the plant is now affected by this action:

$$u(t) = \frac{1}{1 + C_d(z)P(z)}\bar{u}(t) + \frac{C_d(z)}{1 + C_d(z)P(z)}(\bar{y}(t) - \nu(t)). \quad (4.1)$$

For the sake of simplicity the assumption that the user does not provide a setpoint during the experiment can be made ($\bar{y}(t) = 0 \quad \forall t$) and (4.1) can be rewritten as:

$$u(t) = \frac{1}{1 + C_d(z)P(z)}\bar{u}(t) - \frac{C_d(z)}{1 + C_d(z)P(z)}\nu(t). \quad (4.2)$$

When the experimental data is collected in closed-loop it is not always possible to use a second experiment to build the instrumental variable. Indeed, the user can select the same signal $\bar{u}(t)$ in the repeated experiment but this does not imply that the input of the plant $u(t)$ will be the same in the two experiments, violating the assumption made in Section 2.1. Since VRFT is employed in this work to tune the controller of an UAV and VRFT is extended in this section to perform the experiment in flight, a lot of environment uncertainties affect the experiment and it is impossible to ensure that the input of the plant is the same in the two flight tests. Thus, the instrumental variable must be build using the second choice presented in Section 2.1: the identification of the plant. Using (4.2) to build the instrumental variable as described in Algorithm 1 leads to a biased controller parameter vector since the instrumental variable is no longer uncorrelated with the noise $\nu(t)$. Indeed, the instrumental variable is built as

$$\hat{y}(t) = \hat{P}(z)u(t) \quad (4.3)$$

Chapter 4. Data-driven algorithms for multirotor systems

and using (4.2):

$$\hat{y}(t) = \hat{P}(z) \left(\frac{1}{1 + C_d(z)P(z)} \bar{u}(t) - \frac{C_d(z)}{1 + C_d(z)P(z)} \nu(t) \right). \quad (4.4)$$

Following Algorithm 1, the instrumental variable is:

$$\begin{aligned} \zeta(t) &= \beta(z)L(z) (M(z)^{-1} - 1) \hat{y}(t) \\ &= \beta(z)L(z) (M(z)^{-1} - 1) \\ &\quad \hat{P}(z) \left(\frac{1}{1 + C_d(z)P(z)} \bar{u}(t) - \frac{C_d(z)}{1 + C_d(z)P(z)} \nu(t) \right). \end{aligned} \quad (4.5)$$

The previous equation clearly shows the correlation between $\zeta(t)$ and $\nu(t)$. To solve this problem a different instrumental variable must be chosen in order to ensure a correlation with the regression variable and an uncorrelation with the noise. Two choices for the instrumental variable are now proposed. Let

$$\hat{y}_{\bar{u}}^{OL}(t) = \hat{P}(z)\bar{u}(t)$$

be the simulated output of the plant when it is fed only by the excitation signal $\bar{u}(t)$ in open-loop. The first proposed choice is as follows:

$$\begin{aligned} \zeta(t) &= \beta(z)L(z) (M(z)^{-1} - 1) \hat{y}_{\bar{u}}^{OL}(t) \\ &= \beta(z)L(z) (M(z)^{-1} - 1) \hat{P}(z)\bar{u}(t). \end{aligned} \quad (4.6)$$

The second option for the instrumental variable is:

$$\zeta(t) = \beta(z)L(z) (M(z)^{-1} - 1) \hat{y}_{\bar{u}}^{CL}(t) \quad (4.7)$$

where

$$\hat{y}_{\bar{u}}^{CL}(t) = \frac{\hat{P}(z)}{1 + C_d(z)\hat{P}(z)} \bar{u}(t)$$

4.1. Closed Loop Experiments

is the simulated output of the plant when it is fed only by the excitation signal $\bar{u}(t)$ in closed-loop condition. The choices expressed in (4.6) and (4.7) allow the instrumental variable to be uncorrelated with the noise $\nu(t)$ since it depends from $\bar{u}(t)$ and not from $u(t)$. Note that if the instrumental variable is built as in (4.6) the initial controller $C_d(z)$ might be unknown whereas with the second method the user must also know it.

Nevertheless the identification procedure exploited in (4.3) to obtain the model of the plant could be very challenging and the identification method must be selected accordingly since the data is collected in closed-loop and classical identification methods fail with this type of data. Subspace Model Identification (SMI) methods emerge as a viable approach to solve this task, in particular the Predictor Based System Identification (PBSID) method that it will be presented in the next subsection (see [50] for an overview of closed-loop SMI methods).

4.1.1 The PBSID algorithm

As illustrated in [13], PBSID is an advanced and recent model identification algorithm with the ability of dealing with data generated in closed-loop. It belongs to the class of black-box methods: it allows to determine dynamics model of a system using only the input-output data gathered in the identification experiments. The obtained model is unstructured, namely with a non-physically motivated state space. Furthermore, since PBSID is a SMI algorithm, it is a non-iterative method: it can be implemented with numerically stable and efficient tools from numerical linear algebra and it has proved to be extremely successful in dealing with the estimation of state-space models MIMO systems in a number of rotorcraft applications (see [4, 5, 35, 53]).

The PBSID algorithm, which is briefly described in the following, considers the finite dimensional, Linear Time Invariant (LTI) state space model class

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ \tilde{y}(k) &= Cx(k) + Du(k) + \nu(k) \end{aligned} \tag{4.8}$$

Chapter 4. Data-driven algorithms for multirotor systems

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $\tilde{y}(k) \in \mathbb{R}^p$ and $\{\nu(k), w(k)\}$ are ergodic sequences of finite variance satisfying

$$E \begin{bmatrix} w(t) \\ \nu(t) \end{bmatrix} \begin{bmatrix} w(s)^T & \nu(s)^T \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{s,t},$$

with $\delta_{s,t}$ denoting the Kronecker delta function, possibly correlated with the input $u(k)$.

Let now

$$z(k) = \begin{bmatrix} u^T(k) & y^T(k) \end{bmatrix}^T$$

and

$$\bar{A} = A - KC, \quad \bar{B} = B - KD, \quad \tilde{B} = \begin{bmatrix} \bar{B} & K \end{bmatrix},$$

where K is the Kalman gain associated with (4.8), and note that system (4.8) can be written as

$$\begin{aligned} x(k+1) &= \bar{A}x(k) + \tilde{B}z(k) \\ \tilde{y}(k) &= Cx(k) + Du(k) + e(k), \end{aligned} \quad (4.9)$$

where $e(\cdot)$ is the innovation vector. The data equations for the PBSID algorithm can be then derived by noting that propagating $p-1$ steps forward the first of equations (4.9), where p is the so-called past window length, one gets

$$\begin{aligned} x(k+2) &= \bar{A}^2x(k) + \begin{bmatrix} \bar{A}\tilde{B} & \tilde{B} \end{bmatrix} \begin{bmatrix} z(k) \\ z(k+1) \end{bmatrix} \\ &\vdots \\ x(k+p) &= \bar{A}^p x(k) + \mathcal{K}^p Z^{0,p-1} \end{aligned} \quad (4.10)$$

4.1. Closed Loop Experiments

where

$$\mathcal{K}^p = \left[\bar{A}^{p-1} \tilde{B}_0 \quad \dots \quad \tilde{B} \right] \quad (4.11)$$

is the extended controllability matrix of the system and

$$Z^{0,p-1} = \begin{bmatrix} z(k) \\ \vdots \\ z(k+p-1) \end{bmatrix}.$$

Under the considered assumptions, \bar{A} represents the dynamics of the optimal one-step ahead predictor for the system and therefore it has all the eigenvalues inside the open unit circle, so the term $\bar{A}^p x(k)$ is negligible for sufficiently large values of p and we have that

$$x(k+p) \simeq \mathcal{K}^p Z^{0,p-1}.$$

As a consequence, the input-output behavior of the system is approximately given by

$$\begin{aligned} \tilde{y}(k+p) &\simeq C\mathcal{K}^p Z^{0,p-1} + Du(k+p) + e(k+p) \\ &\vdots \\ \tilde{y}(k+p+f) &\simeq C\mathcal{K}^p Z^{f,p+f-1} + Du(k+p+f) + \\ &\quad + e(k+p+f). \end{aligned} \quad (4.12)$$

Chapter 4. Data-driven algorithms for multirotor systems

Introducing the matrix notation defined as:

$$\begin{aligned}\bar{Z}^{p,f} &= \left[Z^{0,p-1} \quad Z^{0,p} \quad \dots \quad Z^{0,p+f-1} \right]^T \\ X^{p,f} &= \left[x(k+p) \quad x(k+p+1) \quad \dots \quad x(k+p+f) \right]^T \\ Y^{p,f} &= \left[\tilde{y}(k+p) \quad \tilde{y}(k+p+1) \quad \dots \quad \tilde{y}(k+p+f) \right]^T \\ U^{p,f} &= \left[u(k+p) \quad u(k+p+1) \quad \dots \quad u(k+p+f) \right]^T \\ E^{p,f} &= \left[e(k+p) \quad e(k+p+1) \quad \dots \quad e(k+p+f) \right]^T,\end{aligned}$$

the data equations are given by

$$\begin{aligned}X^{p,f} &\simeq \mathcal{K}^p \bar{Z}^{p,f} \\ Y^{p,f} &\simeq C \mathcal{K}^p \bar{Z}^{p,f} + D U^{p,f} + E^{p,f}.\end{aligned}\quad (4.13)$$

Considering $p = f$ (where f is the so-called future window length), estimates for the matrices $C \mathcal{K}^p$ and D are first computed by solving the least-squares problem

$$\min_{C \mathcal{K}^p, D} \|Y^{p,p} - C \mathcal{K}^p \bar{Z}^{p,p} - D U^{p,p}\|_F. \quad (4.14)$$

Defining now the extended observability matrix Γ^p as

$$\Gamma^p = \begin{bmatrix} C \\ C \bar{A} \\ \vdots \\ C \bar{A}^{p-1} \end{bmatrix} \quad (4.15)$$

4.1. Closed Loop Experiments

and noting that the product of Γ^p and \mathcal{K}^p can be written as

$$\Gamma^p \mathcal{K}^p \simeq \begin{bmatrix} C\bar{A}^{p-1}\tilde{B} & \dots & C\tilde{B} \\ 0 & \dots & C\bar{A}\tilde{B} \\ \vdots & \ddots & \vdots \\ 0 & \dots & C\bar{A}^{p-1}\tilde{B} \end{bmatrix}, \quad (4.16)$$

such product can be computed using the estimate $\widehat{C\mathcal{K}^p}$ of $C\mathcal{K}^p$ obtained by solving the least squares problem (4.14). Recalling now that

$$X^{p,p} \simeq \mathcal{K}^p \bar{Z}^{p,p} \quad (4.17)$$

it also holds that

$$\Gamma^p X^{p,p} \simeq \Gamma^p \mathcal{K}^p \bar{Z}^{p,p}. \quad (4.18)$$

Therefore, computing the SVD

$$\Gamma^p \mathcal{K}^p \bar{Z}^{p,p} = U\Sigma V^T \quad (4.19)$$

an estimate of the state sequence can be obtained as

$$\widehat{X}^{p,p} = \Sigma_n^{1/2} V_n^T = \Sigma_n^{-1/2} U_n^T \Gamma^p \mathcal{K}^p \bar{Z}^{p,p}, \quad (4.20)$$

from which, in turn, an estimate of C can be computed by solving the least squares problem

$$\min_C \|Y^{p,p} - \widehat{D}U^{p,p} - C\widehat{X}^{p,p}\|_F. \quad (4.21)$$

Chapter 4. Data-driven algorithms for multirotor systems

Algorithm 5 The control system with closed-loop excitation data.

- 1: Compute $U(z)$ such that $|U(e^{j\omega})|^2 = \Phi_u(\omega)$.
 - 2: Compute $L(z) = (1 - M(z))M(z)W(z)U^{-1}(z)$.
 - 3: Compute $u_L(t)$ as $u_L(t) = L(z)u(t)$.
 - 4: Compute $\varphi(t) = \beta(z)L(z)(M(z)^{-1} - 1)\tilde{y}(t)$.
 - 5: Identify the plant model $\hat{P}(z)$ with the PBSID algorithm.
 - 6: Compute $\hat{y}_{\bar{u}}(t) = \frac{\hat{P}(z)}{1 + C_d(z)\hat{P}(z)}\bar{u}(t)$.
 - 7: Compute the instrumental variable $\zeta(t) = \beta(z)L(z)(M(z)^{-1} - 1)\hat{y}_{\bar{u}}(t)$.
 - 8: Compute $\hat{\theta}_N^{IV} = \left[\sum_{t=1}^N \zeta(t)\varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \zeta(t)u_L(t)$.
-

The final steps consist of the estimation of the innovation data matrix $E_N^{p,f}$

$$E_N^{p,f} = Y^{p,p} - \hat{C}\hat{X}^{p,p} - \hat{D}U^{p,p} \quad (4.22)$$

and of the entire set of the state space matrices for the system, which can be obtained by solving the least squares problem

$$\min_{A,B,K} \|\hat{X}^{p+1,p} - A\hat{X}^{p,p-1} - BU^{p,p-1} - KE^{p,p-1}\|_F. \quad (4.23)$$

In [28] a recursive real-time implementation of the PBSID method is presented. These features allow to obtain a suitable online estimation exploiting open-loop or closed-loop data.

Algorithm 5 recaps all the steps needed to exploit VRFT to tune the controller when the data is collected in closed-loop.

4.2 Controller with feed-forward and action on plant output

As the reader should already understand, the definition of a new controller architecture goes beyond the scope of this work and the main goal is to use data-driven methods to tune or re-tune a controller that is already implemented. In particular, in this work the VRFT method will be apply to different UAVs that exploit two controller architectures (see Chapter 3). The first controller is a classical SISO PID controller and it can be represented by the Scheme 1.1 and it can be tuned with the classical VRFT

4.2. Controller with feed-forward and action on plant output

algorithm, as presented in Section 2.1. The second controller architecture is a modification of the first one: a feed-forward term is added and there is an action that is computed on the plant output directly (see Figure 3.4 and Figure 3.5). As shown in Chapter 3, where the controller architectures are illustrated in details, this class of controllers includes a PID controller with a feed-forward gain and with the derivative action computed on the plant output instead on the error. This controller can be considered as a Multiple

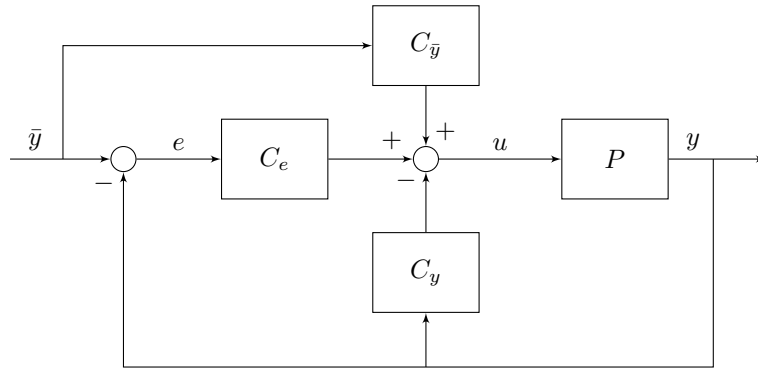


Figure 4.2: Controller with feed forward term and action on plant measurements.

Input Single Output (MISO) system with two inputs, the setpoint $\bar{y}(t)$ and the plant output $\tilde{y}(t)$, and one output, the control action $u(t)$. The VRFT algorithm, in this section, is extended to consider this controller architecture.

A generalisation of the scheme in Figure 3.4 and Figure 3.5 is now considered and it is represented in Figure 4.2 where a noiseless environment is considered for simplicity. The output of the controller is:

$$u(t) = C_{\bar{y}}(z, \theta_{\bar{y}}) \bar{y}(t) + C_e(z, \theta_e) e(t) - C_y(z, \theta_y) y(t). \quad (4.24)$$

Considering the new controller architecture, also the criterion minimised by the VRFT algorithm in (2.1) changes as:

$$J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}}) = \frac{1}{N} \sum_{t=1}^N L(z) \left[u(t) - C_e(z, \theta_e) e(t) - C_{\bar{y}}(z, \theta_{\bar{y}}) \bar{y}(t) + C_y(z, \theta_y) y(t) \right]^2. \quad (4.25)$$

Chapter 4. Data-driven algorithms for multirotor systems

If the controller class is linearly parameterised, the minimum of the criterion (4.25) can be found exploiting the the closed form solution as in (2.5). Indeed, in this case the controllers can be rewritten as:

$$\begin{aligned} C_{\bar{y}}(z, \theta_{\bar{y}}) &= \beta_{\bar{y}}(z)\theta_{\bar{y}} \\ C_e(z, \theta_e) &= \beta_e(z)\theta_e \\ C_y(z, \theta_y) &= \beta_y(z)\theta_y. \end{aligned}$$

Defining

$$\theta = \begin{bmatrix} \theta_{\bar{y}} \\ \theta_e \\ \theta_y \end{bmatrix}$$

and

$$\varphi_L(t) = \begin{bmatrix} \beta_{\bar{y}}(z)M(z)^{-1} \\ \beta_e(z)(M(z)^{-1} - 1) \\ \beta_y(z) \end{bmatrix} L(z)y(t),$$

the criterion in (4.25) can be rewritten as:

$$J_{VR}^N(\theta) = \frac{1}{N} \sum_{t=1}^N (u_L(t) - \varphi_L^T(t)\theta)^2, \quad (4.26)$$

and the optimal parameters are:

$$\hat{\theta}_N = \arg \min_{\theta} J_{VR}^N(\theta).$$

The closed-form solution of the problem in the previous equation exists and it is equal to (2.5).

Considering the new controller architecture, a different selection of the

4.2. Controller with feed-forward and action on plant output

filter $L(z)$ must be made. The filter is shaped starting from the derivation in [32] where the filter is built for a two degree of freedom controller.

Let

$$S(z) = \frac{1}{1 + (\bar{C}_e(z) + \bar{C}_y(z)) P(z)}$$

be the reference model for the sensitivity function, that is the transfer function between $\nu(t)$ and $y(t)$ and

$$M(z) = \frac{(\bar{C}_{\bar{y}}(z) + \bar{C}_e(z)) P(z)}{1 + (\bar{C}_e(z) + \bar{C}_y(z)) P(z)} \quad (4.27)$$

be the reference model for the closed-loop function: the transfer function between $\bar{y}(t)$ and $y(t)$. $\bar{C}_{\bar{y}}(z)$, $\bar{C}_e(z)$ and $\bar{C}_y(z)$ are the optimal transfer functions that solve exactly the model matching problem.

Proposition 2. *The filter $L(z)$ is defined as*

$$|L|^2 = |M|^2 |S|^2 |W|^2 \frac{1}{\Phi_u}, \quad \forall \omega \in [-\pi, \pi]. \quad (4.28)$$

Proof. Note that the criterion defined in (1.1), for this controller architecture is now defined as

$$J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}}) = \left\| \left(\frac{P(z) (C_e(z, \theta_e) + C_{\bar{y}}(z, \theta_{\bar{y}}))}{1 + P(z) (C_e(z, \theta_e) + C_y(z, \theta_y))} - M(z) \right) W(z) \right\|_2^2$$

and it can be written as

$$J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{P(e^{j\omega}) (C_e(e^{j\omega}, \theta_e) + C_{\bar{y}}(e^{j\omega}, \theta_{\bar{y}}))}{1 + P(e^{j\omega}) (C_e(e^{j\omega}, \theta_e) + C_y(e^{j\omega}, \theta_y))} - M(e^{j\omega}) \right|^2 |W(e^{j\omega})|^2 d\omega$$

Chapter 4. Data-driven algorithms for multirotor systems

or, by dropping the argument $e^{j\omega}$:

$$J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{P(C_e(\theta_e) + C_{\bar{y}}(\theta_{\bar{y}}))}{1 + P(C_e(\theta_e) + C_y(\theta_y))} - M \right|^2 |W|^2 d\omega. \quad (4.29)$$

After some manipulations and exploiting the definition of $M(z)$ in (4.27), the criterion (4.29) can be rewritten as:

$$\begin{aligned} J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}}) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{P}{1 + P(C_e(\theta_e) + C_y(\theta_y))} \right|^2 (C_{\bar{y}}(\theta_{\bar{y}}) + C_e(\theta_e)) \\ &\quad - (\bar{C}_{\bar{y}} + \bar{C}_e) - M \left[(C_e(\theta_e) + C_y(\theta_y)) \right. \\ &\quad \left. - (\bar{C}_e + \bar{C}_y) \right] \Big|^2 |W|^2 d\omega. \end{aligned} \quad (4.30)$$

If the involved signals in (4.25) are realisations of stationary and ergodic stochastic processes and $N \rightarrow \infty$, the analysis of $J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}})$ is based on asymptotic results:

$$J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}}) \rightarrow J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}}). \quad (4.31)$$

Exploiting the definition of $M(z)$ in (4.27) and the Parseval theorem (see [36]), the asymptotic criterion (4.31) is written as:

$$\begin{aligned} J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}}) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{|L|^2 |P|^2}{|M|^2} \left| (C_{\bar{y}}(\theta_{\bar{y}}) + C_e(\theta_e)) \right. \\ &\quad - (\bar{C}_{\bar{y}} + \bar{C}_e) - M \left[(C_e(\theta_e) + C_y(\theta_y)) \right. \\ &\quad \left. - (\bar{C}_e + \bar{C}_y) \right] \Big|^2 \Phi_u d\omega. \end{aligned} \quad (4.32)$$

In the following, $J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}})$ is used instead $J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}})$. The main idea of the VRFT algorithm is to minimise $J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}})$ instead of $J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}})$ as described in Section 2.1. If the ideal controllers belong

4.2. Controller with feed-forward and action on plant output

to the class of the available controllers, that is

$$(\bar{C}_e(z), \bar{C}_y(z), \bar{C}_{\bar{y}}(z)) \in \{(C_e(z, \theta_e), C_y(z, \theta_y), C_{\bar{y}}(z, \theta_{\bar{y}}))\},$$

regardless of how the plant, the filters and the reference model are selected, the VRFT method leads to the ideal controller (see [32]). On the other hand, if the ideal controllers do not belong to the class of the available controllers, in order to obtain $J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}}) = J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}})$, a specific selection of the filter $L(z)$ must be made. Indeed, if

$$|L|^2 = \frac{|M|^2|W|^2}{|1 + P(C_e(\theta_e) + C_y(\theta_y))|^2} \frac{1}{\Phi_u}, \quad \forall \omega \in [-\pi, \pi], \quad (4.33)$$

then $J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}}) = J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}})$ and minimising $J_{VR}(\theta_e, \theta_y, \theta_{\bar{y}})$ is the same as minimising $J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}})$.

Since the definition of $L(z)$ in (4.33) depends on $P(z)$, that is unknown, the choice in (4.33) is not feasible. As described in [32] where an exhaustive analysis is proposed, the following assumption can be made:

$$|1 + P(C_e(\theta_e) + C_y(\theta_y))|^2 \simeq |1 + P(\bar{C}_e + \bar{C}_y)|^2$$

and the filter in (4.33) can be rewritten as in (4.28) that it is here proposed again:

$$|L|^2 = |M|^2|S|^2|W|^2 \frac{1}{\Phi_u}, \quad \forall \omega \in [-\pi, \pi].$$

□

All the steps to exploit the VRFT method with the new controller architecture and data collected in open-loop are summarised in Algorithm 6. If the experimental data is collected in closed-loop, the improvements described in Section 4.1 must be implemented in the same way also with this architecture. Finally, if the plant output is affected by noise, an instrumental

Chapter 4. Data-driven algorithms for multirotor systems

Algorithm 6 VRFT algorithm with the new controller architectures.

- 1: Compute $U(z)$ such that $|U(e^{j\omega})|^2 = \Phi_u(\omega)$.
- 2: Compute $L(z) = M(z)S(z)W(z)U^{-1}(z)$.
- 3: Compute $u_L(t)$ as $u_L(t) = L(z)u(t)$.
- 4: Compute $\varphi_L(t) = \begin{bmatrix} \beta_{\tilde{y}}(z)M(z)^{-1} \\ \beta_e(z)(M(z)^{-1} - 1) \\ \beta_y(z) \end{bmatrix} L(z)y(t)$.
- 5: **if** $y(t)$ is affected by noise **then**
- 6: Identify the plant model $\hat{P}(z)$.
- 7: Compute $\hat{y}(t) = \hat{P}(z)u(t)$.
- 8: Compute the instrumental variable $\zeta(t) = \begin{bmatrix} \beta_{\tilde{y}}(z)M(z)^{-1} \\ \beta_e(z)(M(z)^{-1} - 1) \\ \beta_y(z) \end{bmatrix} L(z)\hat{y}(t)$.
- 9: **else**
- 10: Compute $\zeta(t) = \varphi_L(t)$
- 11: **end if**
- 12: Compute $\hat{\theta}_N^{IV} = \left[\sum_{t=1}^N \zeta(t)\varphi_L^T(t) \right]^{-1} \sum_{t=1}^N \zeta(t)u_L(t)$.

variable method must be implemented as in the standard VRFT algorithm presented in Section 2.1.

4.3 Correlation based Tuning for cascade control systems

Although the VRFT algorithm was extended in [19] to deal with a cascade control system with a single set of experimental data, the same extension of the CbT method was not available in the literature. This makes the CbT method inapplicable to tune the controllers presented in Chapter 3 and its comparison with VRFT impossible. In this section this extension is presented also for the CbT method, following the rationale of Section 2.1.1.

Also in this case two loops are considered without loss of generality (Figure 2.2). Let $M_i(z)$ and $M_o(z)$ be the reference models for the inner and the outer loop respectively. Experimental data

$$D_N = \{u(t), \tilde{y}_i(t), \tilde{y}_o(t)\}_{t=1, \dots, N}$$

from an open-loop test is considered available, where $u(t)$ is the control variable, $\tilde{y}_i(t)$ is the output of the inner loop and $\tilde{y}_o(t)$ is the output of the

4.3. Correlation based Tuning for cascade control systems

outer loop. Consider two families of controllers for the inner and the outer loops:

$$\begin{aligned} \mathcal{C}_i(\theta_i) &= \{C_i(z, \theta_i), \theta_i \in R_i^n\} \\ \mathcal{C}_o(\theta_o) &= \{C_o(z, \theta_o), \theta_o \in R_o^n\}. \end{aligned}$$

The inner controller can be tuned by applying the standard CbT method as presented in Section 2.2, setting $\tilde{y}(t) = \tilde{y}_i(t)$ and $\bar{y}(t) = u(t)$.

Considering the outer controller, the same problem of the VRFT approach arises: the input of the system to control is not available in the data. This signal must be build starting from the available information. The procedure to obtain this signal follows the same steps presented in Section 2.1.1 and it is here proposed again for convenience.

Let $\bar{y}_i(t)$ be the output signal of the outer controller. Since the inner and the outer controllers are designed independently, it can be derived from the available data. In detail, once the inner controller $C_i(z, \theta_i)$ is tuned, the reference signal of the inner loop can be computed as

$$\bar{y}_i(t) = e_i(t) + y_i(t),$$

where the tracking error comes from the result of the inner design as

$$e_i(t) = C_i^{-1}(z, \hat{\theta}_i)u(t),$$

where $\hat{\theta}_i$ are the optimal parameters of the inner loop. All the information is now available and the outer loop can be tuned by means of the CbT algorithm as presented in Section 2.2, imposing $\tilde{y}(t) = \tilde{y}_o(t)$ and $\bar{y}(t) = \bar{y}_i(t)$. The steps of the tuning method are presented in Algorithm 7.

Chapter 4. Data-driven algorithms for multirotor systems

Algorithm 7 The CbT method for two nested cascade control loops with a single set of experimental data.

- 1: Compute $U_i(z)$ such that $|U_i(e^{j\omega})|^2 = \Phi_u(\omega)$.
 - 2: **repeat**
 - 3: Compute $F_i(z) = (1 - M_i(z)) W_i(z) U_i^{-1}(z)$.
 - 4: Compute $\bar{u}_F(t)$ as $\bar{u}_F(t) = F_i(z)u(t)$.
 - 5: Choose l_i close to the impulse response of $M_i(z)$.
 - 6: Compute $\varsigma_i(t) = [u_F(t+l) \ \dots \ u_F(t) \ \dots \ u_F(t-l)]^T$.
 - 7: Compute the error $\varepsilon_i(t, \theta_i) = M_i(z)u(t) - C_i(z, \theta_i)(1 - M_i(z))\tilde{y}_i(t)$.
 - 8: Compute $f_i(\theta_i) = \frac{1}{N} \sum_{t=1}^N \varsigma_i(t)\varepsilon_i(t, \theta)$.
 - 9: Compute $J_i(\theta_i) = f_i^T(\theta) f_i(\theta)$.
 - 10: Compute $\hat{\theta}_i = \arg \min_{\theta_i} J_i(\theta_i)$.
 - 11: **until** $C_i(z, \hat{\theta}_i)$ is a minimum phase system, otherwise change $M_i(z)$.
 - 12: Compute $\bar{y}_i(t) = C_i^{-1}(z, \hat{\theta}_i)u(t) + \tilde{y}_i(t)$.
 - 13: Compute $U_o(z)$ such that $|U_o(e^{j\omega})|^2 = \Phi_{\bar{y}_i}(\omega)$ where $\Phi_{\bar{y}_i}(\omega)$ is the spectral density of $\bar{y}_i(t)$.
 - 14: Compute $F_o(z) = (1 - M_o(z)) W_o(z) U_o^{-1}(z)$.
 - 15: Compute $\bar{y}_{iF}(t)$ as $\bar{y}_{iF}(t) = F_o(z)\bar{y}_i(t)$.
 - 16: Choose l_o close to the impulse response of $M_o(z)$.
 - 17: Compute $\varsigma_o(t) = [\bar{y}_{iF}(t+l) \ \dots \ \bar{y}_{iF}(t) \ \dots \ \bar{y}_{iF}(t-l)]^T$.
 - 18: Compute the error $\varepsilon_o(t, \theta_o) = M_o(z)\bar{y}_i(t) - C_o(z, \theta_o)(1 - M_o(z))\tilde{y}_o(t)$.
 - 19: Compute $f_o(\theta_o) = \frac{1}{N} \sum_{t=1}^N \varsigma_o(t)\varepsilon_o(t, \theta)$.
 - 20: Compute $J_o(\theta_o) = f_o^T(\theta) f_o(\theta)$.
 - 21: Compute $\hat{\theta}_o = \arg \min_{\theta_o} J_o(\theta_o)$.
-

CHAPTER 5

Considered multirotor platforms

THREE different multirotor systems have been adopted in this work to test the data-driven methods presented in the previous chapters. All the multirotor systems considered here are quadrotors which differ in dimensions, actuation and control architectures. Indeed, as it can be more clear in the following sections where each quadrotor is presented in details, the data-driven algorithms are applied to a large, a medium and a micro quadrotor and to UAVs with variable and fixed blade pitch. This is made both to test if the data-driven algorithms are flexible and to provide a different way to tune the controllers for all the UAVs in our laboratory. Exploiting the different features of each UAV and the chance to execute specific tests, different analysis have been performed. The results will be presented in Chapter 6.

Chapter 5. Considered multirotor platforms

5.1 Aermatica P2-A1

The first quadrotor studied in this work is the Aermatica P2-A1 prototype (see Figure 5.1), a platform having a maximum take-off weight of about 5 kg and an arm length of 0.415 m. The four rotors have a radius of 0.27 m and a teetering articulation with flapping motion partially restrained by rubber elastic elements. Unlike most quadrotors, which use variable rotor angular rates as control inputs (with fixed rotors blade pitch), Aermatica P2-A1’s rotors are operated at a fixed angular rate and use variable collective pitch as control variables. While this choice leads to a more complex design of the rotor hub and a slight weight penalty, it has been shown (see, *e.g.*, [15,45]) that variable pitch control can overcome the limitations on the achievable quadrotor performance associated with the bandwidth of motor dynamics for rate-controlled configurations.

All the experiments considered in this work with this quadrotor have been conducted operating the UAV on a single degree of freedom test-bed (only pitch rotation allowed, see Figure 5.1): as discussed in [44], this set-up guarantees safer, faster and more repeatable operations with respect to flight while remaining representative of the pitch attitude dynamics in flight for near hovering conditions. Indeed, the test bench brings the vehicle rotors at a height from ground sufficient to assure Out of Ground Effect (OGE) (see [34]) aerodynamic conditions. Also, aerodynamic interferences on the rotors caused by the test-bed structure can be considered negligible. Furthermore, identification experiments for the attitude dynamics of Aermatica P2-A1 have been carried out both on the test bench and in flight. The models obtained from indoor data and flight data have the same structure and nearly identical numerical values of the parameters (see, again [44] for details).

As for the level of disturbances experienced by the platform on the test bench, it is worth to remark that with respect to an outdoor flight condition, in which the rotor-induced wakes develop free from obstacles, working indoor in a closed volume with limited dimensions implies a complex recirculation of rotor wakes. This, in turn, determines a non-negligible and

5.1. Aermatica P2-A1



Figure 5.1: *Aermatica P2-A1 on laboratory test-bed.*

non-deterministic disturbance on the vehicle during the test, with effects assimilable to free air turbulence. As a result, the platform is subjected to a significant level of aerodynamic disturbance also during indoor tests.

Concerning the control architecture, the Aermatica P2-A1 platform adopts a classical attitude control scheme based on decoupled cascaded PID loops for the pitch, roll and yaw axes (see the block diagram in Figure 3.3, where the pitch control loop is represented). More precisely, an outer PD loop based on attitude feedback and an inner PID loop on angular rate feedback. The overall delay of the control loop, from IMU measures, through acquisition and processing, to servo actuation of blade collective pitch, is estimated to be 0.06 s.

Due to the size of the quadrotor and the lack of customisation of the firmware, only few data-driven methods have been considered. In particular, all the tests to collect the experimental data are performed in quasi open-loop conditions: while the nominal attitude and position controllers were disabled, a supervision task enforcing attitude limits during the experiment was left active (maximum attitude excursion guaranteed from adopted test-bed is $\pm 20^\circ$). As it will be illustrated in Section 6.1, the excitation procedure was already implemented on the quadrotor and each time an excita-

Chapter 5. Considered multirotor platforms

tion test is performed, a new Pseudo Random Binary Sequence (PRBS) is built starting from some excitation signal parameters (*e.g.*, amplitude and min/max switching interval). The user can modify this signals parameters but he can not impose a pre-computed excitation signal. Due to this limitation the instrumental variable in the VRFT method can not be built starting from a second experiment since the two experiments must share the same excitation signal as described in Section 2.1.

VRFT and CbT algorithms have been tested on this UAV. In particular both data-driven methods exploit a single experimental dataset to tune the cascade attitude control system as described in Sections 2.1 and 4.3. The results are then compared with a manual tuning and a model-based tuning from a previous work on the Aermatica P2-A1 platform (reported in [47]) concerning the development of a dedicated design procedure for the attitude control laws, combining a black-box model identification step, followed by a robust control law design step. In particular a LTI state-space SISO model for the dynamics of the pitch angular rate was obtained using a subspace identification method (see [42, 46]). The model-based tuning of the Aermatica P2-A1 pitch attitude control law (see [47] for details) was carried out using a structured H_∞ synthesis approach: for the assigned controller structure and the above-discussed back-box model for pitch dynamics, the procedure finds the locally optimal parameters for the two PID controllers so as to satisfy the imposed closed-loop stability and performance requirements (*e.g.*, crossover frequency of each loop in a specified bandwidth, attitude angle set-point tracking behavior defined in terms of target response time and maximum steady-state error, process noise disturbance rejection capability specified assigning a maximum gain constraint profile as function of frequency). It is interesting to point out that the above requirements for H_∞ synthesis were defined in order to obtain an improvement in terms of wind gust rejection compared to the standard tuning obtained through the trial and error empirical procedure done manually, that presently does not guarantee a fully satisfactory performance in flight. On the contrary, as the actual quadrotor performance in terms of set-point tracking is considered adequate, the optimal tuning requirements were defined in order to

5.2. Proto-1

replicate the standard tuning, as a benchmark. The standard tuning was also used as starting guess for the optimisation procedure. Finally, changing the SNR in the experimental data, a sensitivity analysis to SNR is performed comparing the VRFT and the CbT algorithms.

5.2 Proto-1

The second quadcopter used in this work was entirely developed at Politecnico di Milano (see [22]) based on the following requirements:

- Frame configuration: X-quadrotor (this configuration allows future upgrades such as an on board camera).
- Frame dimensions: medium size (450mm to 550 mm as distance between opposite motors).
- Overall weight: less than 2 kg.
- Flight time: about 10 minutes.
- Payload: at least 500 g.

Off-the-shelf components were selected to satisfy these requirements. The assembled quadrotor helicopter is shown in Figure 5.2 and Figure 5.3.

The FCU, that manages the communication between the sensors and the generated code in order to control the quadcopter, was implemented on the Rapid Robot Prototyping (R2P) boards [6]. These boards are an open source hardware and software framework that enables the rapid development of robotic applications. These boards provide different functions, in particular, the Inertial Measurement Unit (IMU) module provides the main functions of attitude estimation and attitude control, the USB module provides serial communication with a computer or other serial devices and at last the RC module allows the Pulse Width Modulation (PWM) communication with the motors controllers. The modules use a publish/subscribe architecture to communicate between each others. The control portion is implemented in *Simulink* and compiled to C++ code. The FCU implements the same control architecture of the Aermatica P2-A1 UAV with decoupled cascaded PID loops as it is illustrated in Figure 3.3.

Chapter 5. Considered multirotor platforms

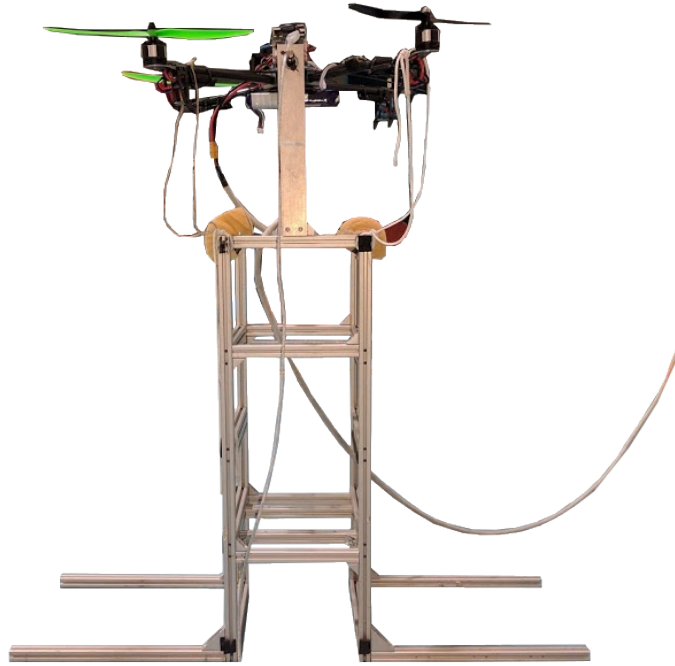


Figure 5.2: *Proto-1 on its test-bed: front view.*



Figure 5.3: *Proto-1 on its test-bed: top view*

5.2. Proto-1

The quadcopter also holds a *RaspberryPi 2* board used to interface the R2P modules with a Robotic Operating System (ROS) network but this functionality was unused for the tests. In addition a small ultrasonic sensor has been mounted on the drone to measure the distance from the ground when landing.

Unlike the Aermatica P2-A1 UAV, a fixed-pitch actuation scheme was retained and it exploits variable rotor angular rates as control inputs.

All the tests on the quadrotor were performed on a test-bed that constrains all translational degrees of freedom as well as the roll and yaw motions. Only the pitch rotation is left unconstrained (see [14]). This ensures that the tests are repeatable and safe and that an erroneous choice of the controller parameters will not send the system crashing through the room. The test-bed is built out of *x-frame* aluminium rods and weighted with sacks of concrete. The upper part of the frame has a smooth rod resting on ball bearings at each extremity for frictionless rotation. The quadrotor is then securely fastened to this rod. In the current mounting scheme the rod passes as close as possible to the centre of mass of the system in order to interfere as little as possible with the dynamics of the quadrotor. Because of the physical configuration of the system there is a small distance between the rod and the actual centre of mass. In turn, this causes the system to act like a very small pendulum and the test-bed adds some damping when the system quadrotor achieves higher pitch angles. In practice this damping is negligible for small oscillations. The test-bed holds the quadcopter high enough that ground effect disturbances are avoided however, since the test takes place in a closed space some re-circulation of rotor wakes occurs. This represents a discrepancy when compared to outdoor flight conditions where the rotor wakes develop free from obstacles. Even so, it has been shown in [22] that such a test-bed is representative of actual attitude dynamics in flight.

As will be illustrated in Section 6.2, only the VRFT method was considered with this UAV in order to show how the data-driven method is able to operate also on a different test case. The results are compared with a model-based structured H_∞ approach developed in [22].

Chapter 5. Considered multirotor platforms

5.3 ANT-1

As it was presented in Section 4.1, an extension for the VRFT method is proposed to deal with closed-loop experimental data collected in flight. This extension can not be tested with the previous UAVs due to their dimensions. Indeed the Italian Civil Aviation Authority (ENAC) allows the outdoor flights without restriction only for remotely piloted vehicles that weight less then 300 g (see [18]). Furthermore, the laboratory was recently equipped with the motion capture technology that, using an array of infra-red sensitive cameras, provides the position and the attitude of the UAV inside an indoor flight arena.

Both to comply with the ENAC standard and to facilitate the indoor flight, a new UAV has been developed. In particular, the new quadrotor, called ANT-1, belongs to the class of Micro Aerial Vehicles (MAVs) and respects the following requirements:

- Maximum take-off weight: less than 300 g.
- Flight time: at least 10 minutes.
- Reduced geometric dimension: an inter axis smaller than 200 mm.

As for the quadrotor presented in Section 5.2, off-the-shelf components are employed to build the UAV after a preliminary analysis performed with some optimisation methods. The assembled quadrotor is shown in Figure 5.4.

The development of a MAV opens the world of UAVs to some interesting possibilities: the creation of aerial collective systems able to fly in cluttered environments such as cities or inside buildings. By working together, multiple flying vehicles can perform a given task quicker and more efficiently than a single system. In fact, multiple UAVs can share computing, sensing and communication payload so that they result faster and quicker than a unique, large UAV. Additionally, they can cover a wider area than a single aerial vehicle when flying outdoors. Thus, it is clear that aerial collective systems have a huge potential in terms of application as monitoring of toxic clouds and meteorological conditions, security and artistic shows.

5.3. ANT-1

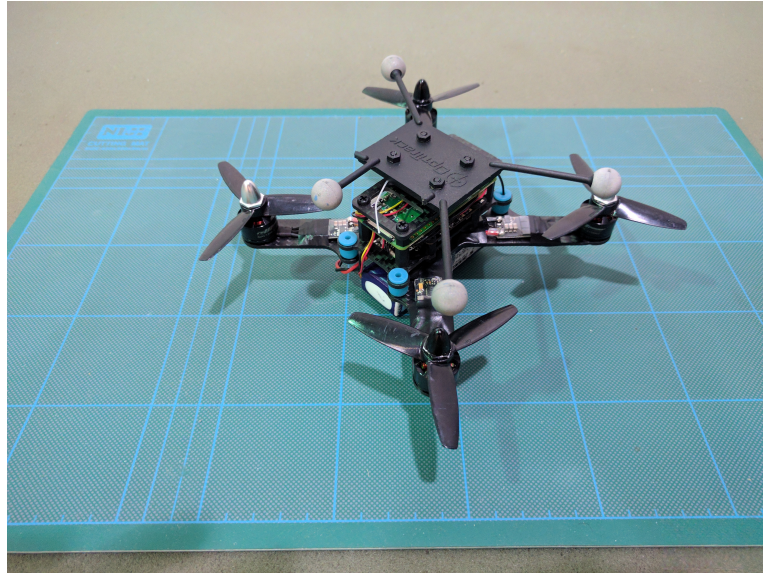


Figure 5.4: *The ANT-1 quadrotor.*

A high level of customisation is required for the FCU because specific tests must be performed. For this reason the *Pixfalcon* FCU has been selected. It is a low cost and an open autopilot shield, suitable for remotely controlled vehicles such as quadrotors and fixed wing aircraft. It is equipped with a 3 axes accelerometer, a 3 axes gyroscope, a magnetometer and a pressure sensor. Pixfalcon has 8 channels of PWM outputs. Additionally GPS can be mounted extending the autopilot capabilities. The firmware that runs on the Pixfalcon board is the *PX4 Pro Autopilot*: it is an open-source software, fully compatible with the Pixfalcon board and constantly under development. It is able to estimate the attitude and the position of the UAV and it supports the information that comes from the optical motion capture system. It contains also the controller code and, in particular, it implements the scheme presented in Figure 3.4 and Figure 3.5.

The Pixfalcon board is connected to a *Raspberry Pi Zero W* board which is a small and light single-board computer. It is responsible for the communication to and from the UAV. It connects to the laboratory Local Area Network (LAN) through WI-FI, allowing ANT-1 to be controlled from the ground station. Moreover, due to its computational power, it is going to

Chapter 5. Considered multirotor platforms



Figure 5.5: *The ANT-1 quadrotor on the test-bed.*

have a primary role in collaborative flight when ANT-1 will be involved in MAV swarming.

The high flexibility of the ANT-1 platform allows to test all the data-driven algorithms proposed in this work. First, the tests executed with the other UAVs are repeated. The test-bed that was built for the quadrotor presented in Section 5.2 is also exploited for ANT-1 (see Figure 5.5). Furthermore, after the attitude controller is tuned, the position controller is also considered for this quadrotor. As will be explained in Chapter 6, a specific test is developed to reach this goal and it must be performed in flight. Finally, the VRFT algorithm is employed with data that come from a closed-loop experiment in flight and the result is compared with the one obtained with the same data-driven method but exploiting open-loop experimental data.

CHAPTER 6

Results

IN this chapter the simulation and the experimental results obtained applying the data-driven methods are illustrated. In particular, Section 6.1 contains the results on the Aermatica P2-A1 quadrotor and two analyses have been performed: first the VRFT tuning is compared with a model-based and manual tuning, then the sensitivity of the VRFT and the CbT methods to SNR is examined. In Section 6.2 the second quadrotor is considered. With this UAV only the VRFT algorithm is employed and it is compared with a structured H_∞ method. Finally, Section 6.3 contains the results with the ANT-1 quadrotor. A complete analysis is accomplished with this UAV and it includes both experimental and simulated results, both for the position and the attitude controllers and using open-loop and closed-loop experiments.

Chapter 6. Results

6.1 Aermatica P2-A1

Considering the Aermatica P2-A1, only the pitch attitude controller is taken into account and all the experiments have been conducted operating the UAV on a single DoF test-bed as described in Section 5.1. To make the obtained results as representative as possible, during the experiments all four rotors are working, with a base collective pitch command of 60% that guarantees a total thrust equal to the vehicle weight (hovering), and only the pitch attitude controller is enabled. The remaining 40% is shared between the DoFs controllers for maneuvering as follows: 10% increment/decrement equal on each rotor for climb/descent, 15% used differentially (plus and minus) on opposite rotors for roll/pitch control and 15% used differentially on clockwise and counterclockwise pairs of rotors for yaw control. This determines that the nominal saturation limit for the pitch control variable is equal to 30%.

As described in Section 5.1, the PRBS excitation signal is applied in quasi open-loop conditions and during the tests the pitch angular velocity and the pitch angle, measured by the on-board IMU, are logged with sampling time equal to 0.02 s, together with the control variable (see Figure 6.1). The measurement signals come from the Kalman-based estimator implemented in the FCU that employs only on-board sensors (*e.g.*, accelerometer and gyroscope).

This UAV has been employed to perform two types of tests: the first one compares the tuning obtained with the VRFT method with a model-based tuning and a manual tuning, while in the second analysis the VRFT and the CbT algorithms are compared and a discussion on how the noisy data affect both the data-driven methods is provided.

6.1.1 VRFT setpoint tracking and load disturbance evaluation

The model-based tuning of the Aermatica P2-A1 pitch attitude control law (see [44] for details) was carried out using a structured H_∞ synthesis approach: for the assigned controller structure as presented in Figure 3.3, the procedure finds the (locally) optimal parameters for the two PID controllers

6.1. Aermatica P2-A1

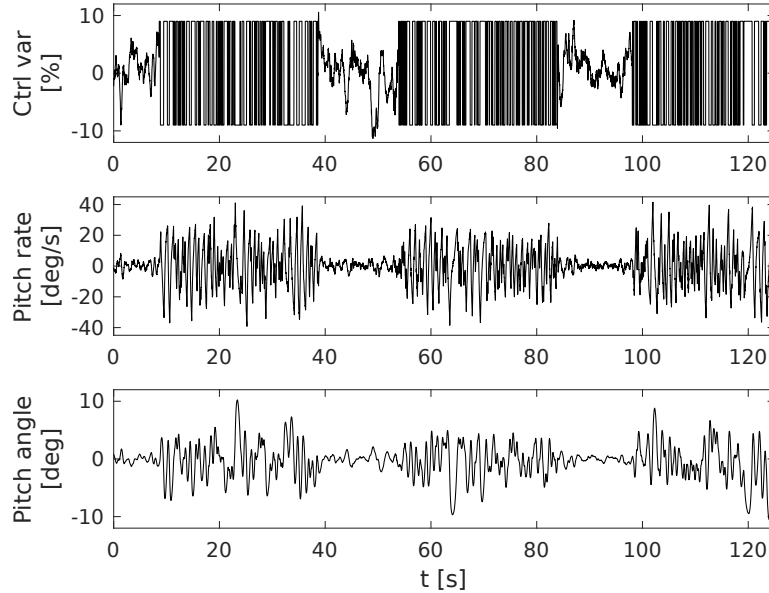


Figure 6.1: *Aermatica P2-A1: open-loop experimental dataset used by VRFT and CbT.*

so as to satisfy the imposed closed-loop stability and performance requirements (e.g., crossover frequency of each loop in a specified bandwidth, attitude angle setpoint tracking behavior defined in terms of target response time and maximum steady-state error, process noise disturbance rejection capability specified assigning a maximum gain constraint profile as function of frequency). It is interesting to point out that the above requirements for H_∞ synthesis were defined in order to obtain an improvement in terms of wind gust rejection compared to the standard tuning obtained through the trial and error empirical procedure done manually, that presently does not guarantee a fully satisfactory performance in flight. On the contrary, as the actual quadrotor performance in terms of setpoint tracking is considered adequate, the optimal tuning requirements were defined in order to replicate the standard tuning, as a benchmark. The standard tuning was also used as starting guess for the optimisation procedure.

Two types of tests have been performed:

- setpoint tracking evaluation: a desired pitch angle command history was assigned manually by the operator, with step amplitudes of 5 deg and 10 deg;

Chapter 6. Results

- load disturbance rejection evaluation: in order to simulate on the test-bed the effect of a wind gust, a rope was fixed at the tip of the front (or back) vehicle arm, with a weight of 0.7 kg at the end. The operator can act manually on the weight in order to engage/disengage its effect, applying and maintaining the disturbance torque for about 10 s and then suddenly releasing it. A null angular setpoint is required throughout the operations.

As discussed in Section 2.1, the VRFT method is essentially a model-reference approach, so from the user’s perspective the main degree of freedom in the design procedure is the selection of the reference model. As explained in Chapter 1, the design of the inner reference model $M_i(z)$ and the outer reference model $M_o(z)$ requires some prior knowledge about the system to be controlled. In the present study, the results of previous work on model identification for the Aermatica P2-A1 platform (see [44]) provided significant insight in the definition of the structure for the reference models. In particular, the reference models $M_i(z)$ and $M_o(z)$ for, respectively, the inner and the outer control loop, have been defined on the basis of available requirements for the desired bandwidth and damping factor of the inner and outer complementary sensitivity functions. More precisely, the desired bandwidth of the inner loop is set to 25 rad/s while the one of the ideal outer loop is of 20 rad/s. $M_i(z)$ and $M_o(z)$ have been defined as second order systems with a damping ratio of 0.7 and a time delay of 3 samples (corresponding to 0.06 s and representing the overall delay of the control loop, as mentioned in Section 5.1). The reference models of the two control loops used to compute the virtual reference signals are:

$$M_i(z) = \frac{0.09833z + 0.07778}{z^2 - 1.32z + 0.4966} \frac{1}{z^3}$$

$$M_o(z) = \frac{0.06609z + 0.05481}{z^2 - 1.45z + 0.5712} \frac{1}{z^3}.$$

In Figure 6.2 the Bode diagram of both the models is displayed. As for the weighting functions $W_i(z)$ and $W_o(z)$ defining, respectively, the model reference cost function (1.1) for the inner and the outer loops, they have

6.1. Aermatica P2-A1

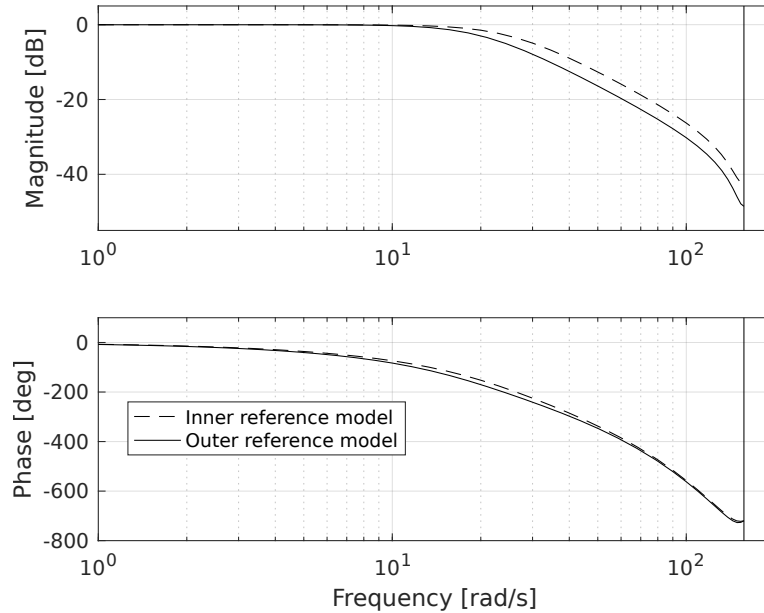


Figure 6.2: *Aermatica P2-A1: Bode diagram of the inner and the outer reference models.*

been chosen as $W_i(z) = 1$ and $W_o(z) = 1$.

Since the experimental data is affected by noise, as already mentioned, the VRFT algorithm used in this work implements an instrumental variable method to counteract the effect of noise (see Section 2.1). The instrumental variable is constructed through the identification of an ARX(15,15) model for the inner and the outer loops.

In Table 6.1 the resulting parameters for both the outer loop PD and the inner loop PID controllers are listed (see again Figure 3.3 that illustrates the controller scheme), as obtained with the VRFT approach. The standard tuning obtained through the manual procedure and the one from structured H_∞ synthesis are also reported. Figures 6.3, 6.4 and 6.5 show the setpoint tracking test of the manual, the model-based H_∞ and the VRFT tuning. As can be seen from the figures, the H_∞ approach provides the best performance whereas the VRFT method performs better than the manual approach. This confirms that the VRFT approach can lead to a satisfactory closed-loop performance level starting from a limited prior knowledge about the plant.

Chapter 6. Results

Controller parameter	Standard tuning	H_∞ tuning	VRFT tuning
K_P^O	9.26	4.7314	5.6364
K_D^O	1.11	0.8453	0.3683
K_P^I	0.257	0.3297	0.4979
K_I^I	0.643	1.6186	2.0685
K_D^I	0.0231	0.0079	0.0111

Table 6.1: *Aermatica P2-A1: optimal controller parameters considering manual, VRFT and H_∞ methods.*

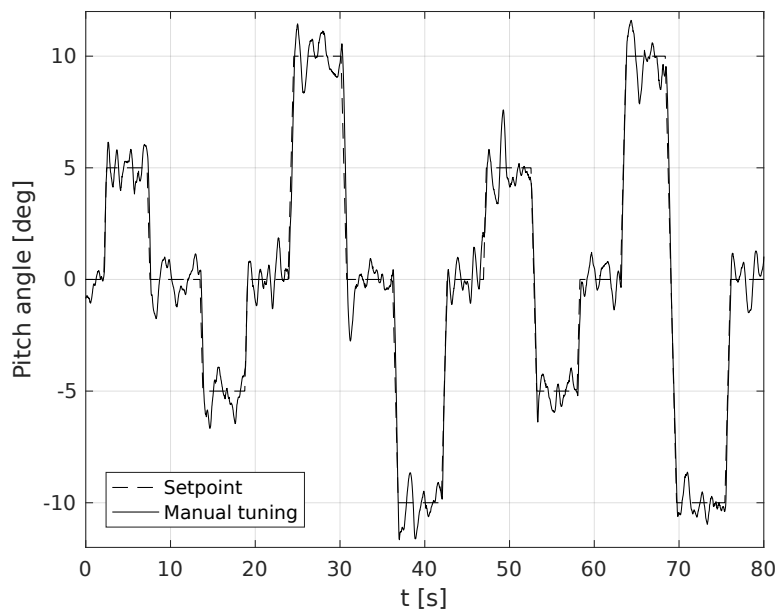


Figure 6.3: *Aermatica P2-A1: setpoint tracking with manual tuning (experiment).*

6.1. Aermatica P2-A1

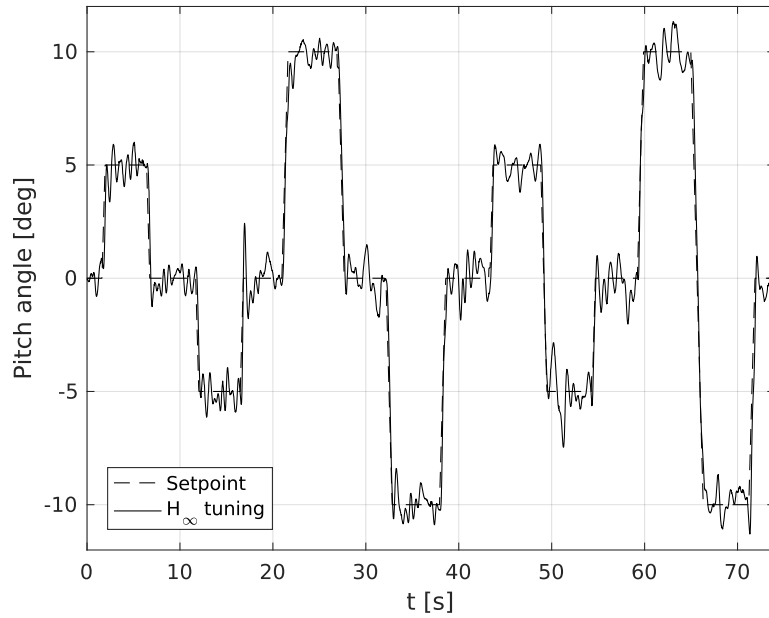


Figure 6.4: Aermatica P2-A1: setpoint tracking with model-based H_∞ tuning (experiment).

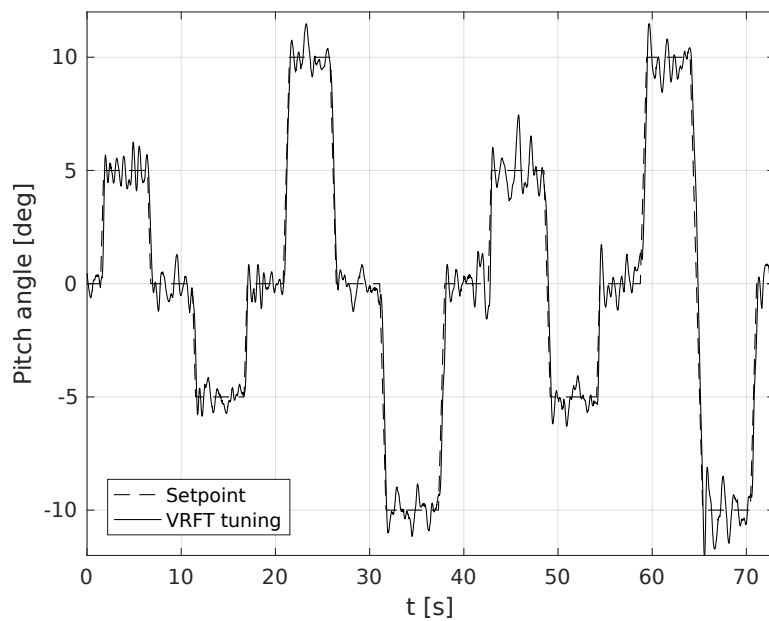


Figure 6.5: Aermatica P2-A1: setpoint tracking with VRFT tuning (experiment).

Chapter 6. Results

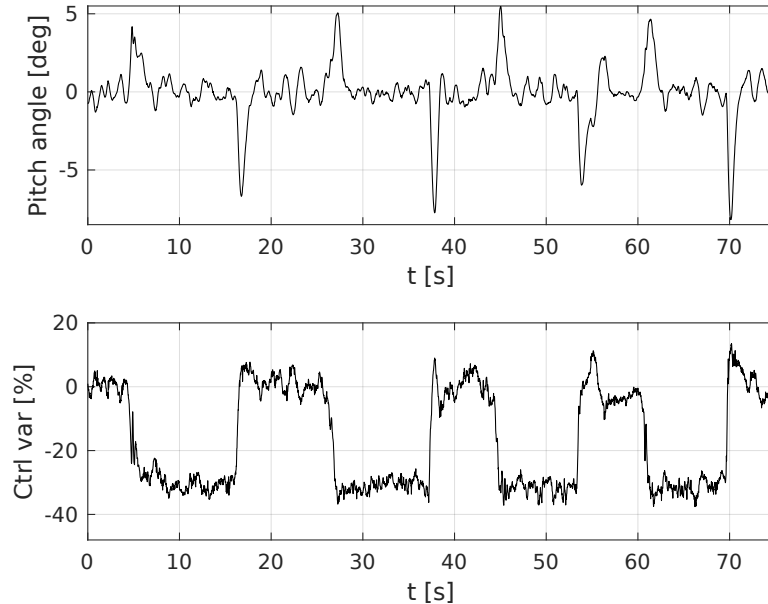


Figure 6.6: *Aermatica P2-A1: load disturbance rejection with manual tuning (experiment).*

When considering the load disturbance rejection test (Figures 6.6, 6.7 and 6.8), it is worth to observe that, for all the considered tuning set, the control variable overcomes the saturation limit: the control allocation routine gives priority to roll/pitch attitude degrees of freedom, allowing an enlarged margin on demand by subtracting it from the yaw and vertical degrees of freedom (in this order). Hence the applied load disturbance is representative of a heavy wind gust, testing the limit of vehicle capabilities. The performance of the H_∞ and VRFT controllers is similar and represents a significant improvement with respect to the one of the manually tuned controller. As can be seen from the figures, however, this comes with a cost in terms of increased control effort with respect to the standard method.

In conclusion, it has been shown that the VRFT algorithm can be successfully applied to tune a cascade control system and that the designed controller provides a performance level comparable with the one of a model-based H_∞ controller. In particular, the data-driven controller presents good tracking and disturbance rejection capabilities and therefore represents a viable solution for the fast deployment of high performance attitude con-

6.1. Aermatica P2-A1

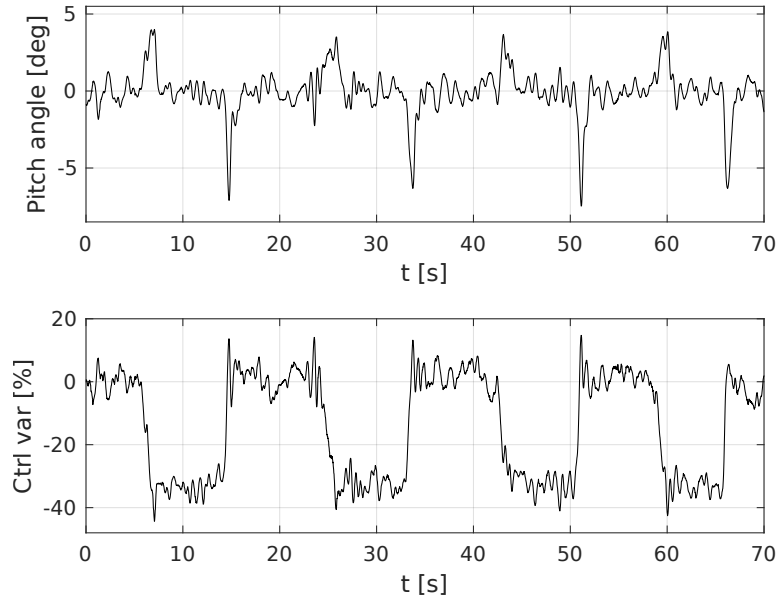


Figure 6.7: *Aermatica P2-A1: load disturbance rejection with model-based H_∞ tuning (experiment).*

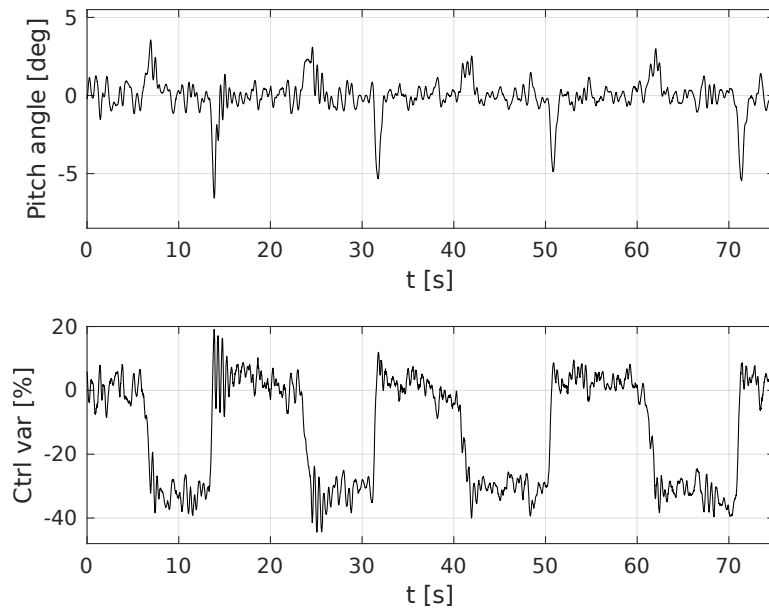


Figure 6.8: *Aermatica P2-A1: load disturbance rejection with VRFT tuning (experiment).*

Chapter 6. Results

trollers for this platform.

6.1.2 CbT and sensitivity to SNR analysis

Since the VRFT method is successfully applied to the Aermatica P2-A1 quadrotor, also the CbT approach is considered. In particular, the extension of the CbT presented in Section 4.3 to deal with cascade control systems is used.

The reference models of the two loops have been modified with respect to the previous section because the experimental setup slightly changed. The reference models has been defined as second order systems with a damping ratio of 0.7, a time delay of 3 samples and a desired bandwidth of 24 rad/s (inner) and 16 rad/s (outer):

$$M_i(z) = \frac{0.09151z + 0.07308}{z^2 - 1.346z + 0.5107} \frac{1}{z^3}$$

$$M_o(z) = \frac{0.04397z + 0.03786}{z^2 - 1.557z + 0.6389} \frac{1}{z^3}.$$

In Figure 6.9 the Bode diagram of both the models is displayed. For simplicity the weighting functions are $W_i(z) = 1$ and $W_o(z) = 1$.

In order to apply the CbT approach, the parameter l in (2.10) has to be selected. As explained in Section 2.2, this parameter should be close to the length of the impulse response of the reference model (see Figure 6.10) ($l_i = 20, l_o = 35$).

For what concerns the VRFT approach, the order of the ARX model, which provided satisfactory results, is $p = 15$, both for the inner and outer loop.

First the VRFT and the CbT are compared. Both the setpoint tracking and the load disturbance rejection evaluations are performed as explained in the previous section. The CbT and the VRFT tunings are reported in Table 6.2 both for the outer loop PD and the inner loop PID controller parameters. Figures 6.11 and 6.12 show the setpoint tracking tests of the CbT and the VRFT tuning. As can be seen from the figures, both approaches provide a satisfactory closed-loop performance level even starting from a

6.1. Aermatica P2-A1

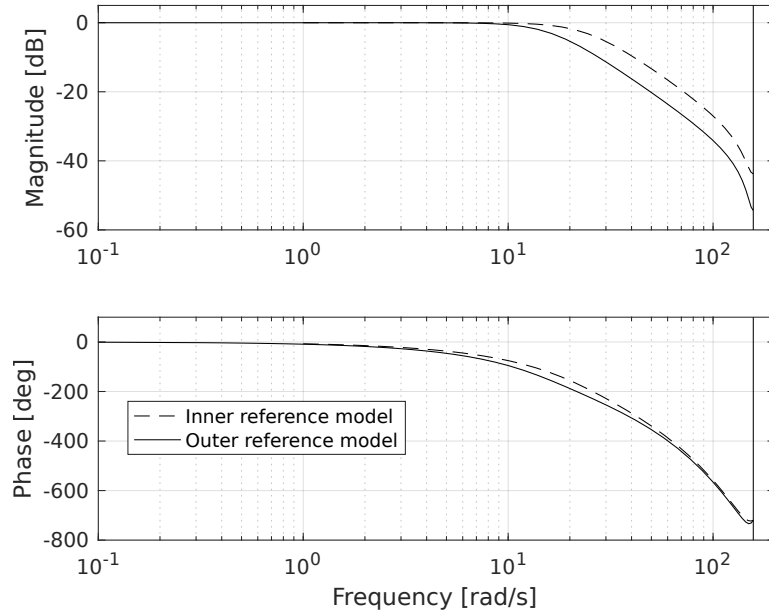


Figure 6.9: Aermatica P2-A1: Bode diagram of the inner and the outer reference models.

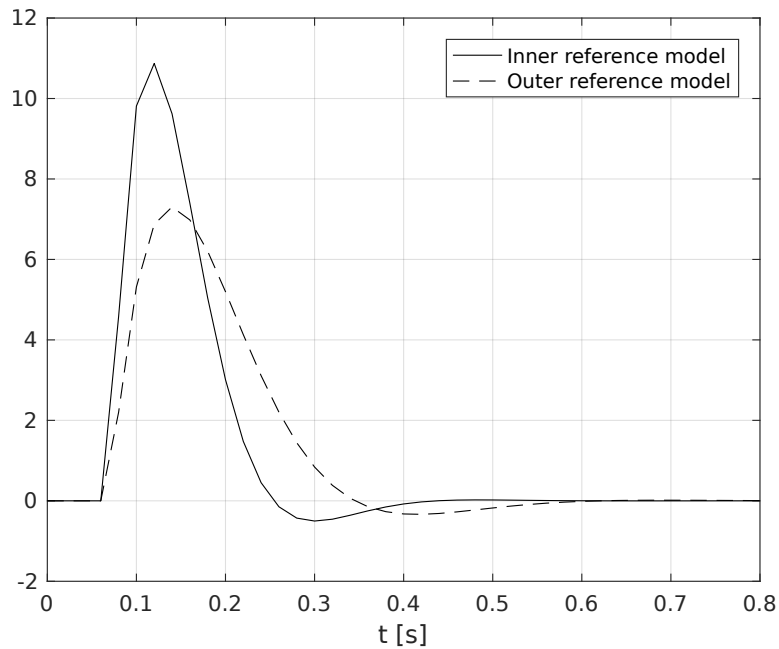


Figure 6.10: Aermatica P2-A1: impulse response of the inner and the outer reference models.

Chapter 6. Results

Table 6.2: *Aermatica P2-A1: optimal controller parameters considering VRFT and CbT.*

Controller parameter	CbT tuning	VRFT tuning
K_P^O	6.3194	5.5726
K_D^O	0.1751	0.2775
K_P^I	0.4786	0.5369
K_I^I	1.79	1.9782
K_D^I	0.0097	0.0094

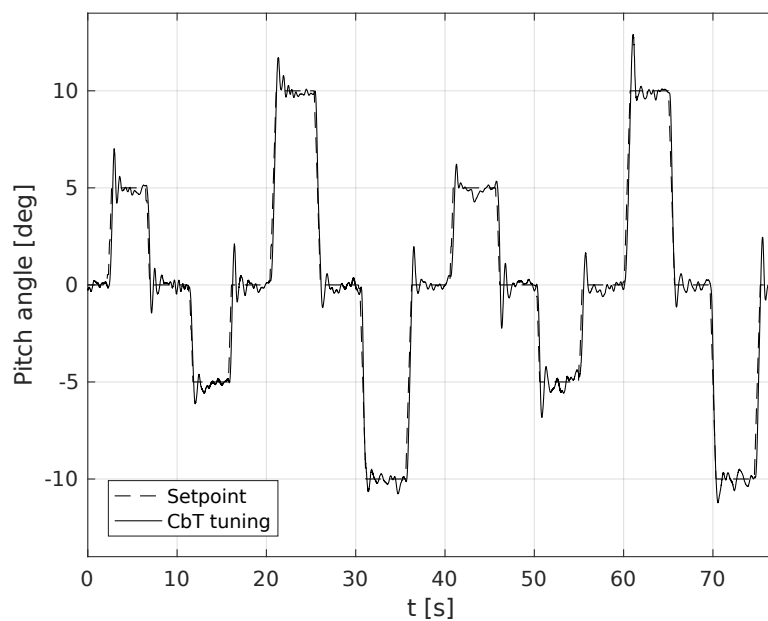


Figure 6.11: *Aermatica P2-A1: setpoint tracking with CbT tuning (experiment).*

limited prior knowledge about the plant.

When considering the load disturbance rejection test (Figures 6.13 and 6.14), the performance of the methods is analogous both considering the perturbed pitch angle and the control effort.

Now the behaviour of the VRFT and of the CbT approach are discussed with respect to the influence of noisy data. When the SNR is low, the use of instrumental variables through the identification procedure for the VRFT can lead to destabilising controllers even if the order of the ARX model is large. On the contrary, the CbT algorithm is more robust and the parameter

6.1. Aermatica P2-A1

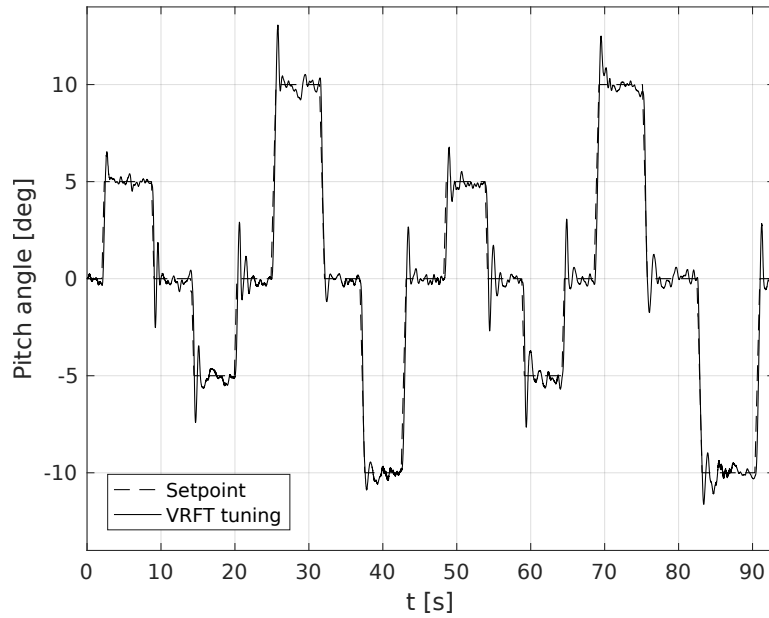


Figure 6.12: *Aermatica P2-A1: setpoint tracking with VRFT tuning (experiment).*

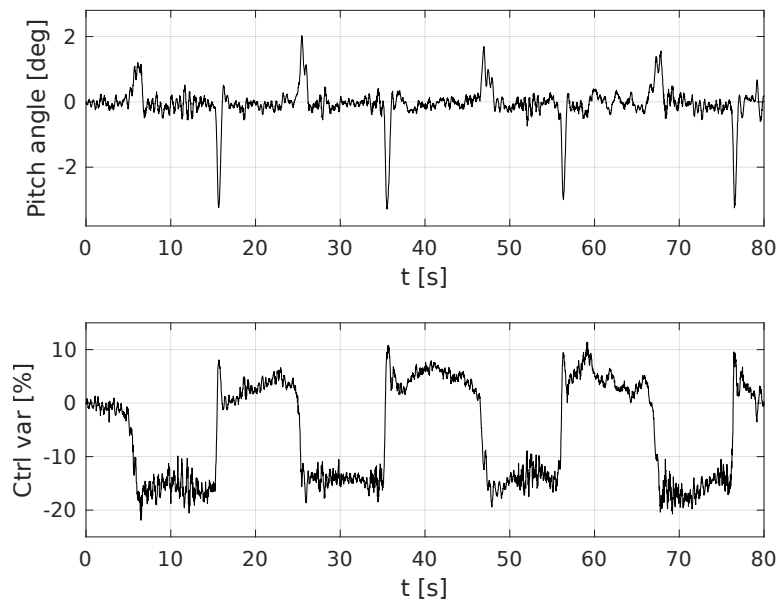


Figure 6.13: *Aermatica P2-A1: load disturbance rejection with CbT tuning (experiment).*

Chapter 6. Results

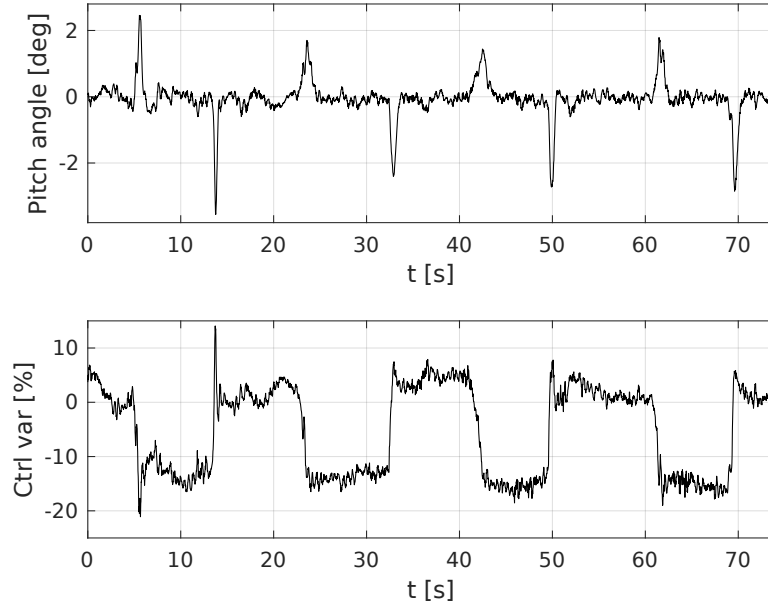


Figure 6.14: *Aermatica P2-A1: load disturbance rejection with VRFT tuning (experiment).*

Table 6.3: *Aermatica P2-A1: MSE of CbT and VRFT considering three different datasets.*

PRBS amplitude [%]	[-13,13]	[-11,11]	[-9,9]
CbT	0.7978	0.8305	0.8349
VRFT	0.6759	0.6894	X

l represents a trade-off between accuracy and bias. These considerations have been highlighted by running three tests where PRBS sequences of different amplitude are employed to tune the controller parameters with both algorithms. In the following, $l_i = 20$, $l_o = 35$ and $p = 15$ have been considered. Since the statistical properties of the noise do not change over the experiments, increasing the input amplitude is equivalent to raising the SNR. Each test consists of ten realisations of the setpoint tracking experiment shown in Figure 6.11. The mean values of the Mean Square Error (MSE) (between setpoint and measured pitch angle) are reported in Table 6.3. As can be seen in Table 6.3, the CbT algorithm gives similar results in all the tests and it performs slightly worse than the VRFT method. The differences between the methods are due to the use of different instrumental

6.2. Proto-1

variables. In case the ARX model employed in VRFT perfectly identifies the model dynamics, the basic instrument of such a method is known to be the most statistically efficient choice. However, in case of overparameterisation, overfitting could occur, thus leading to a performance worse than that given by the extended instrument of the CbT method. This is very likely when the SNR is particularly low, and in this case the VRFT approach even yields a destabilizing controller. It should be noted that a different choice of l can lead to better performance of the CbT algorithm. For instance, as far as the dataset with the highest SNR is concerned, a larger value of l ($l_i = 200, l_o = 350$) can be reasonably employed and the computed mean MSE is 0.5912.

In conclusion, the two methods lead to similar results in terms of set-point tracking and disturbance rejection but CbT is more robust than VRFT in the presence of a low level of SNR.

6.2 Proto-1

In order to understand if the VRFT method can be easily adopted to tune a different platform, a second quadrotor is considered. It is smaller than the Aermatica P2-A1 UAV and, furthermore, it has a different actuation architecture. As described in Section 5.2, all the experiments are made with a single DoF test-bed. Thus only the pitch attitude controller is considered. The structure of this regulator is displayed in Figure 3.3.

The PRBS is selected also in this case as excitation signal and the experiments are performed in open-loop conditions. The FCU works at 100 Hz. Figure 6.15 shows the excitation signal, the measured pitch angular rate and the measured pitch angle $\vartheta(t)$. The measurement signals come from the Kalman-based estimator that employs only on-board sensors (*e.g.*, accelerometer and gyroscope).

For the VRFT method, the following closed-loop reference models for

Chapter 6. Results

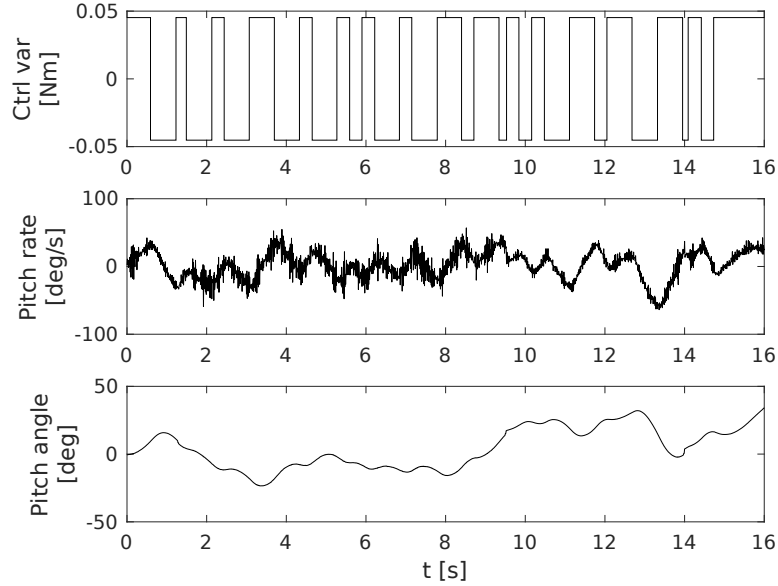


Figure 6.15: *Proto-I: open-loop experimental dataset used by VRFT.*

the inner and the outer control loop are considered:

$$M_i(z) = \frac{0.1221z - 0.1162}{z^2 - 1.86z + 0.8659} \quad (6.1)$$

$$M_o(z) = 10^{-4} \frac{7.81z + 7.625}{z^2 - 1.929z + 0.9305}. \quad (6.2)$$

More precisely, $M_i(z)$ and $M_o(z)$ are second order systems with a desired bandwidth of 8 rad/s and 4 rad/s respectively. The damping ratio of both $M_i(z)$ and $M_o(z)$ is 0.9. As for the weighting functions $W_i(z)$ and $W_o(z)$ defining, respectively, weights for the inner and the outer loops, they have been chosen as $W_i(z) = 1$ and $W_o(z) = 1$. To deal with noisy data, an instrumental variable is employed and it is built through the identification of an ARX(7,17) model for the inner and the outer loops.

Running the VRFT procedure on both the inner and outer loops with the input-output data shown in Figure 6.15 and the reference models (6.1) and (6.2) leads to the controller parameters shown in Table 6.4. Also shown are the parameters for the pre-existing H_∞ controller (see [22] where both stability constraints and performance requirements are discussed) to be

6.2. Proto-1

Table 6.4: *Proto-1: optimal controller parameters considering H_∞ and VRFT methods.*

Controller parameter	H_∞ tuning	VRFT tuning
K_P^O	2.0	1.6057
K_D^O	0.00522	0.0
K_P^I	0.298	0.2978
K_I^I	0.304	0.514
K_D^I	0.0499	0.0

Table 6.5: *Proto-1: average MSE of H_∞ and VRFT methods computed on 10 setpoint tracking tests.*

Tuning method	MSE
VRFT	6.3384
H_∞	10.719

used as a point of reference.

6.2.1 Setpoint tracking evaluation

To validate the controllers, a test sequence with steps of increasing amplitude was generated and this signal was fed as a set a pitch angle setpoint to the quadcopter. The test was repeated 10 times using both the VRFT and the H_∞ tuned controllers and the average MSE, comparing the measured pitch angle to the setpoint, was computed. Note that the average MSE gives a valuation on the controller setpoint tracking performance but the VRFT and the H_∞ methods employ different cost functions to obtain the optimal tunings. The values of the average MSE for the VRFT and H_∞ tuned controllers are shown in Table 6.5. The MSE of the VRFT controller is a little smaller than that of the H_∞ controller. It can be explained by the slightly more oscillatory nature of the H_∞ -tuned controller.

The setpoint tracking performance of the quadcopter with, respectively, the VRFT and H_∞ controllers during one, randomly chosen, run is shown in Figures 6.16 and 6.17.

Chapter 6. Results

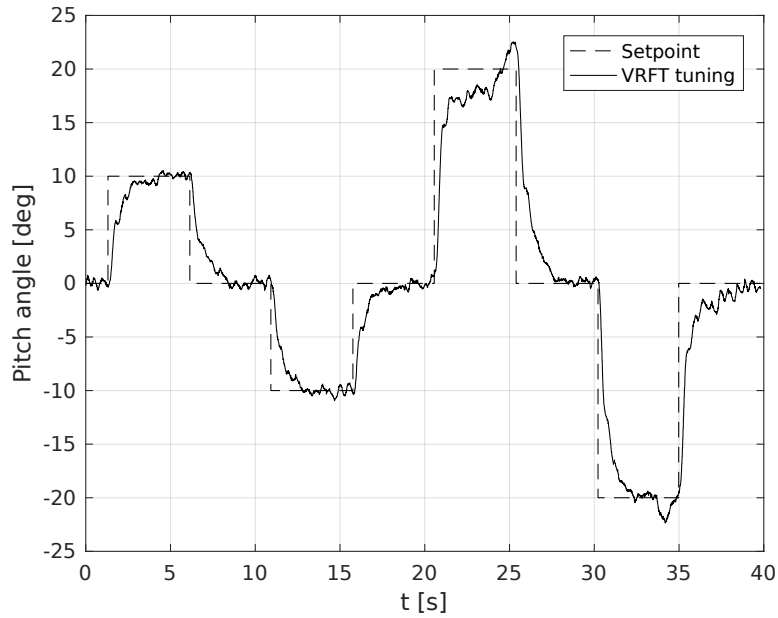


Figure 6.16: Proto-I: setpoint tracking performance with VRFT tuning (experiment).

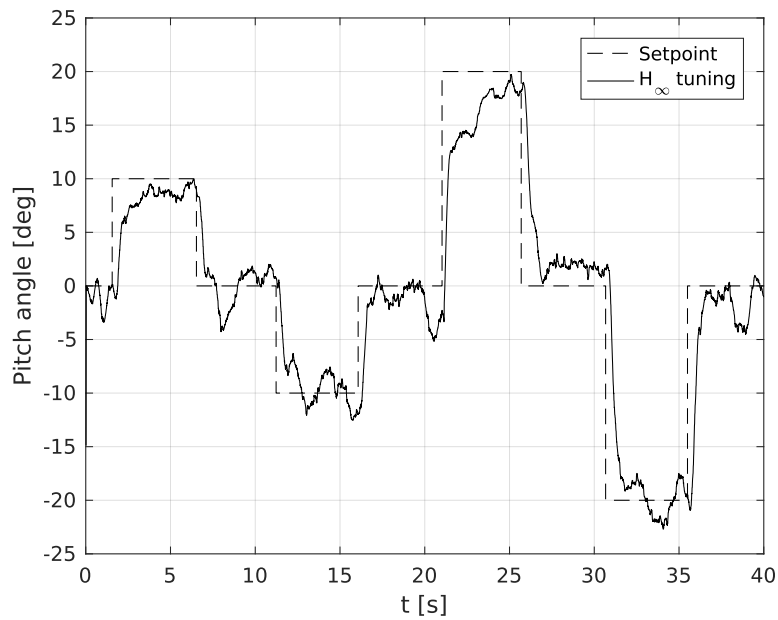


Figure 6.17: Proto-I: setpoint tracking performance with H_{∞} tuning (experiment).

6.2. Proto-1

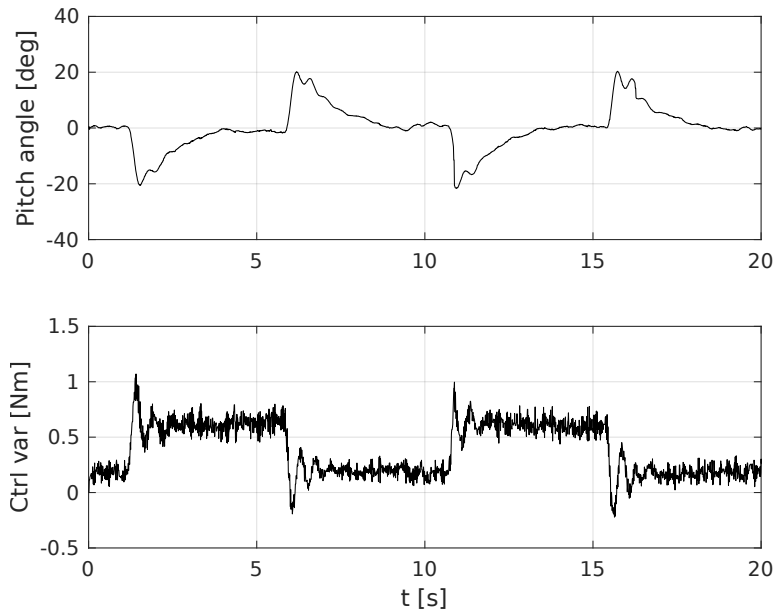


Figure 6.18: Proto-1: load disturbance rejection with VRFT tuning (experiment).

6.2.2 Load disturbance rejection evaluation

The disturbance rejection property is tested using a similar method. The firmware of the quadcopter provides a method to introduce a step disturbance on the speed of the motors. This was used to repeatedly reduce by 10% the lift generated by the motors on the front of the quadcopter. The test was repeated 10 times with both the VRFT and H_∞ tuned controllers and the average of the MSE was computed. The disturbance rejection performance of the quadcopter with, respectively, the VRFT and H_∞ controllers during one, randomly chosen test, are shown in Figures 6.18 and 6.19.

It is immediately apparent that the VRFT-tuned controller offers significant improvements to the steady state error. The H_∞ -tuned controller settles with a steady state error of several degrees whereas the VRFT-tuned controller achieves zero state error. In addition, the control effort required by both controllers is quite similar even if the VRFT-tuned controller has slightly higher peaks.

The improvement is reflected in the steady-state error of the two controllers as shown in Table 6.6. The MSE of the VRFT-tuned controller is

Chapter 6. Results

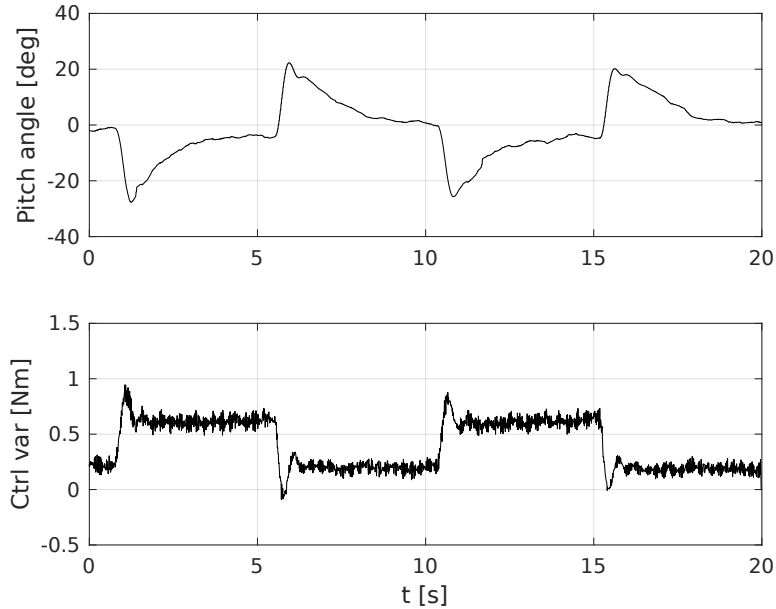


Figure 6.19: Proto-1: load disturbance rejection with H_∞ tuning (experiment).

Table 6.6: Proto-1: average MSE of H_∞ and VRFT methods computed on 10 load disturbance rejection tests.

Tuning method	MSE
VRFT	4.7908
H_∞	5.5686

slightly lower than that of the H_∞ controller.

6.3 Ant-1

The data-driven methods are applied also on a third UAV: the ANT-1 quadrotor (see Section 5.3). Thanks to its size and the high customisation of its FCU, the data-driven algorithms have been employed to tune both the pitch attitude controller and the longitudinal position controller. Furthermore, in order to better understand the results also a simulation environment is exploited where a plant model is exploited to better illustrate the results obtained with the experimental data. In order to display the capability of VRFT to tune a MIMO control system, roll and pitch controllers are tuned

6.3. Ant-1

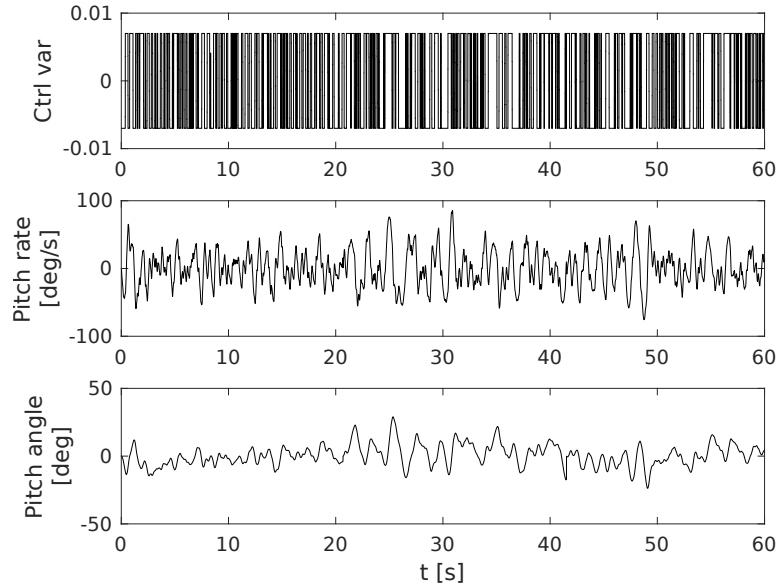


Figure 6.20: ANT-1: open-loop experimental dataset used by data-driven methods.

at the same time in Section 6.3.3. Finally, closed-loop experimental data collected in flight have been used to tune the pitch attitude controller.

6.3.1 Pitch attitude controller with open-loop experiments

In the following only the pitch attitude controller is considered with data collected exploiting the test-bed setup that constrains all translational and rotational degrees of freedom, except for pitch rotation. The excitation test has been performed in open-loop. The same setup was used with the two previous UAVs.

Figure 6.20 shows the input and output signals used for data-driven tuning. The entire dataset has been obtained by combining three subsequent open-loop tests of 20 seconds. The input signal is a PRBS pitching moment, which has been applied in open-loop conditions, thus with the nominal attitude and position controllers deactivated. The control variable $\mathcal{M}(t)$ is expressed as percentage of the maximum total thrust. The PRBS amplitude and the time duration of the switching intervals were defined so as to obtain a forcing spectrum large enough to excite the dominant pitch dynamics. The output signals, pitch rate and pitch angle, are measured by

Chapter 6. Results

the on-board IMU and recorded on a memory card at a sampling rate of 250 Hz. The measurement signals come from the Kalman-based estimator that employs only on-board sensors (*e.g.*, accelerometer and gyroscope).

Reference models

In order to define the closed-loop reference models, it is advisable to rely on a second-order model characterised by properly chosen natural frequency and damping ratio. Such choice allows to easily impose the value of the desired bandwidth, the unit static gain and, at least roughly, the desired damping of the closed-loop system. Nevertheless, such a simple model often does not guarantee a sufficiently high-performance controller tuning. Indeed, if some information on the plant dynamics is available, in addition to the knowledge of the controller structure, it is possible to augment the reference model so that it better matches the closed-loop behaviour of the real plant. The simplest property to include is the delay, but sometimes it is also necessary to change the numbers of poles and zeros, as will be shown below.

Before presenting the reference models, it is convenient to better describe the chosen controller families because the structure of the reference model depends also on the structure of the controller.

First, the controller architecture that was already implemented in the ANT-1 FCU is considered (see Figure 3.4). In Section 4.2, VRFT was extended to consider this control structure but the same extension is not applicable to the CbT and to the controller unfalsification methods. The extended VRFT algorithm is compared with the H_∞ controller considering the control architecture in Figure 3.4. The inner loop reference model is a second order model with 5 samples of delay, a desired bandwidth of 56 rad/s and a damping ratio of 0.33:

$$M_i(z) = z^{-5} \frac{0.0238z + 0.02265}{z^2 - 1.816z + 0.8626}. \quad (6.3)$$

In this specific case no filtering action was needed, thus the weighting function has been defined as $W_i(z) = 1$.

6.3. Ant-1

Unlike the inner loop, after a preliminary analysis, it is advisable to increase the complexity of the reference model by adding a suitable pole. Therefore, the resulting continuous time reference model is a second order system with a desired bandwidth of 45 rad/s and a damping ratio of 0.52 with an additional pole in 45 rad/s:

$$M_o(z) = z^{-5} \frac{0.0008861z^2 + 0.003231z + 0.0007374}{z^3 - 2.635z^2 + 2.333z - 0.6927}. \quad (6.4)$$

As before, a time delay of 5 samples has been added to the final discrete time transfer function. Finally, unlike before, the use of a weighting function proved to be useful for improving the output performance obtained through the VRFT tuning. In particular, $W_o(z)$ has been defined as a ninth-order lowpass digital *Butterworth filter*, with a cut-off frequency equal to 200 rad/s.

In order to compare VRFT, CbT and the controller unfalsification method, the control architecture implemented in the FCU is modified. In particular, the inner controller with the feedback action and the derivative term computed on the plant output can not be exploited. The new inner controller structure is based on SISO PID regulators as presented in Section 3.2 while the outer controller remains the same as in Figure 3.4: it is a proportional gain.

Since the controller architecture is changed, a new family of reference models is considered. The new family relies on a second-order model characterised by properly chosen natural frequency and damping ratio with additional poles/zeros. Table 6.7 summarises the characteristics of all the reference models designed for the VRFT, the CbT and the controller unfalsification tuning algorithms. Obviously, reference models must include, just like before, the time delay. It has been verified that the use of low-pass filters as weighting functions ensures a closer match between actual and desired behaviour within the frequency band of interest, ultimately leading to a significant improvement in output performance. Ninth-order lowpass digital Butterworth filters with cut-off frequencies of 500 and 100 rad/s were adopted for the inner and outer loop respectively. Notice that the controller

Chapter 6. Results

		ω [rad/s]	ξ	Added poles/zeros	Weighting function (Low-pass filter)
VRFT	Inner loop	56	0.6	Zero in 100 rad/s	$f_c = 500$ rad/s
	Outer loop	43	0.6	Pole in 70 rad/s	$f_c = 100$ rad/s
CbT	Inner loop	56	0.6	Zero in 100 rad/s	$f_c = 500$ rad/s
	Outer loop	36	0.8	Pole in 70 rad/s	$f_c = 100$ rad/s
Controller Unfalsification	Inner loop	56	0.6	Zero in 100 rad/s	-
	Outer loop	52	0.3	Pole in 70 rad/s	$f_c = 100$ rad/s

Table 6.7: *ANT-1: inner and outer reference models for VRFT, CbT and controller unfalsification algorithms.*

unfalsification method does not require any weighting function.

Looking to Table 6.7, different outer reference models for different data-driven methods should surprise the reader. It is easy to understand that the outer reference model heavily depends on the inner closed-loop system. As will be described below, where the results are presented, the performance of the CbT tuning method is overall lower than that obtained by the VRFT approach when the inner loop is considered. The outer reference model employed with CbT must consider the inner loop performance. Indeed the bandwidth of the outer reference model is lower than that one of the reference model of the VRFT method. Exploiting the VRFT outer reference model also for the CbT method leads to the instability of the outer loop.

On the contrary, the controller unfalsification method leads to better performance than VRFT in the inner loop. Thus, the bandwidth of the reference model is increased to allow better outer loop performances.

As described in Section 2.3, in order to apply the controller unfalsification method, the user must defined the reference model also for the desired control sensitivity function. This is mandatory to ensure the internal stabil-

6.3. Ant-1

	K_{FF}^I	K_P^I	K_I^I	K_D^I	K_P^O
VRFT	0	0.09143	0.2067	0.0016	12.11
H_∞	0	0.0849	0.2138	0.0014	11.75
Manual tuning	0	0.06	0.1	0.0010	8

Table 6.8: *ANT-1: optimal controller parameters considering manual, VRFT and H_∞ methods.*

	K_P^I	K_I^I	K_D^I	K_P^O
VRFT	0.05146	0.1935	0.0012	12.78
CbT	0.04696	0.06792	0.00091	12.92
Controller Unfalsification	0.06280	0.3303	0.0017	14.10

Table 6.9: *ANT-1: optimal controller parameters considering VRFT, CbT and controller unfalsification methods.*

ity of the closed-loop system. Following the conditions in Section 2.3, the model is defined as:

$$Q(z) = \frac{1.481z^2 - 2.943z + 1.462}{17.8z^2 - 31.77z + 13.97}.$$

Controller parameter values

The controller parameters obtained with the VRFT tuning applied to the control architecture in Figure 3.4 are summarised in Table 6.8. Table 6.9 shows the tunings generated by VRFT, CbT and controller unfalsification algorithms by adopting the SISO PID control architecture for the inner loop and the proportional gain for the output regulator. To deal with noisy data, an instrumental variable is employed and it is built through the identification of an ARX(5,5) model for the inner and the outer loops.

Chapter 6. Results

Simulation results

Before testing the controller on the real plant, some iterations in a simulation environment have been performed in order to speed-up the tuning process. It is straightforward that the more accurate the identified model is, the closer the simulation results will be compared to the experimental ones. Basically, the tuning process is intended to outperform the manual tuning and, in principle, achieve performance comparable with the pre-existing H_∞ controller. The reader should notice that the plant model is employed only to show the closed-loop results for both the inner and the outer loop and not to tune the controllers.

An accurate ANT-1 pitch attitude model has been identified in [12] by means of the black-box PBSID method. The resulting identified model $P(z)$ is a third-order dynamic system describing the relationship between pitch moment $\mathcal{M}(t)$ and pitch rate $q(t)$. Then, the pitch angle $\vartheta(t)$ is obtained by integrating the output of the above-defined model. So, as schematically shown in Figure 6.21, the attitude dynamics has been mod-

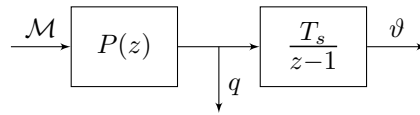


Figure 6.21: ANT-1: pitch attitude model.

elled by means of the cascade of two dynamic systems:

$$P(z) = z^{-5} \frac{0.2858z^2 - 0.2068z - 0.0781}{z^3 - 2.82z^2 + 2.643z - 0.8228} \quad (6.5)$$

and an integrator block, allowing the cascade control to be actually implemented in the simulation environment. The Bode diagram of the identified model $P(z)$ is shown in Figure 6.22.

Inner loop results. The main results achieved by closing the inner control loop will be shown below. More specifically, frequency domain and time domain results will be discussed.

First the tunings from the VRFT and the H_∞ methods are compared

6.3. Ant-1

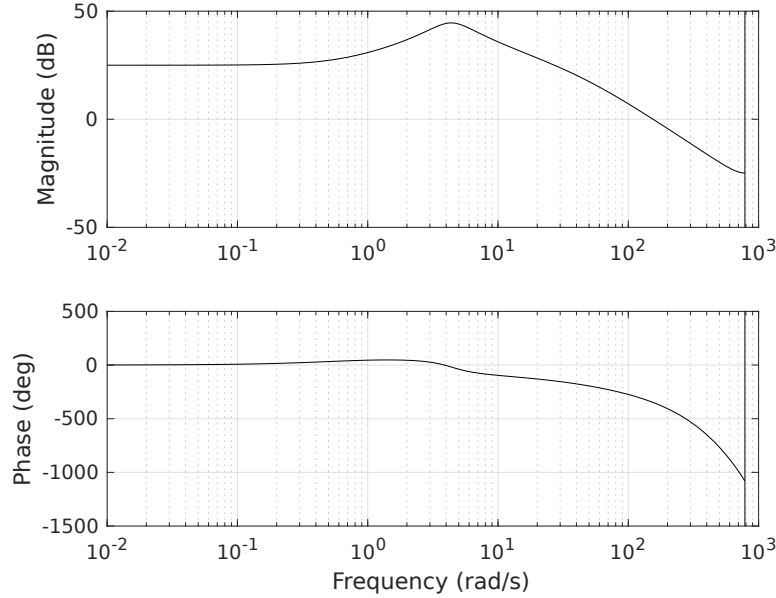


Figure 6.22: ANT-1: Bode diagram of the identified pitch attitude model.

considering the control architecture in Figure 3.4. In this case the reference models for the inner and the outer loops are selected as in (6.3) and (6.2) respectively. Figure 6.23 shows the Bode diagram obtained by the manual, the H_∞ and the VRFT tunings of the inner controller. As can be seen, the frequency response of the VRFT tuning is very close to that obtained with the H_∞ controller and approximates quite well the desired behaviour. The same remarks can be made by comparing the step responses of the different tunings (Figure 6.24). Furthermore, it is clearly visible that the manual tuning is significantly slower than both the VRFT and H_∞ controller tunings.

The VRFT, the CbT and the controller unfalsification methods are compared in Figure 6.25 and Figure 6.26. It is clearly evident that the VRFT tuning ensures a better performance than the CbT counterpart, as already shown in Section 6.1. The CbT approach is highly recommended whenever the gathered data is rather noisy. In this condition, in fact, the use of instrumental variables could be ineffective, therefore leading to an unstabilising controller. On the contrary, the CbT method maintains roughly constant performance as data SNR decreases. As a drawback, for suffi-

Chapter 6. Results

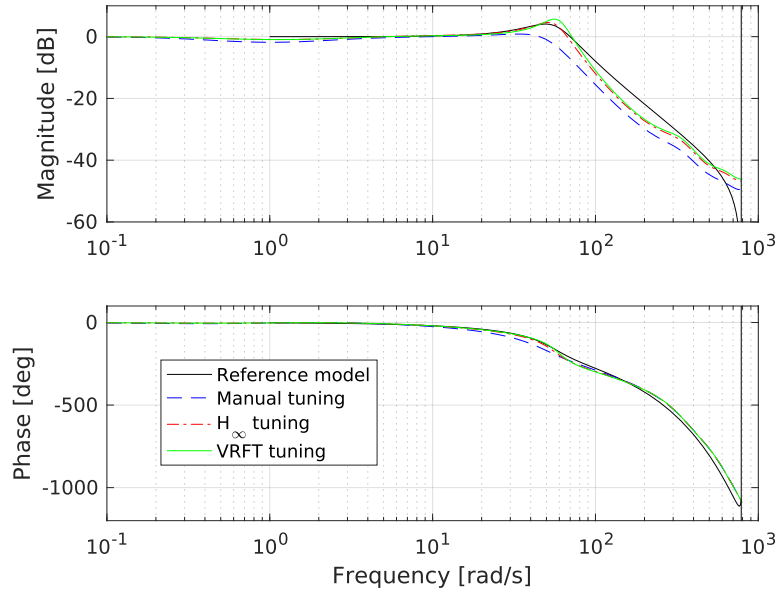


Figure 6.23: ANT-1: comparison of the inner loop Bode diagrams considering VRFT, H_∞ and manual tunings (simulation).

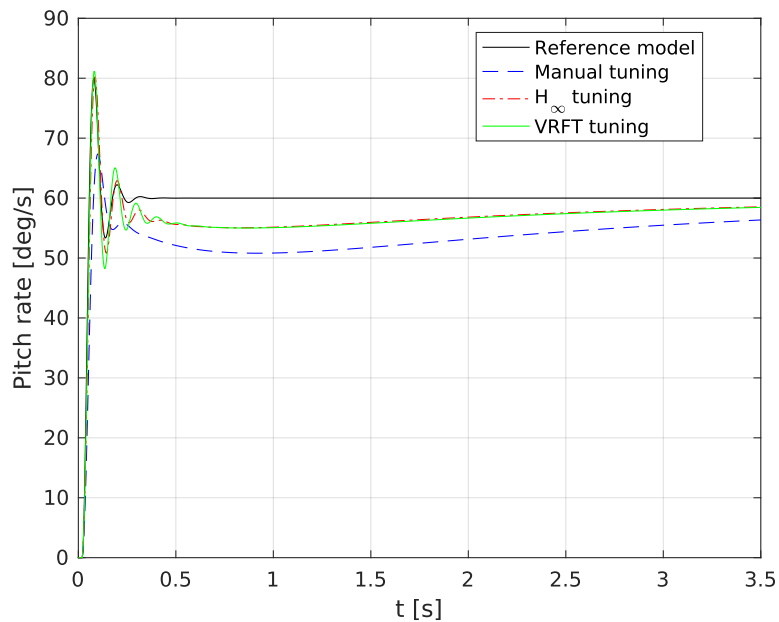


Figure 6.24: ANT-1: comparison of the inner loop step responses considering VRFT, H_∞ and manual tunings (simulation)

6.3. Ant-1

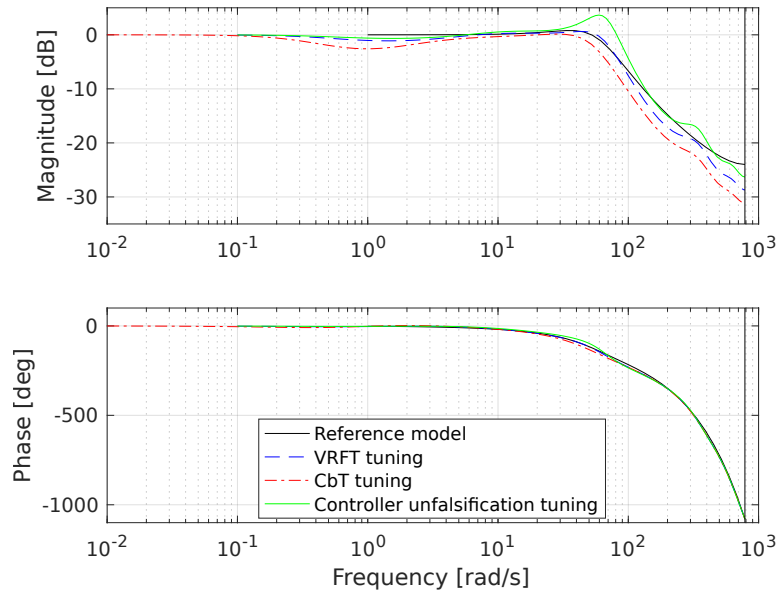


Figure 6.25: ANT-1: comparison of the inner loop Bode diagrams considering VRFT, CbT and controller unfalsification tunings (simulation).

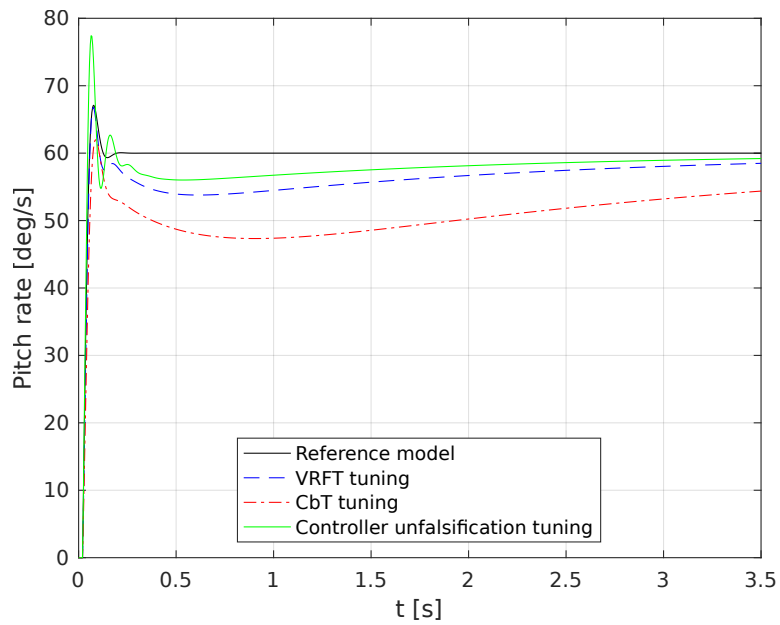


Figure 6.26: ANT-1: comparison of the inner loop step responses considering VRFT, CbT and controller unfalsification tunings (simulation).

Chapter 6. Results

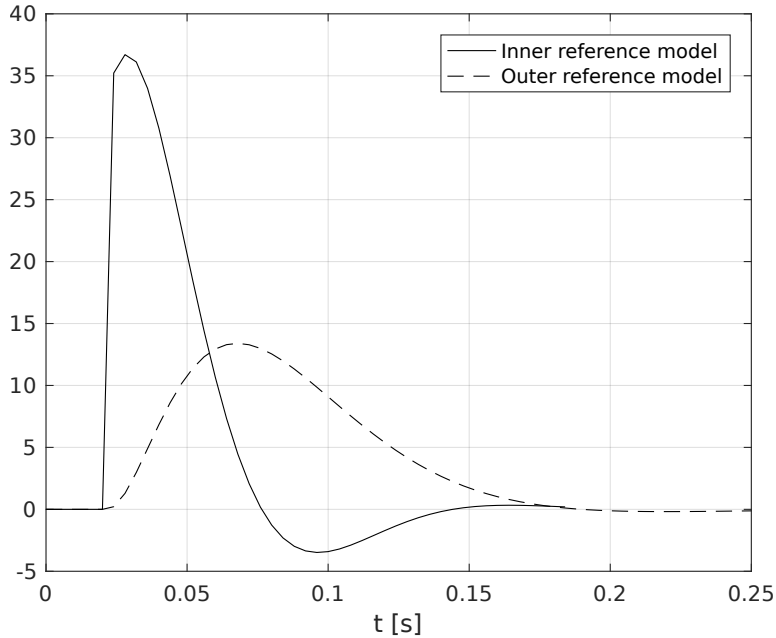


Figure 6.27: *ANT-1: impulse response of the inner and the outer reference models.*

ciently high SNR values the VRFT ensures quite better performance than the CbT. In our case, the experimental datasets are characterised by a high SNR, making it useless to rely on the most robust CbT algorithm. Due to the high SNR the parameter l of the CbT algorithm, representing the trade-off between accuracy and bias, has been set equal to ten times the length of the impulse response of the reference model (Figure 6.27). In the inner loop, l has been set equal to 500.

Considering the reference models for the inner loop in Table 6.7, the unfalsification algorithm did not have to execute any iteration, because even with $\delta = 1$ a stabilising controller is obtained. In order to show how the stability constraint works, an undoubtedly unachievable reference model has been imposed. Starting from the model defined in Table 6.7 a higher natural frequency ($\omega_n = 65$ rad/s) has been required and it has been verified that the VRFT algorithm yields a destabilising controller. To comply with the stability constraint, the algorithm progressively reduces the δ value, which means greater importance is given to control sensitivity with respect to the output one, until the stability test has passed. Figure 6.28 shows the gradual

6.3. Ant-1

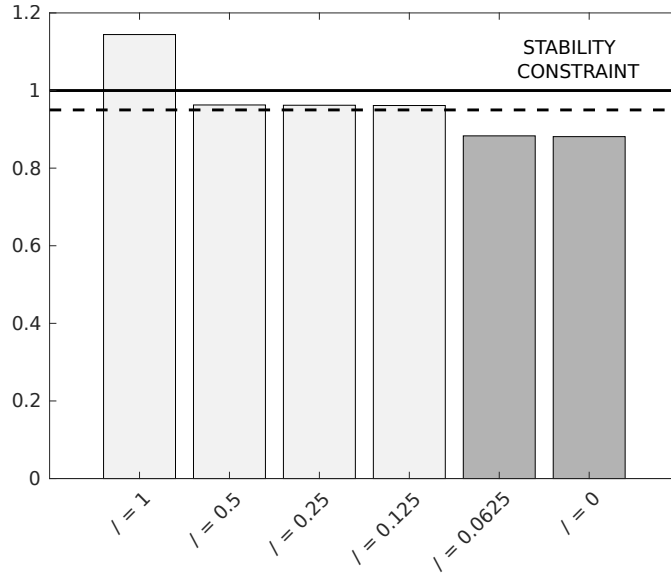


Figure 6.28: ANT-1: infinity norm discrepancy.

reduction of the infinity norm discrepancy in (2.27) as δ is reduced. According to the small-gain theorem, stability is ensured if the infinity norm in (2.26) is less than one. As suggested in [3], it is advisable to set a kind of stability margin to take into account the infinity norm estimation error in (2.27), that is why the stability test is passed only when the infinity norm is less than 0.95 (dashed line). To conclude, while the performance of the unfalsified controller with achievable reference model is very good both in terms of rise time and settling time, the one obtained imposing an unachievable reference is significantly slower (Figures 6.29 and 6.30). Nevertheless, it has been proved that the yielded controller is actually stabilising, contrary to what it would have been achieved with the VRFT algorithm.

Outer loop results. Once the controller parameters of the inner loop are defined and the inner closed-loop performance has been suitably validated, attention can be turned to the outer control loop. In particular, by tuning the outer controller, the complete attitude dynamics can be analysed. Unlike before, the tuning procedure is much simpler since the outer controller consists of a simple proportional term. Indeed, unlike the inner loop,

Chapter 6. Results

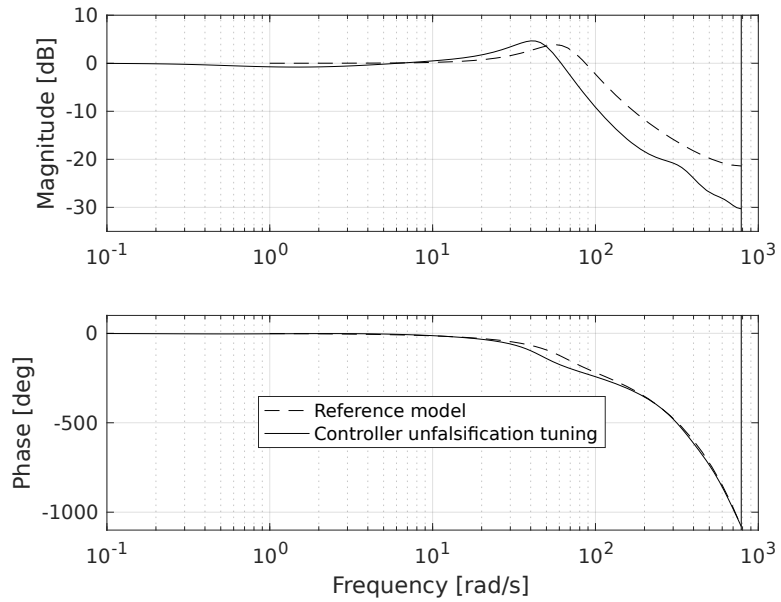


Figure 6.29: ANT-1: comparison of the inner loop Bode diagram considering the controller unfalsification tuning with unachievable reference model (simulation).

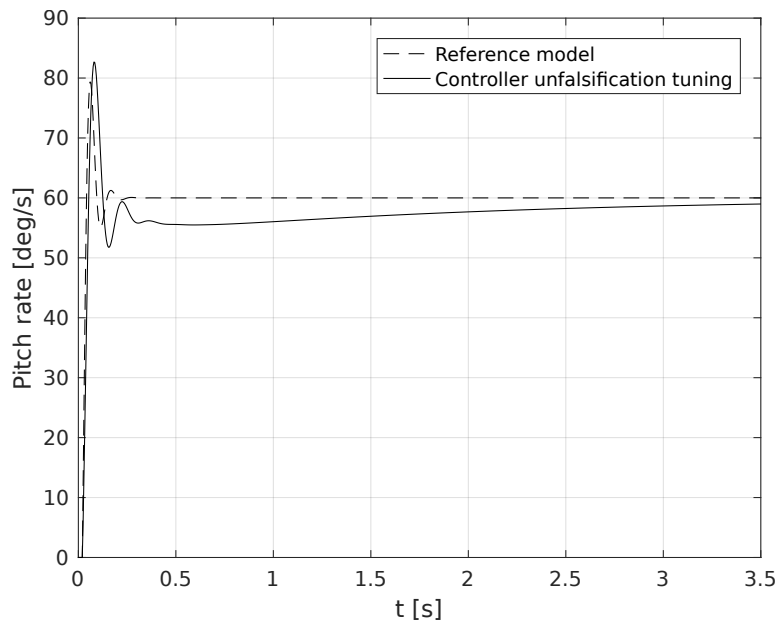


Figure 6.30: ANT-1: comparison of the inner loop step response considering the controller unfalsification tuning with unachievable reference model (simulation).

6.3. Ant-1

the same control architecture has been adopted for both the VRFT and the CbT tuning algorithms, that is a proportional controller as illustrated in Figure 3.4.

Note that, for simplicity, the controller unfalsification method has been applied only in the inner loop and the corresponding outer loop has been tuned through a simple VRFT algorithm. This is because the inner loop is far more critical from the closed-loop stability point of view. Furthermore, since the controller is defined only by the proportional term it is quite easy to understand whether the system is stable or not by comparing the parameter value with that of the manual tuning. Indeed, it has been verified that instability occur with very high values of K_P^O (approximately $K_P^O \in [35, 40]$), but such values have never been obtained with any reference model.

As shown in Figures 6.31 and 6.32, the closed-loop performance guaranteed by the VRFT tuning is comparable if not better than that provided by the H_∞ tuning. More specifically, the rise time is similar in both cases but the overshoot associated with the VRFT tuning is slightly smaller. Furthermore, the simulated system appears to be a bit slower than the reference model but overall the two are in accordance.

The VRFT, the CbT and the controller unfalsification methods exploit different reference models as illustrated in Table 6.7 and the results are displayed in different figures for the sake of clarity. Figure 6.33 and Figure 6.34 show the closed-loop performance guaranteed by the VRFT tuning considering the inner controller with the new SISO control architecture. The same performance considering the CbT is displayed in Figures 6.35 and 6.36. The outer proportional gain, associated with the inner unfalsified controllers, has been derived by means of the VRFT algorithm (Figures 6.37 and 6.38). The three methods are compared in Figures 6.39 and 6.40 without displaying the reference models. The outer loop confirms what has already been outlined by analysing the results of the inner loop: the performance ensured by the CbT tuning method is overall lower than that obtained by the VRFT approach, while it can be noted that the unfalsified controller has a very fast response with a minimal overshoot.

Chapter 6. Results

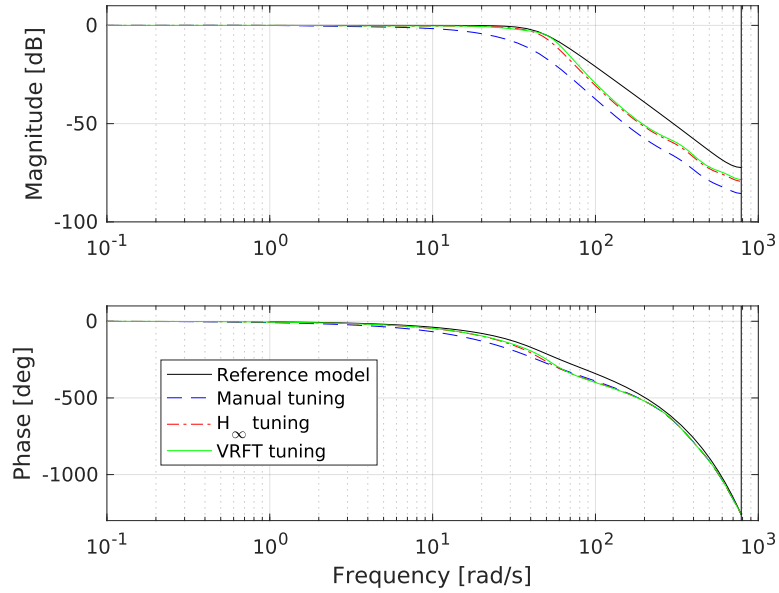


Figure 6.31: ANT-1: comparison of the outer loop Bode diagrams considering manual, VRFT and H_∞ tunings (simulation).

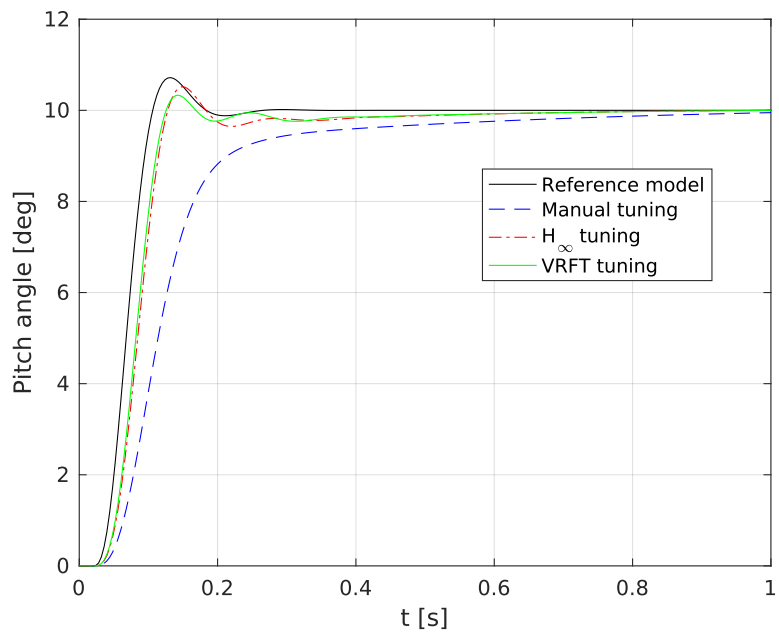


Figure 6.32: ANT-1: comparison of the outer loop step responses considering manual, VRFT and H_∞ tunings (simulation).

6.3. Ant-1

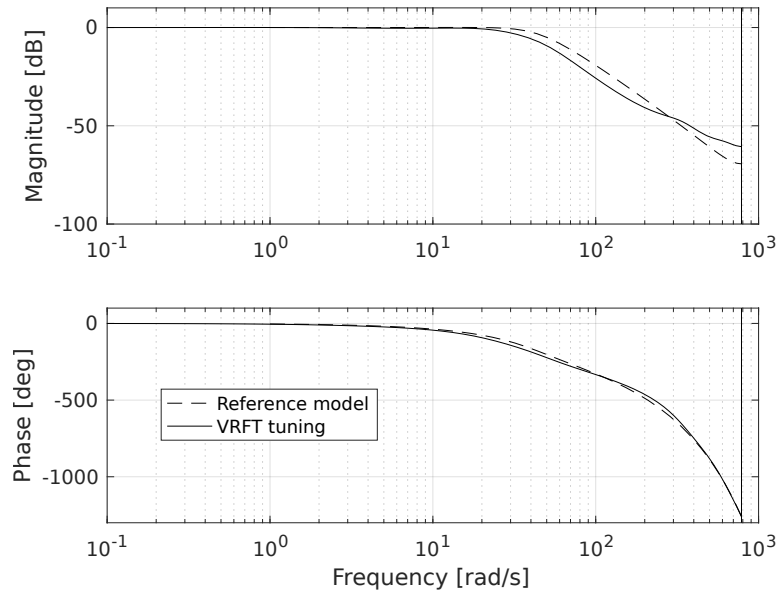


Figure 6.33: ANT-1: the outer loop Bode diagram considering the VRFT tuning (simulation).

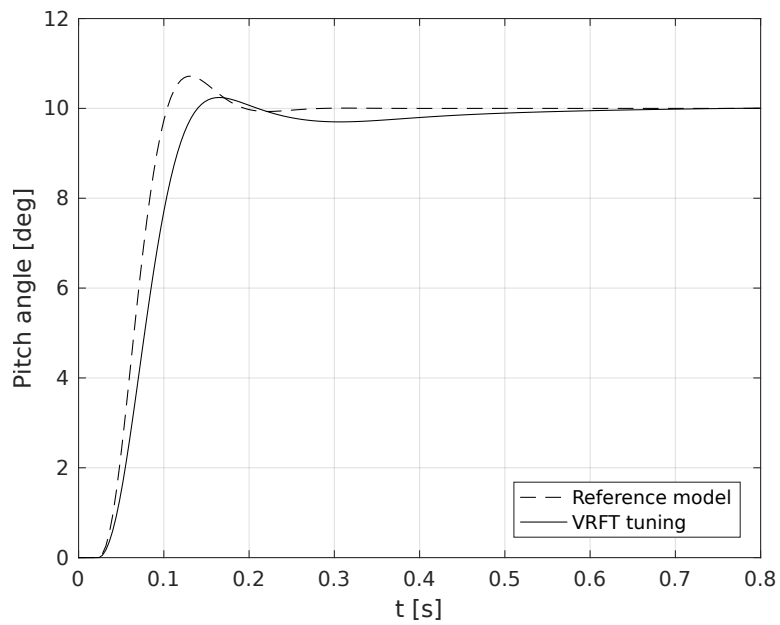


Figure 6.34: ANT-1: the outer loop step response considering the VRFT tuning (simulation).

Chapter 6. Results

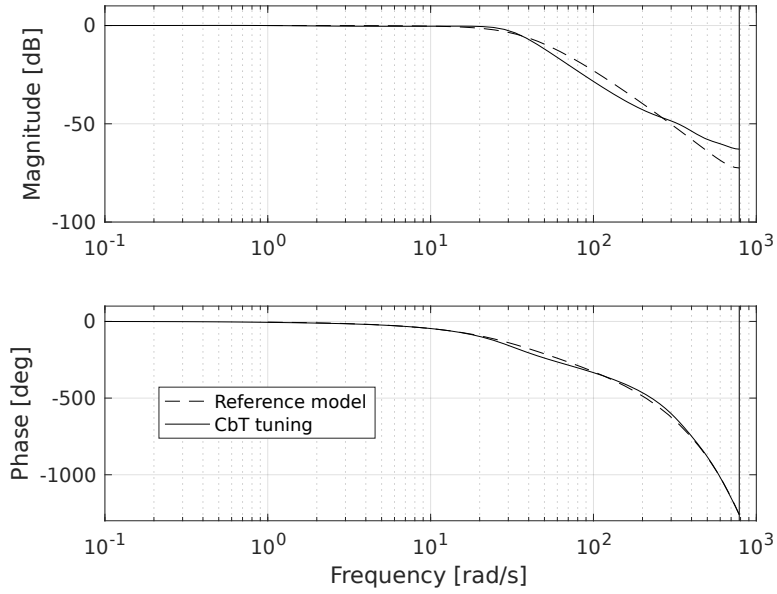


Figure 6.35: ANT-1: the outer loop Bode diagram considering the CbT tuning (simulation).

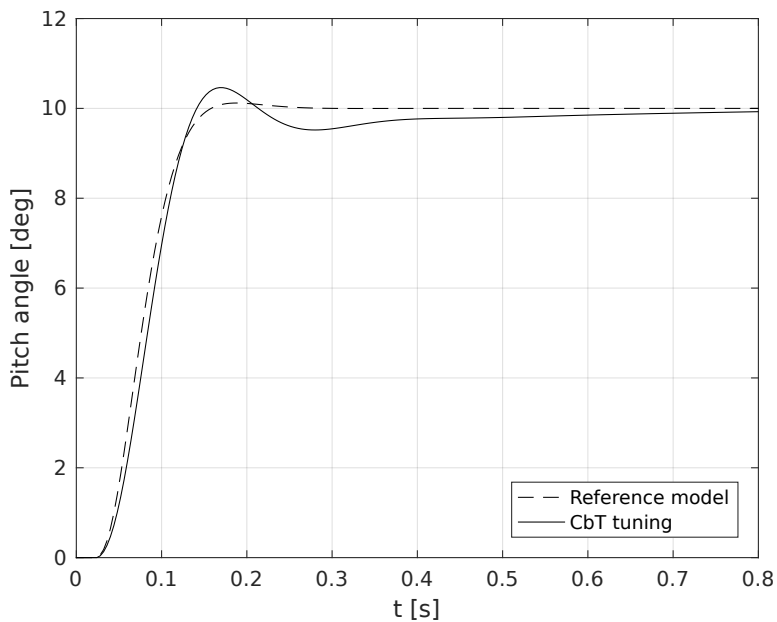


Figure 6.36: ANT-1: the outer loop step response considering the CbT tuning (simulation).

6.3. Ant-1

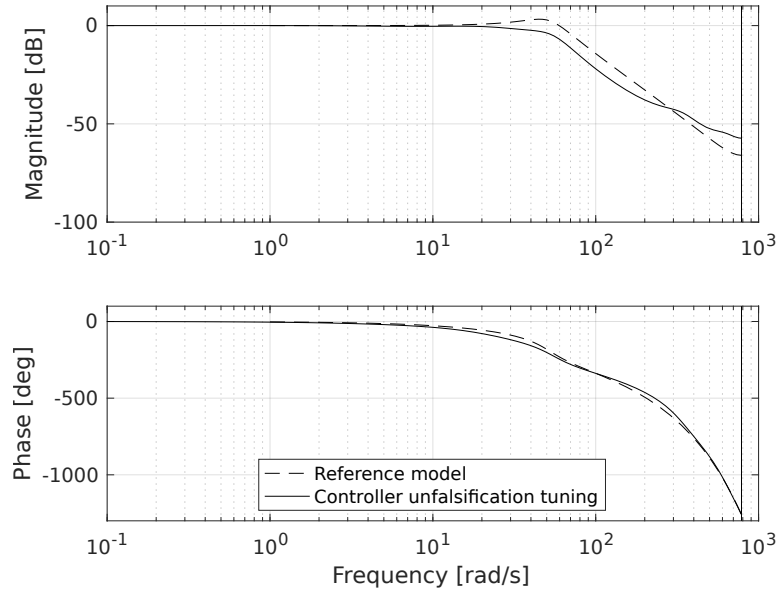


Figure 6.37: ANT-1: the outer loop Bode diagram considering the controller unfalsification tuning (simulation).

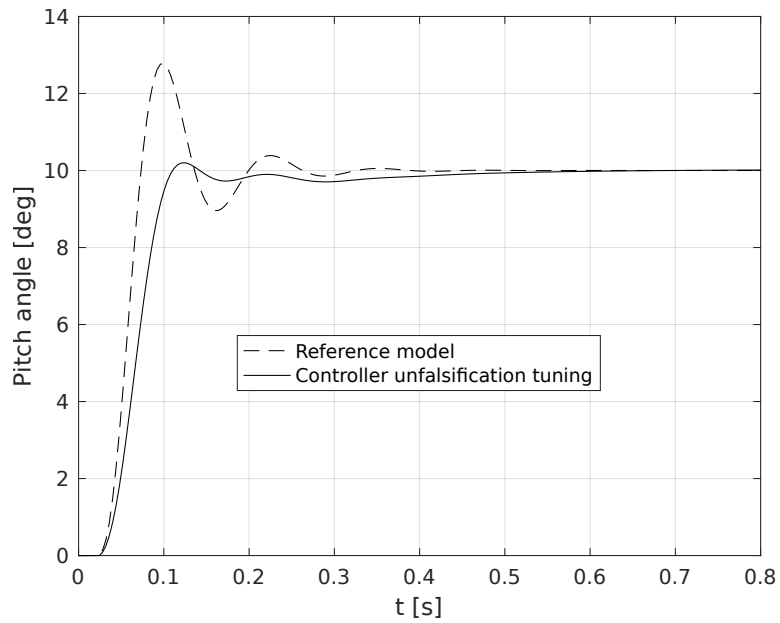


Figure 6.38: ANT-1: the outer loop step response considering the controller unfalsification tuning (simulation).

Chapter 6. Results

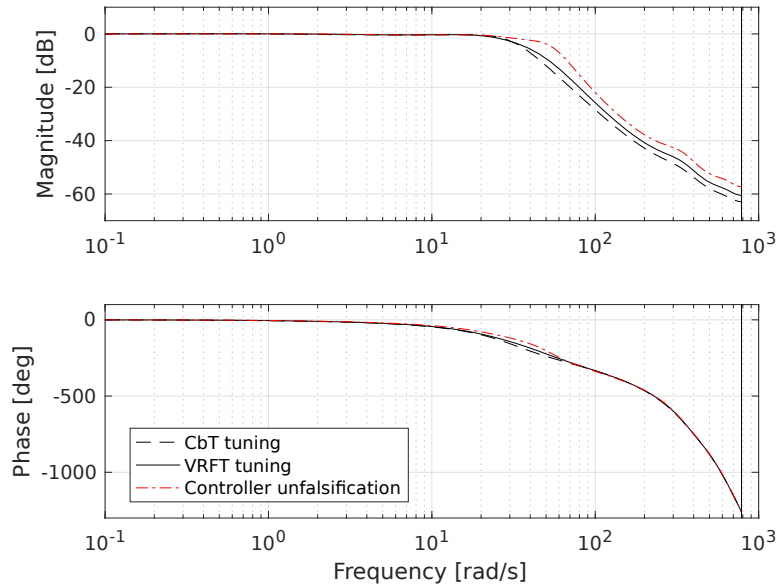


Figure 6.39: ANT-1: comparison of the outer loop Bode diagrams considering VRFT, CbT and controller unfalsification tunings (simulation).

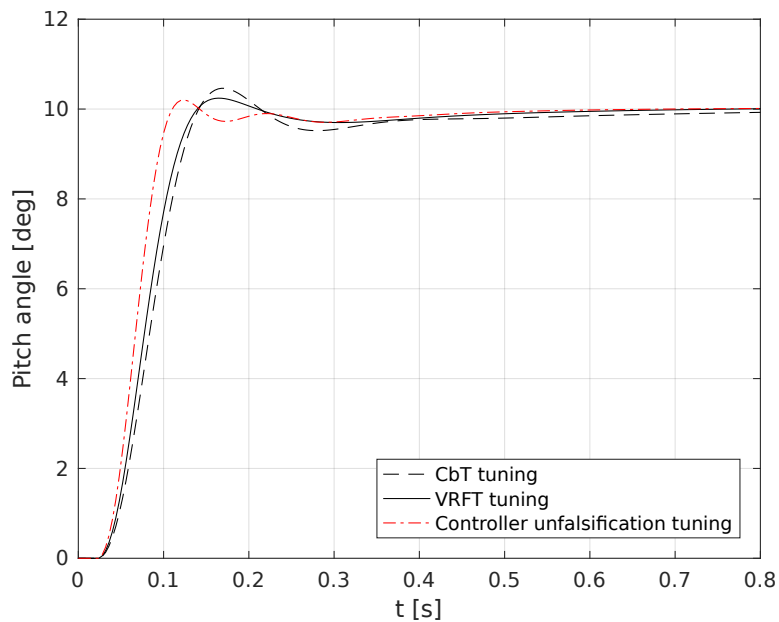


Figure 6.40: ANT-1: comparison of the outer loop step responses considering VRFT, CbT and controller unfalsification tunings (simulation).

Experimental results

The reliability of the simulation results is closely related to the accuracy of the identified model but, even if the model response has proved to be very similar to the real one, experimental validating tests are essential. Indeed, it is important to verify that the performance achieved in the simulation environment is achievable even in practice without making the system unstable, which should not be taken for granted because of the unavoidable approximations of the model.

The attitude tests have been performed on the test-bed that constrains all translational and rotational degrees of freedom except the pitch rotation. This ensures that the tests are repeatable and safe, preventing crashes due to erroneous choice of the controller parameters or wrong test characterisation. Two different types of tests have been performed, the first one consists of a sequence of steps of constant duration and increasing amplitude, while the second one is a disturbance rejection test as for the previous UAVs.

The first test consist of assigning a desired pitch angle command history and recording the system response. The setpoint time history has been defined as a sequence of steps with amplitudes of 5 deg and 10 deg. The second test allows to assess the effect of a wind gust on the quadrotor. The pitch angle setpoint has been set to zero for the whole test duration and a pitch moment disturbance has been applied and maintained constant for 5 seconds. Both tests have been performed for all available controller tunings, that is manual, H_∞ , CbT, controller unfalsification and VRFT tunings.

The performance obtained with the VRFT algorithm has been compared with H_∞ and manual tunings with the control architecture illustrated in Figure 3.4. Figure 6.41 shows the entire time history of the setpoint tracking test, but the fast system response does not allow to perceive the differences between the closed-loop responses. For this reason, an enlargement of a single step response has been reported for both amplitudes, 5 deg (Figure 6.42) and 10 deg (Figure 6.43). As expected in simulation, VRFT and H_∞ tunings yield a similar dynamic response that is significantly faster than that associated to the manually tuned controller. When considering the disturbance rejection test (Figure 6.44) the performance of the VRFT

Chapter 6. Results

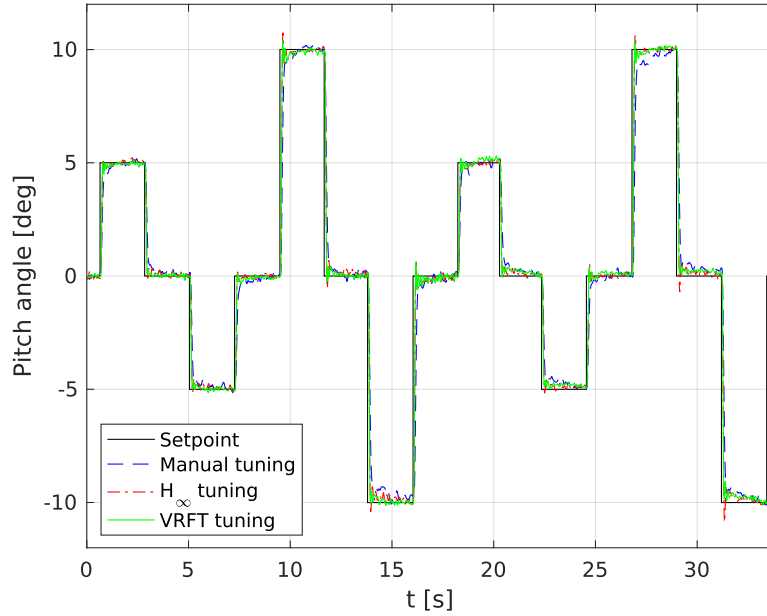


Figure 6.41: ANT-1: setpoint tracking with manual, H_∞ and VRFT tuning (experiment).

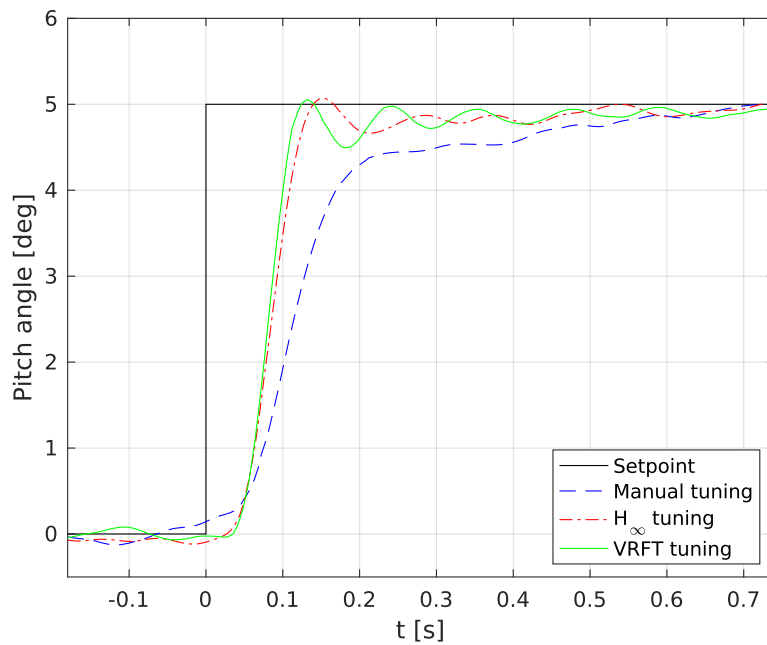


Figure 6.42: ANT-1: setpoint tracking (5 deg step) with manual, H_∞ and VRFT tuning (experiment).

6.3. Ant-1

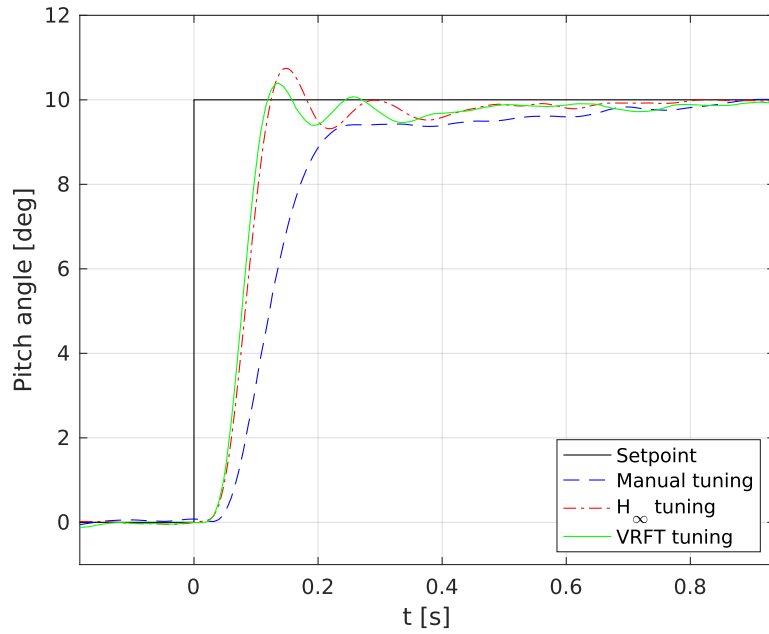


Figure 6.43: ANT-1: setpoint tracking (10 deg step) with manual, H_{∞} and VRFT tuning (experiment).

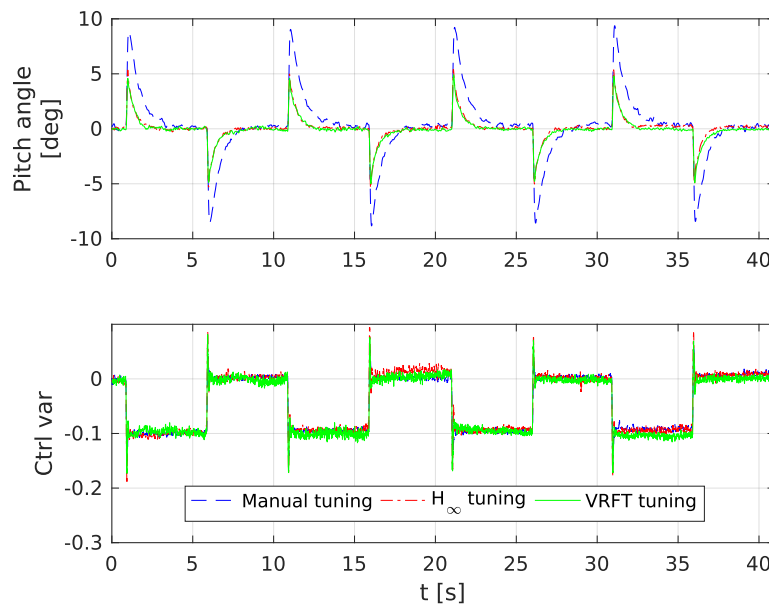


Figure 6.44: ANT-1: load disturbance rejection with manual, H_{∞} and VRFT tuning (experiment).

Chapter 6. Results

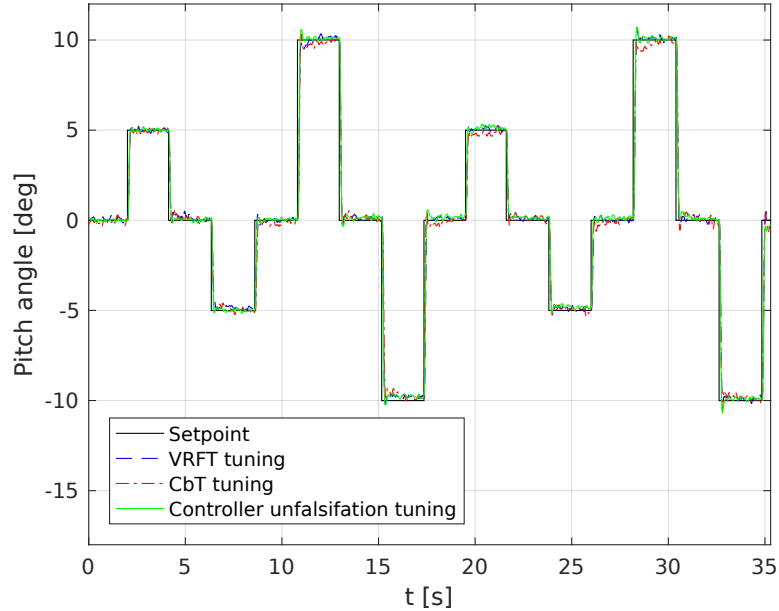


Figure 6.45: ANT-1: setpoint tracking with VRFT, CbT and controller unfalsification tuning (experiment).

method is slightly better than that of the H_∞ tuning both considering the pitch angle and the control effort. On the other hand, both methods ensure a significantly better performance with respect to the manual tuning but, at the same time, require an higher control effort.

As explained before, the control architecture implemented in the firmware of the quadrotor has been modified to handle the tunings yielded by the CbT and the controller unfalsification methods. Figures 6.45, 6.46 and 6.47 show the setpoint tracking tests of VRFT, CbT and the controller unfalsification tuning. As can be noted from the figures, there are no significant differences in performance between the three methods. On the contrary, by looking at the disturbance rejection test (Figure 6.48) the performance of the CbT tuning turns out to be lower than that of the VRFT and the unfalsified controller. In particular, the latter ensures the fastest disturbance rejection, even though the control effort is slightly larger than the one requested by VRFT and CbT tunings.

6.3. Ant-1

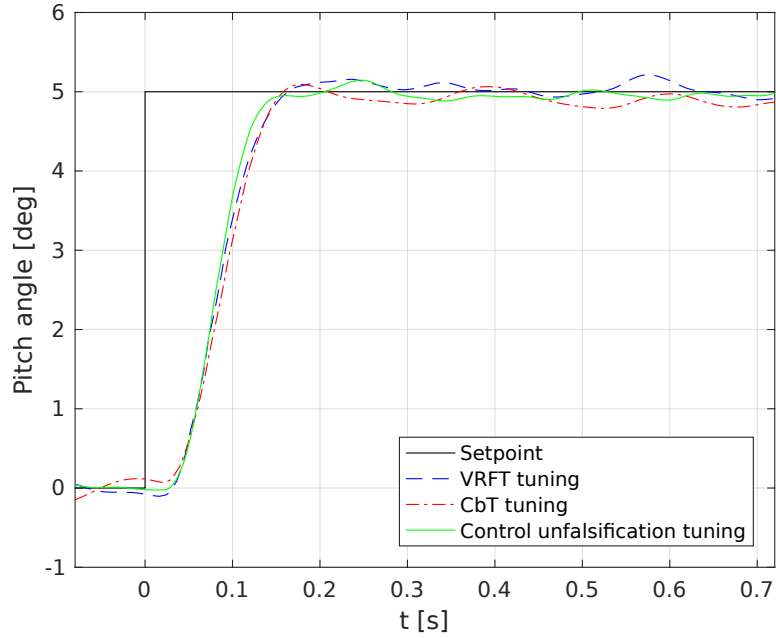


Figure 6.46: ANT-1: setpoint tracking (5 deg step) with VRFT, CbT and controller unfalsification tuning (experiment).

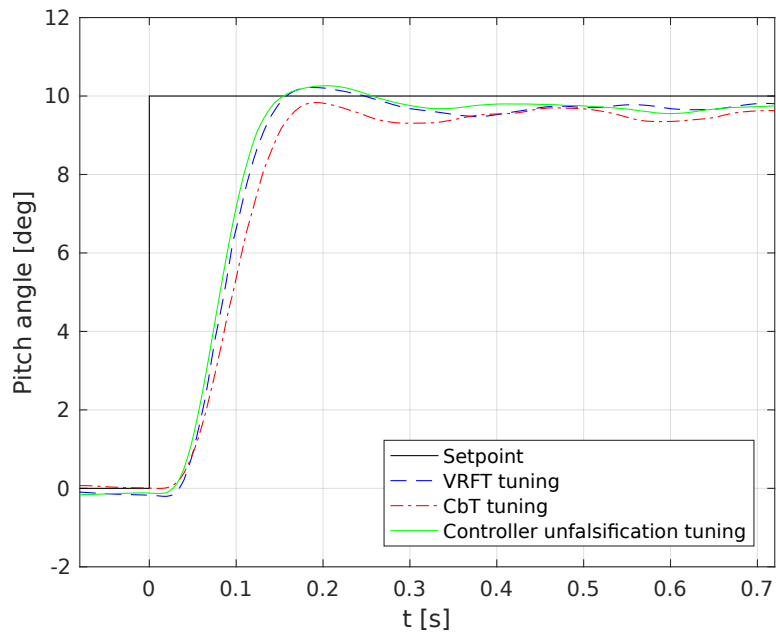


Figure 6.47: ANT-1: setpoint tracking (10 deg step) with VRFT, CbT and controller unfalsification tuning (experiment).

Chapter 6. Results

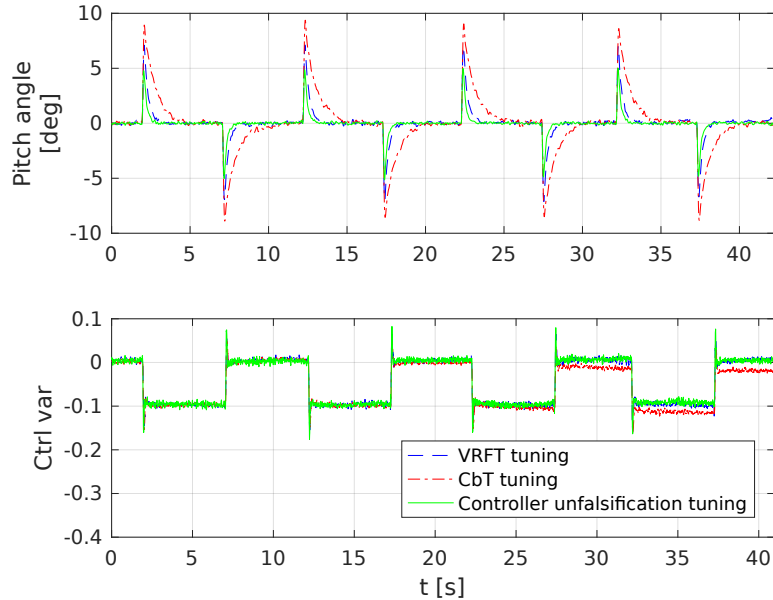


Figure 6.48: *ANT-1: load disturbance rejection with VRFT, CbT and controller unfalsification tuning (experiment).*

6.3.2 Pitch attitude controller with closed-loop experiments

In Section 6.3.1 the data-driven methods employ experimental data that come from open-loop experiment carried out on the test-bed setup. Also in this section the pitch attitude controller is considered but the data is collected in flight. In this situation, as explained in Section 4.1, the data must be collected in closed-loop allowing the user to control the system also when the experiment is been conducting.

For the sake of simplicity, only the VRFT method is considered and the original control architecture displayed in Figure 3.4 is exploited. The results will be compared with the VRFT tuning obtained with open-loop experimental data as presented in Section 6.3.1.

As illustrated in Section 4.1, an initial controller $C_d(z)$ that stabilises the system must be available. This controller has the same structure and the manual tuning as presented in Table 6.8 emerges as the most obvious choice since it is the first controller that was obtained with a trial and error procedure.

6.3. Ant-1

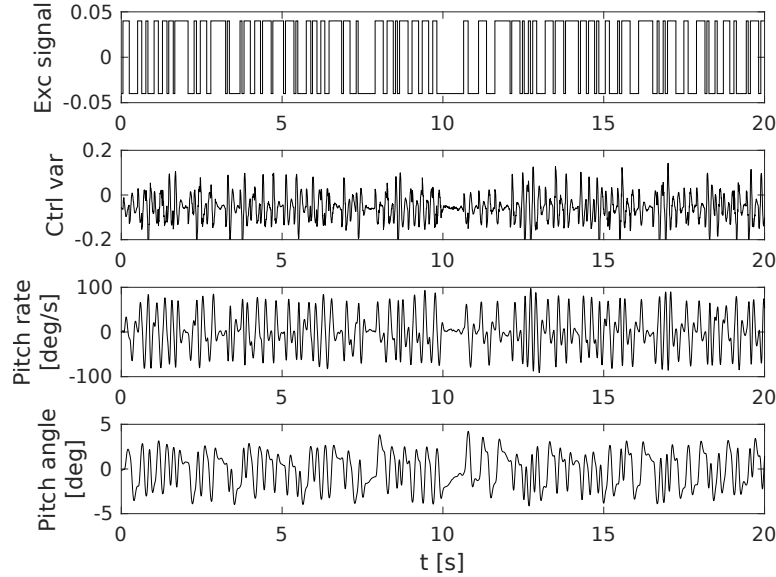


Figure 6.49: ANT-1: closed-loop experimental dataset used by data-driven methods.

As illustrated in Figure 4.1, the excitation signal $\bar{u}(t)$ is added to the output of the controller $C_d(z)$. All the involved signals in the data-driven tuning procedure are illustrated in the Figure 6.49. In this case, the control variable that is the pitch moment applied to the UAV (it is illustrated in the second plot of Figure 6.49) does not clearly show the PRBS excitation signal because in the closed-loop condition it is the sum of the excitation signal and the output of the controller that tries to obtain a null pitch angle. During the experiment, the user did not provide a pitch angle setpoint ($\bar{y}(t) = 0$, see again Figure 4.1) because the UAV operated always in safe conditions and it did not reach the limits of the test area.

Reference models

Comparing this section and Section 6.3.1, the only thing that changes is how the experimental data is obtained. The system to be controlled and the controller architecture do not change. Since the closed-loop reference model depends implicitly from these two last systems, the reference model both for the inner and the outer loops do no change. Thus the closed-loop reference model for the inner loop is expressed in (6.3) and (6.4) is the

Chapter 6. Results

Table 6.10: *ANT-1: optimal controller parameters for outer and inner controllers considering the VRFT method with open-loop and closed-loop experimental data.*

Controller parameter	VRFT with open-loop data	VRFT with closed-loop data
K_P^O	12.11	12.4751
K_{FF}^I	0	0
K_P^I	0.09143	0.1359
K_I^I	0.2067	0.1008
K_D^I	0.0016	0.0019

reference model for the outer loop. Since the reference model are the same in the open-loop and closed-loop frameworks, it is easy to compare the two tunings obtained with the different data without performing another tuning with the new reference model exploiting the data in Section 6.3.1.

Controller parameter values

Exploiting the reference models and closed-loop experimental data the VRFT method leads to the parameter values reported in Table 6.10. Both the parameters for the inner and the outer controllers are displayed. Since this tuning is compared with the one obtained with open-loop data in Section 6.3.1, also this tuning is reported in the table. To deal with noisy data, an instrumental variable is employed and it is built through the identification of the inner and the outer loops. For the inner loop, a third-order of the plant model is identified with the PBSID algorithm with $p = 40$ and $f = 40$, while for the outer loop, an ARX(5,5) model is employed.

Simulation results

Also in this case the plant model in (6.5) and the scheme in Figure 6.21 are exploited to better show the results.

First the inner loop on the pitch angular rate is considered.

As can be seen, the frequency response of the VRFT tuning is very close to that obtained with the H_∞ controller and approximates quite well the desired behaviour. The same remarks can be made by comparing the step responses of the different tunings (Figure 6.24).

6.3. Ant-1

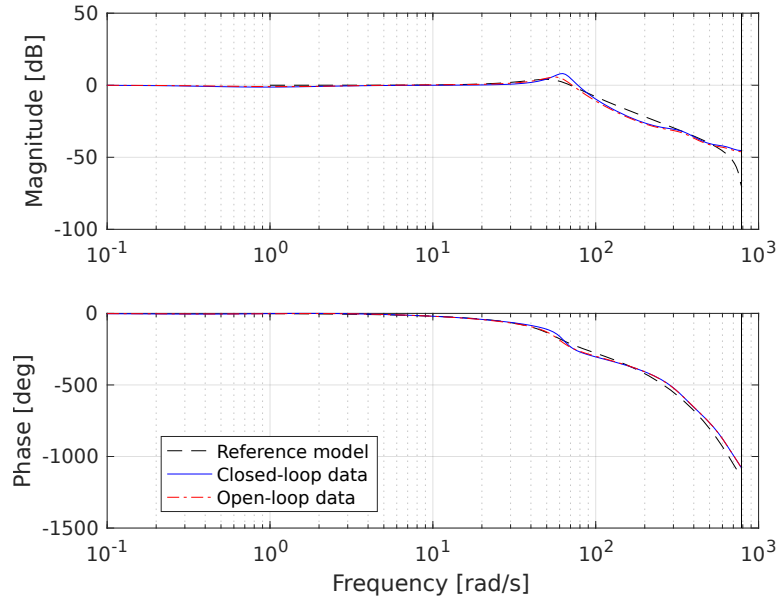


Figure 6.50: ANT-1: comparison of the inner loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

Figure 6.50 shows the Bode diagram obtained by the VRFT tuning considering experimental data from a closed-loop test. Comparing this tuning with the one obtained with the VRFT method using open-loop data, the frequency responses of the closed-loop systems are quite similar. Nevertheless considering the step responses of the two VRFT tunings, the VRFT method that employed closed-loop experimental data leads to a more oscillating behaviour (see Figure 6.51).

Now the attention is moved to the outer proportional controller that acts on the pitch angle setpoint and pitch angle measurements generating the setpoint signal for the inner controller (see again Figure 3.4).

As illustrated in Figures 6.52 and 6.53, the closed-loop performance guaranteed by the VRFT tuning obtained with closed-loop data is comparable with the one provided by the VRFT tuning with open-loop data. The rise times are the same and the step response with the closed-loop data VRFT methods is slightly more oscillating.

Chapter 6. Results

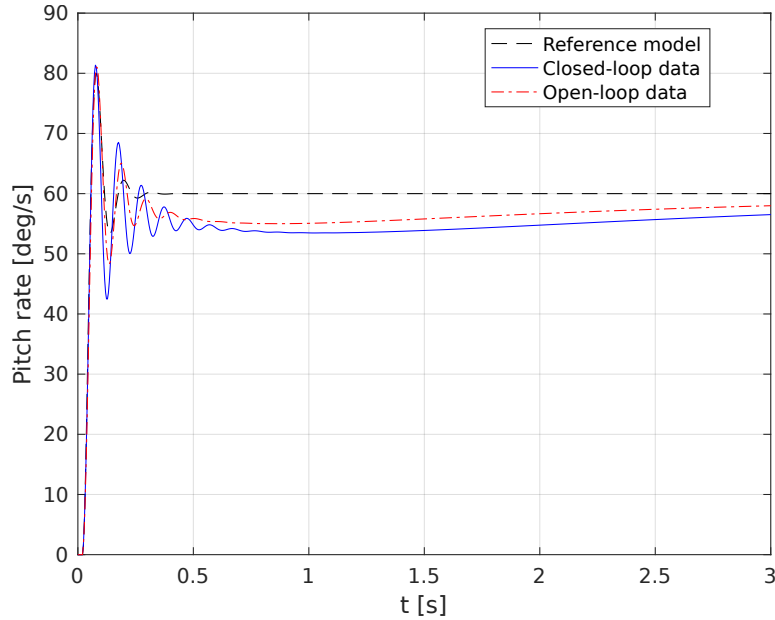


Figure 6.51: *ANT-1: comparison of the inner loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation).*

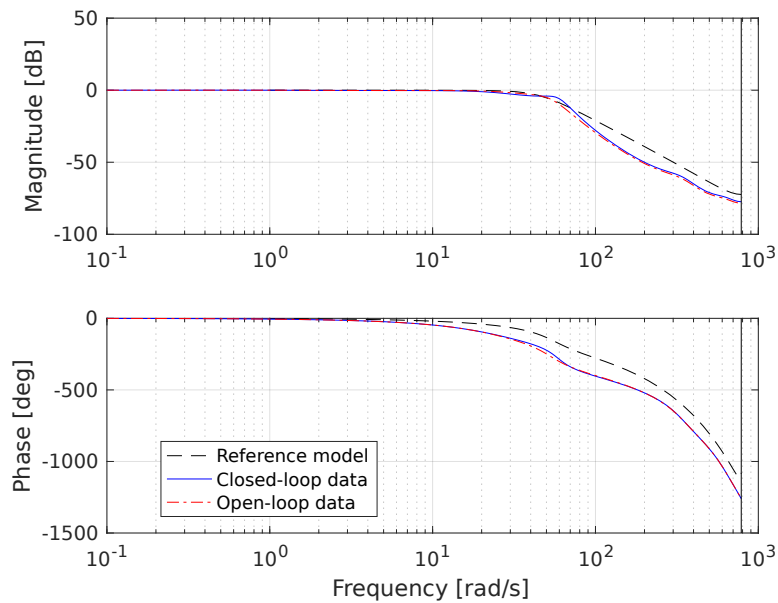


Figure 6.52: *ANT-1: comparison of the outer loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation).*

6.3. Ant-1

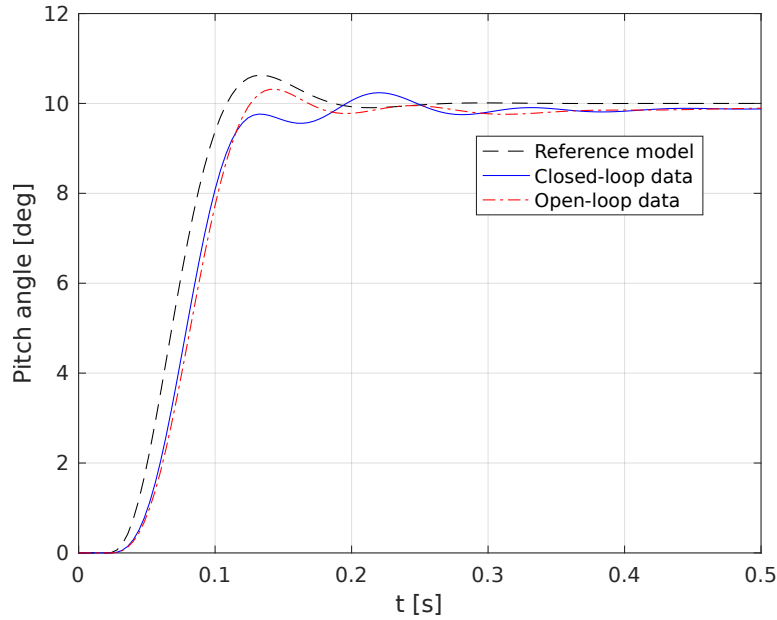


Figure 6.53: ANT-1: comparison of the outer loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

Experimental results

As explained in Section 6.3.1, the performance achieved in the simulation environment must be validated also in practice, operating on the real system. The attitude tests have been performed with the test-bed setup and also in this case two different types of tests have been performed as in Section 6.3.1: a setpoint tracking and a disturbance rejection load disturbance evaluation.

As in the simulation environment, the VRFT method exploiting closed-loop experimental data is compared with the VRFT algorithm that instead exploits open-loop data. Figure 6.54 shows the entire time history of the setpoint tracking test and an enlargement of a single step response has been reported for both amplitudes, 5 deg and 10 deg in Figure 6.55 and in Figure 6.56 respectively. As expected in simulation, the VRFT tuning obtained with closed-loop experimental data leads to a more oscillating response but the rise time is the same of the response of the VRFT tuning that exploits open-loop data.

Chapter 6. Results

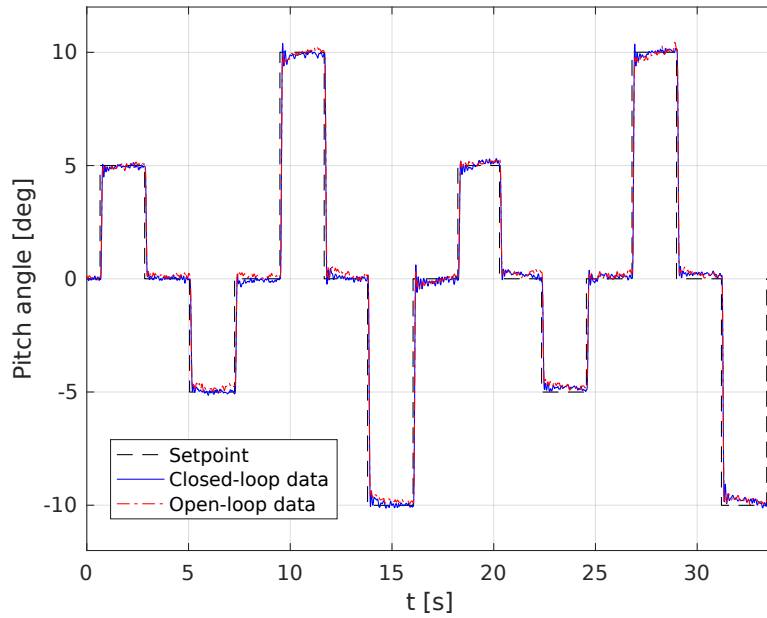


Figure 6.54: ANT-1: setpoint tracking comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

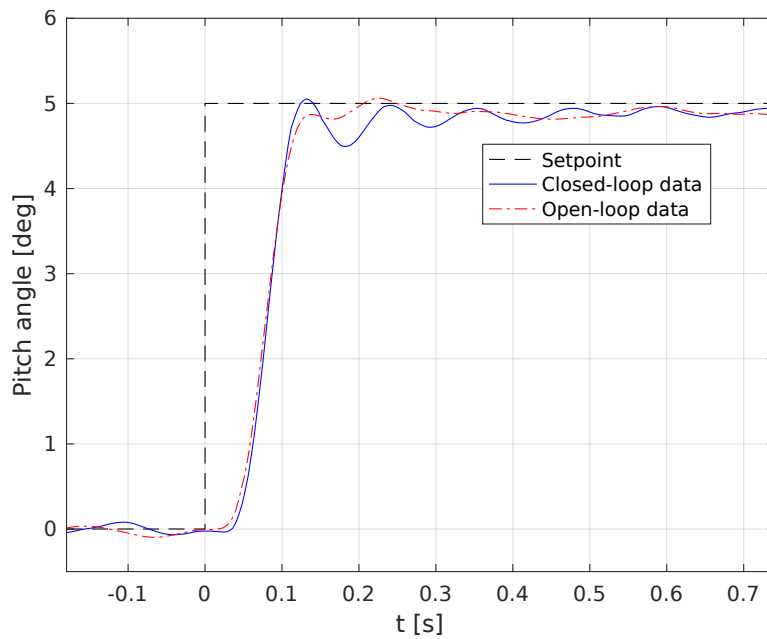


Figure 6.55: ANT-1: setpoint tracking (5 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

6.3. Ant-1

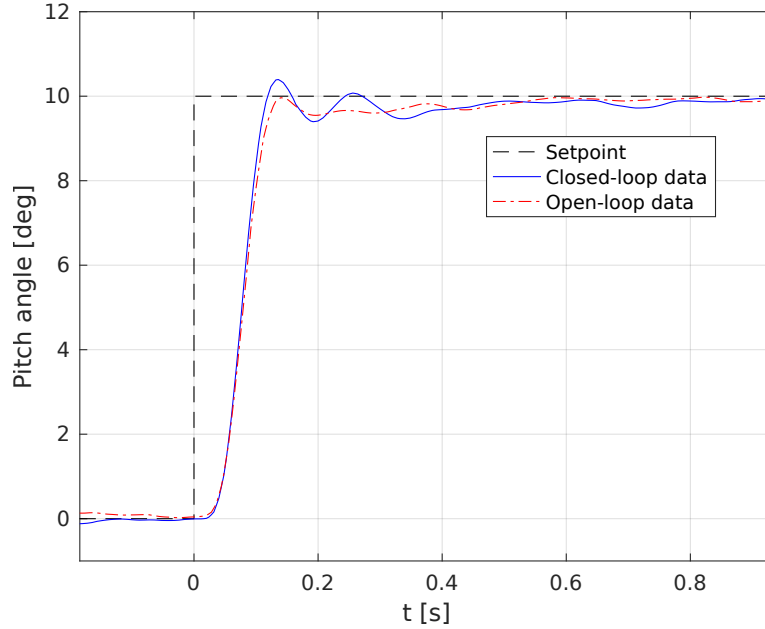


Figure 6.56: *ANT-1: setpoint tracking (10 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).*

When considering the disturbance rejection test (Figure 6.57) the performance of VRFT method obtained with open-loop data is slightly better than that of the same data-driven method but obtained with closed-loop experimental data.

6.3.3 Pitch-roll attitude controllers with closed-loop experiments

Until now only the pitch attitude controller was considered. As explained in Section 3.1, the roll DoF are controlled with the same regulator scheme in Figures 3.4 thanks to the geometrical symmetry of the quadrotor. Obviously, the involved signals are different: the user provides the roll angle setpoint $\bar{\phi}(t)$ and the proportional outer controller generates the roll angular rate reference signal. The inner regulator, starting from this information, computes the roll pitch moment $\mathcal{L}(t)$.

As illustrated in Section 3.1, when the quadrotor is in near-hovering conditions, the roll and pitch DoFs could be assumed decoupled. In this case the pitch and the roll can be tuned independently. If, for any reason,

Chapter 6. Results

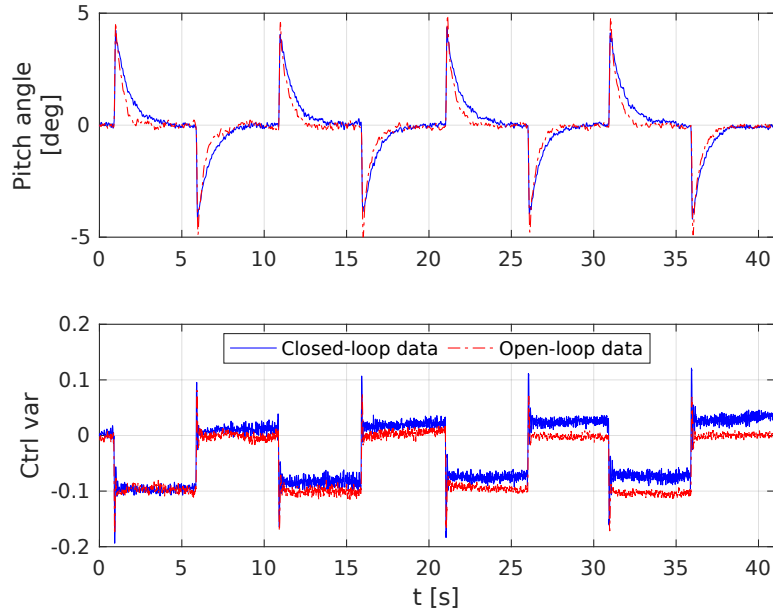


Figure 6.57: ANT-1: load disturbance rejection comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

the system to control does not have a decoupled dynamics between the DoFs, the different DoF controllers must be tuned at the same.

For the sake of simplicity, only the VRFT method is considered. As described in Section 2.1, the Algorithm 1 is not limited to SISO control systems and VRFT can deal also with MIMO regulators.

In this section only the inner loop controllers, based on the roll and pitch angular feedbacks, are taken into account. In particular the considered controller has four inputs (pitch and roll angular setpoints, and pitch and roll angular rate measurements) and two outputs (pitch and roll moments). Internally it has four independent regulators as displayed in Figure 3.4: one for the pitch DoF, one for the roll DoF and two for the coupled dynamics. In this case the controller parameters are not scalar but they are 2×2 matrices: on the main diagonal there are the parameters for the pitch and the roll DoF controllers and on the secondary diagonal the parameters of the coupled dynamics controllers.

The experimental data are collected in closed-loop. Two PRBS excitation signals, one for the pitch moment and one for the roll moment, are

6.3. Ant-1

Table 6.11: ANT-1: parameters for the initial controller exploited in the test to collect data for the MIMO controller tuning procedure.

Controller parameter	
K_P^O	$\begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$
K_{FF}^I	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
K_P^I	$\begin{bmatrix} 0.06 & 0 \\ 0 & 0.06 \end{bmatrix}$
K_I^I	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$
K_D^I	$\begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$

applied at the same time. The two signals are different but they share the same PRBS parameters (signal amplitude and min/max switching interval).

As illustrated in Section 4.1, an initial controller $C_d(z)$ that stabilises the system must be available. In this case the parameters of this initial controller are illustrated in Table 6.11.

Figure 6.58 and Figure 6.59 show the involved signal in the data-driven tuning procedure. For the sake of clarity, the signals are represented in two figures but all the signals feeds the data-driven procedure at the same time.

Reference models

Comparing this section and Section 6.3.2, now the reference models are 2x2 matrices of transfer functions. In particular, since only the inner controller is considered, the reference model is a 2x2 matrix with the transfer function defined in (6.3) on the main diagonal and zeros on the secondary diagonal.

Controller parameter values

Exploiting the reference models and closed-loop experimental data the VRFT method leads to the parameter values reported in Table 6.12. To deal with

Chapter 6. Results

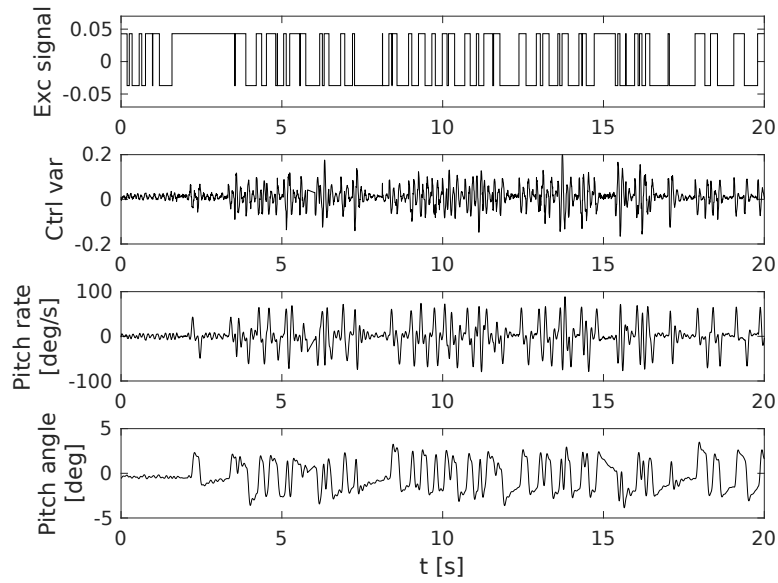


Figure 6.58: *ANT-1: closed-loop pitch experimental dataset used by MIMO data-driven method.*

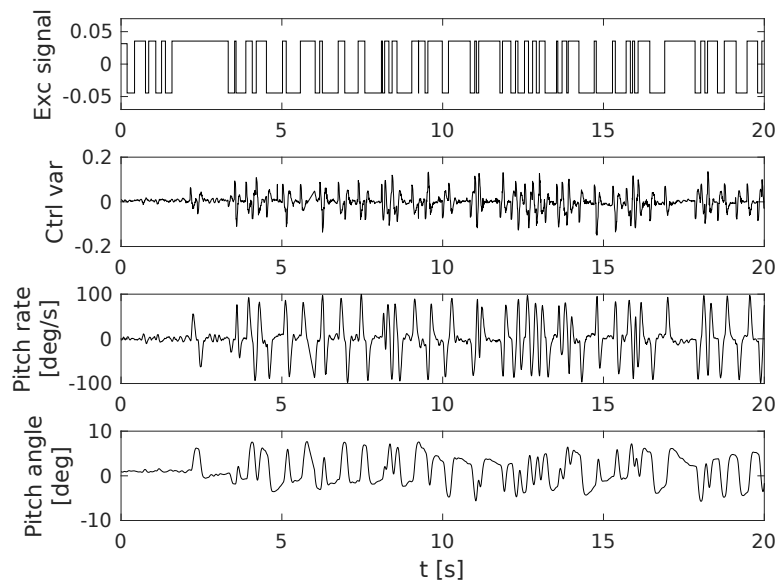


Figure 6.59: *ANT-1: closed-loop roll experimental dataset used by MIMO data-driven methods.*

6.3. Ant-1

Table 6.12: *ANT-1: optimal controller parameters for outer and inner controllers considering the VRFT MIMO method with closed-loop experimental data.*

Controller parameter	
K_{FF}^I	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
K_P^I	$\begin{bmatrix} 0.0896 & 0 \\ 0 & 0.0901 \end{bmatrix}$
K_I^I	$\begin{bmatrix} 0.1854 & 0 \\ 0.0001 & 0.1873 \end{bmatrix}$
K_D^I	$\begin{bmatrix} 0.0015 & 0 \\ 0 & 0.0015 \end{bmatrix}$

noisy data, an instrumental variable is employed and it is built through the identification of the plant model with the PBSID algorithm with $p = 40$, $f = 40$ and model order equal to 10.

Since the experiments are conducting in near-hovering conditions, the secondary diagonal of the parameters in Table 6.12 is almost always equal to zero confirming decoupled dynamics between the DoFs in the quadrotor platform. Furthermore, the results for the the pitch controller (first element of the matrices) are similar to those obtained exciting only the pitch DoF in Table 6.10.

6.3.4 Position controller

In this section, the results of the position controller tuning, considering the ANT-1 quadrotor, will be discussed.

For the sake of simplicity, only the VRFT method is considered and the original control architecture displayed in Figure 3.5 is exploited. Just for recall, the inner controller is a PID with derivative action applied on the measurement and supplemented with a feedforward term, while the external controller consists of a simple proportional gain. Since this controller was never previously tuned, the closed-loop results will be compared with the closed-loop reference model.

Chapter 6. Results

The FCU of the ANT-1 quadrotor is modified to perform the required excitation test. Furthermore, thanks to the small size of the UAV the test has been performed indoor in the flying arena. In the initial part of the test, the quadrotor is manually controlled in attitude via a remote controller. The excitation test begins as soon as the drone is approximately stable in mid-air position. The thrust required in hovering conditions is kept constant throughout the test. In this situation, a PRBS pitch reference is applied as input in open-loop conditions. The reader should notice that, considering the position controller, an open-loop experiment means that the attitude controllers are enabled and the position controllers are disabled (see Section 3.4). Although the excitation test is performed in safe conditions since the system is stabilised by the attitude controller, each test duration does not exceed 5 seconds because, during the test, the drone tends to move away from the initial position and could potentially collide with the walls of the cage.

The input and output time histories used for the data-driven controller design are shown in Figure 6.60. The input signal is the demanded pitch and it is defined as a PRBS signal switching from -5 deg and 5 deg. Whereas, the output signals are the longitudinal linear velocity $v_x(t)$ and the longitudinal position $x(t)$. Two excitation tests are concatenated. The measurement signals come from the Kalman-based estimator that employs both on-board sensors (*e.g.*, accelerometer and gyroscope) and the information from the motion capture system that, using an array of infra-red sensitive cameras, provides the position of the UAV inside an indoor flight arena.

Reference models

As with the tuning of the attitude controller, a second-order system, defined by the natural frequency and the damping ratio, has been adopted as a baseline reference model. By doing so, the desired bandwidth and damping of the closed-loop system can be assigned.

As already mentioned for the attitude controller design, if the dynamic response of the system is roughly known, it is possible to augment the desired model so that it better matches the closed-loop behaviour of the real

6.3. Ant-1

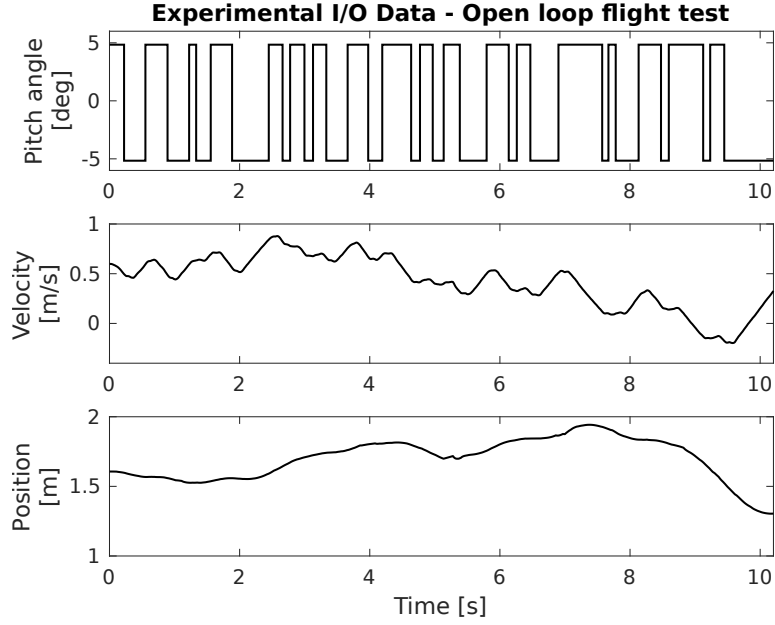


Figure 6.60: ANT-1: open-loop experimental dataset used by data-driven methods.

plant. This allows the VRFT method to achieve a controller tuning that ensures better closed-loop performance. Exactly as in the reference model of the attitude dynamics, a time delay of 5 samples. The inner loop reference model is a second order model with desired bandwidth of 1.3 rad/s and a damping ratio of 0.7 with an additional zero in 1.8 rad/s:

$$M_i(z) = z^{-5} \frac{0.003755z - 0.003728}{z^2 - 1.993z + 0.9927}.$$

In this specific case no filtering action was needed, thus the weighting function has been defined as $W_i(z) = 1$.

In a similar way, the outer loop reference model is defined: it is a second order model with desired bandwidth of 1.1 rad/s and a damping ratio of 0.6 and it does not contain additional poles or zeros:

$$M_o(z) = z^{-5} \frac{9.663 \cdot 10^{-6}z + 9.646 \cdot 10^{-6}}{z^2 - 1.995z + 0.9947}.$$

Chapter 6. Results

Table 6.13: *ANT-1: optimal controller parameters for outer and inner position controllers considering the VRFT method.*

Controller parameter	VRFT with open-loop data
k_P^O	0.77
k_{FF}^I	0
k_P^I	0.1806
k_I^I	0.05906
k_D^I	0.02172

Finally, unlike the inner loop, the use of a weighting function in the outer controller design has improved the matching between closed-loop response and reference model. In particular, $W_o(z)$ has been defined as a ninth-order lowpass digital *Butterworth filter*, with a cut-off frequency equal to 80 rad/s.

Controller parameter values

The VRFT algorithm is fed with the two reference models and the experimental data in Figure 6.60. The optimal values of the controller parameters computed by this data-driven method are reported in Table 6.13 where both the parameters for the inner and the outer controllers are displayed. To deal with noisy data, an instrumental variable is employed and it is built through the identification of an ARX(5,5) model for the inner and the outer loops.

Simulation results

The controller tunings obtained with the VRFT algorithm have been tested in a simulation environment. Unlike the attitude control loop, no previously identified position dynamics model was available. In particular, it should be defined as the dynamic relationship between the pitch angle $\vartheta(t)$ and the linear longitudinal velocity $v_x(t)$. Without carrying out any identification tests, it is possible to obtain a very basic model by means of a simple forces equilibrium in hovering conditions.

The discrete time transfer function between the pitch angle $\vartheta(t)$ and the

6.3. Ant-1

linear longitudinal velocity $v_x(t)$ is the following:

$$P_o(z) = -g \frac{T_s}{z - 1} \tag{6.6}$$

where g is the gravitational acceleration. As schematically shown in Figure 6.61, to obtain the position a second integrator block is needed. Obvi-

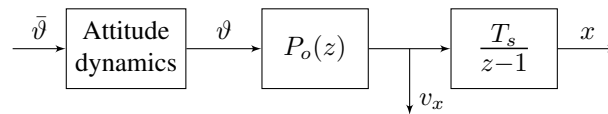


Figure 6.61: ANT-1: longitudinal position plant model.

ously, this model is characterised by significant approximations. In particular, all the aerodynamic effects are completely neglected. Both the aerodynamic drag and the inflow effect on the rotors yield a damping term in the dynamic equation that becomes more and more important moving away from hovering conditions.

Figure 6.62 and Figure 6.63 respectively represent the Bode diagram and the step response of the inner control loop. The simulated system appears to be slightly faster than the reference model but with larger overshooting, nevertheless, the two curves are very similar.

Regarding the position dynamics (Figure 6.64 and Figure 6.65), the simulated system with the VRFT tuned controller almost perfectly match the desired closed-loop behaviour.

Experimental results

As already stated before, the adoption of a very basic model for the plant does not ensure that good simulation results lead to the closed-loop stability considering the real system. Therefore, validation experiments conducted on the real plant are essential.

During the validation experiments both attitude and position controllers were enabled. To evaluate the closed-loop performance, a setpoint tracking test has been performed. More specifically, a desired position command history with respect to the centre of the testing cage has been provided as

Chapter 6. Results

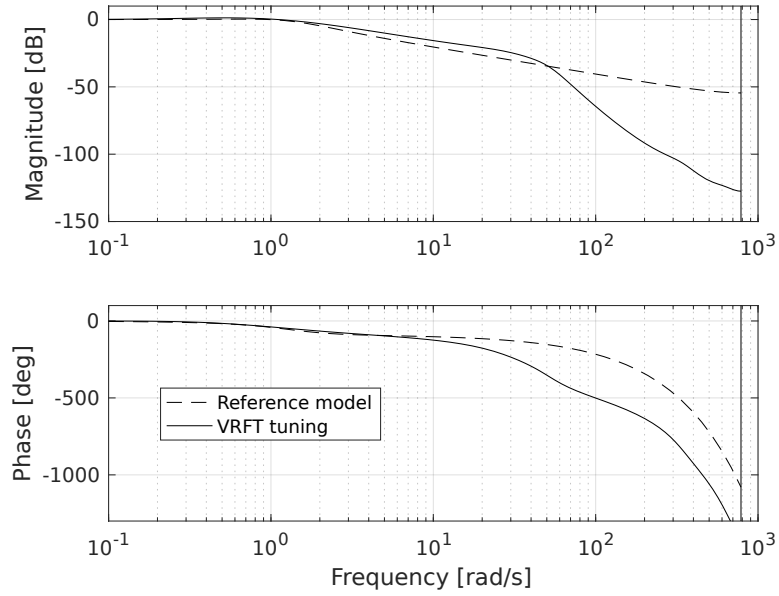


Figure 6.62: ANT-1: comparison of the position inner loop Bode diagram considering the VRFT tuning (simulation).

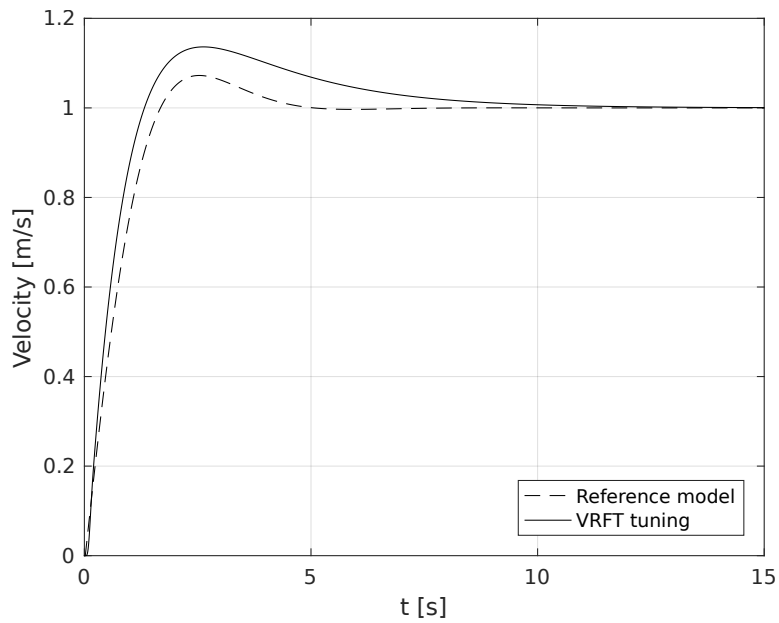


Figure 6.63: ANT-1: comparison of the position inner loop step response considering the VRFT tuning (simulation).

6.3. Ant-1

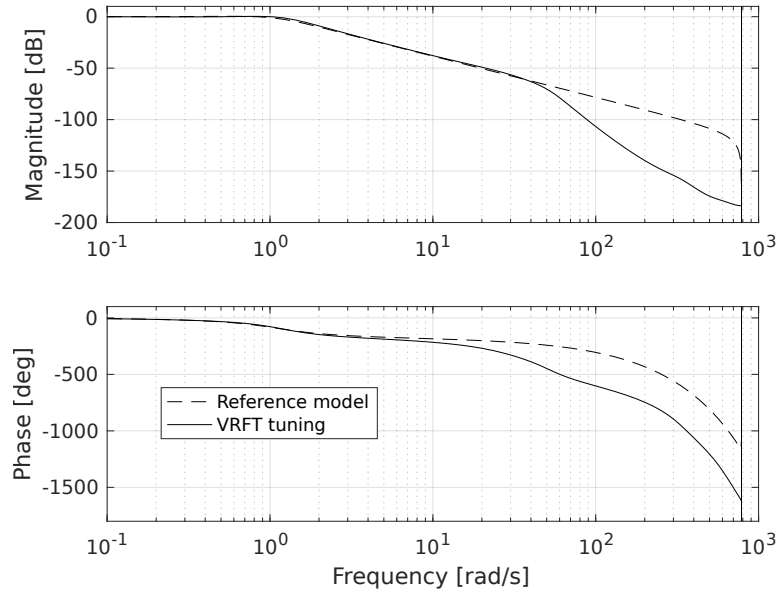


Figure 6.64: ANT-1: comparison of the position outer loop Bode diagram considering the VRFT tuning (simulation).

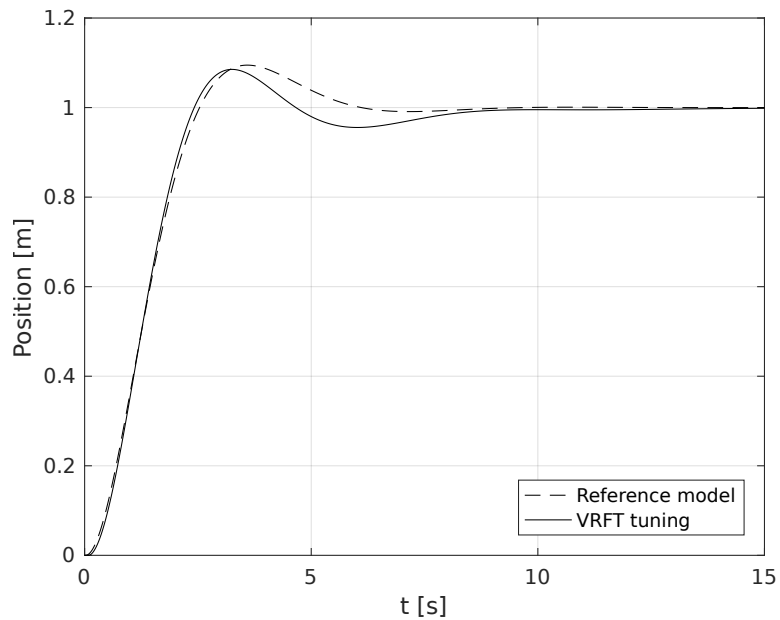


Figure 6.65: ANT-1: comparison of the position outer loop step response considering the VRFT tuning (simulation).

Chapter 6. Results

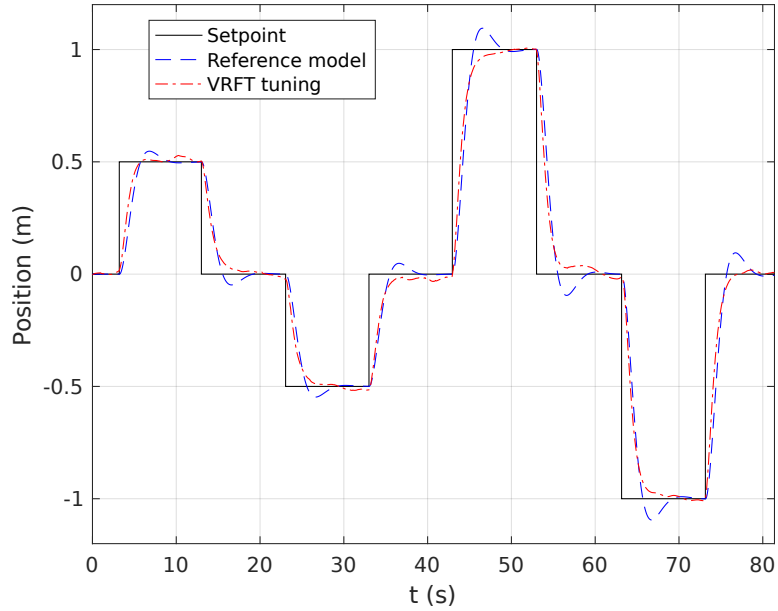


Figure 6.66: *ANT-1: position setpoint tracking with VRFT tuning and ideal reference model response (experiment).*

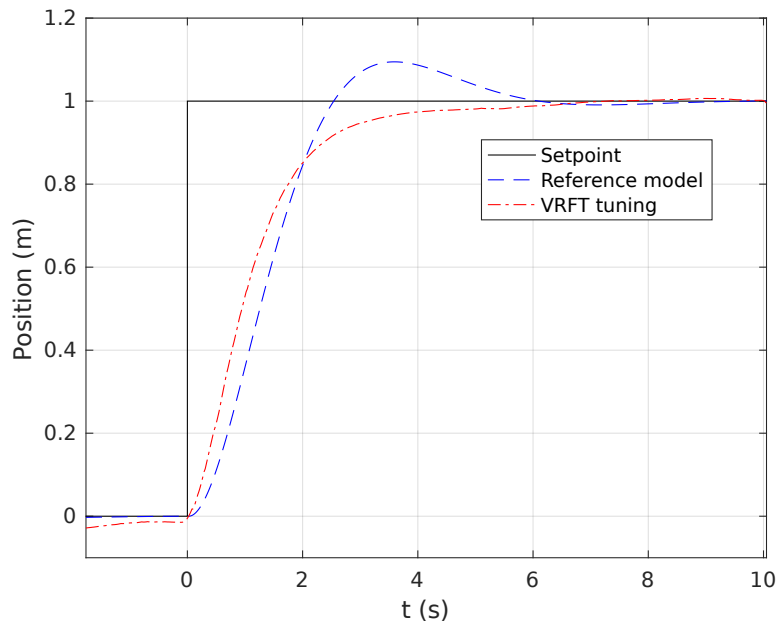
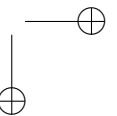
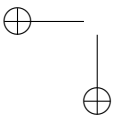
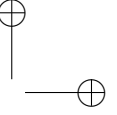
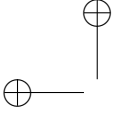


Figure 6.67: *ANT-1: setpoint tracking (1 m step) with VRFT tuning and ideal reference model response (experiment).*

6.3. Ant-1

input to the quadrotor. The chosen setpoint time history is a sequence of steps with amplitudes of 0.5 m and 1 m and duration equal to 10 seconds. Figure 6.66 shows the complete setpoint tracking test obtained operating the quadrotor with VRFT tuned attitude and position controllers. On the same figure, the ideal dynamic response of the reference model has been plotted. Figure 6.67 shows a zoomed-in view of a single step. As can be seen, recalling that the simulated step response was very similar to the reference model one, the real plant is significantly more damped than the simulated model due to the neglected aerodynamic effects.



CHAPTER 7

Conclusions

IN this thesis, data-driven methods have been applied to tune the attitude and the position controllers of UAVs. After a detailed classification of these methods, the more promising algorithms are presented. From this analysis, they emerge as a valuable approach to tune the UAV controllers. Indeed, these methods exploit only experimental input-output data and basic information on the plant, thus avoiding the requirement of an accurate plant model. Furthermore, since the considered data-driven methods are computationally efficient, they allow also a fast re-tuning of the controller when the plant performance is reduced or operating conditions change.

Three data-driven algorithms have been selected. The first is the VRFT method and it is also chosen because it was already extended in [19] to tune a cascade control system with data from a single experiment. This makes

Chapter 7. Conclusions

VRFT the best candidate to solve the tuning problem since all the considered UAVs exploit a cascade control architecture. In order to overcome the limitation of the VRFT method when the data are affected by noise, the CbT algorithm is adopted. Finally, the lack of an accurate plant model makes it impossible to guarantee the stability of the closed-loop system before implementing and testing the controller on the real plant. Recently, a new data-driven method, called controller unfalsification, has been proposed and it includes an effective stability constraint. In this work also this latter algorithm is tested.

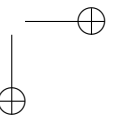
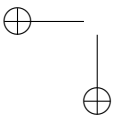
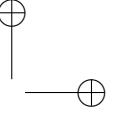
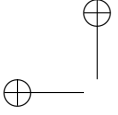
Currently, the problem of tuning the attitude control system of a rotorcraft through data-driven methods is not yet fully explored in the literature, mainly due to the complexity of performing open-loop experiments to collect the required data. To deal with this problem, in Chapter 4 some extensions to the standard data-driven methods are presented. In particular, the extensions allow to apply the algorithms on the classical control architecture of UAVs (presented in Chapter 3) and to consider data obtained with closed-loop tests performed in flight.

Three different multirotor systems have been analysed in this work: they are all quadrotors and they differ from size, actuation and control architectures. The same UAVs were previously employed in several different works both from the control and the system identification point of view. These previous works are exploited here to better tune the parameters of the data-driven methods (*e.g.*, the reference models) and to compare the new tunings obtained with the data-driven algorithms with those already implemented on the UAVs. In particular, the reference model represents the main DoF in the design procedure of all the considered data-driven methods. As explained before, the selection of the model parameters in our cases exploits the information of previous works but the required information is usually available from the plant manufacturer. Without this information, obtaining a satisfactory tuning can be challenging but, since the proposed data-driven methods are computational efficient, the selection of the reference model could be made also in an iterative way.

Exploiting the different features of each UAV and the chance to execute

specific tests, different analyses have been performed. With the first UAV, VRFT and CbT are employed and compared with a model-based and a manual tuning. The customisation of the UAV firmware was not possible but it allows us to perform an analysis with different SNR showing that CbT is more robust than VRFT. In order to show how the VRFT method is able to adapt to a different platform, VRFT has been applied to a smaller quadrotor than the first one. Also in this case the results are compared with a model-based tuning. The high level of FCU customisation and the small size of the third UAV allows us to test all the considered data-driven methods and to perform different analyses. In particular, VRFT is applied with both open-loop and closed-loop experimental data and compared with a model-based tuning. Furthermore, changing the controller architecture, VRFT has been compared with CbT and the controller unfalsification method. Finally, also the position controller of this UAV is considered.

The three quadrotors lead to similar results: the data-driven methods are successfully applied to tune a cascade control system and the designed controller provides a performance level comparable with the one of a model-based controller. In particular, the data-driven controller presents good tracking and disturbance rejection capabilities and therefore represents a viable solution for the fast deployment of high performance controllers for UAVs.



List of Figures

1.1	The control system.	6
2.1	The control system with measurement noise.	13
2.2	Cascade control scheme with two nested loops.	16
2.3	The tuning scheme for the CbT method.	18
2.4	The approximate tuning scheme for CbT method.	19
2.5	The tuning scheme for the controller unfalsification method.	25
3.1	The overall control architecture for multirotor system.	33
3.2	The controller architecture for the longitudinal/pitch DoFs.	35
3.3	The pitch attitude controller based on SISO PID architecture.	35
3.4	The pitch attitude controller with feed-forward gain and derivative action on plant measurements.	37
3.5	The longitudinal position controller with feed-forward and derivative action based on the plant measurements.	38
4.1	VRFT experiment in closed-loop operation.	42
4.2	Controller with feed forward term and action on plant measurements.	51
5.1	Aermatica P2-A1 on laboratory test-bed.	61

List of Figures

5.2	Proto-1 on its test-bed: front view.	64
5.3	Proto-1 on its test-bed: top view	64
5.4	The ANT-1 quadrotor.	67
5.5	The ANT-1 quadrotor on the test-bed.	68
6.1	Aermatica P2-A1: open-loop experimental dataset used by VRFT and CbT.	71
6.2	Aermatica P2-A1: Bode diagram of the inner and the outer reference models.	73
6.3	Aermatica P2-A1: setpoint tracking with manual tuning (experiment).	74
6.4	Aermatica P2-A1: setpoint tracking with model-based H_∞ tuning (experiment).	75
6.5	Aermatica P2-A1: setpoint tracking with VRFT tuning (experiment).	75
6.6	Aermatica P2-A1: load disturbance rejection with manual tuning (experiment).	76
6.7	Aermatica P2-A1: load disturbance rejection with model-based H_∞ tuning (experiment).	77
6.8	Aermatica P2-A1: load disturbance rejection with VRFT tuning (experiment).	77
6.9	Aermatica P2-A1: Bode diagram of the inner and the outer reference models.	79
6.10	Aermatica P2-A1: impulse response of the inner and the outer reference models.	79
6.11	Aermatica P2-A1: setpoint tracking with CbT tuning (experiment).	80
6.12	Aermatica P2-A1: setpoint tracking with VRFT tuning (experiment).	81
6.13	Aermatica P2-A1: load disturbance rejection with CbT tuning (experiment).	81
6.14	Aermatica P2-A1: load disturbance rejection with VRFT tuning (experiment).	82
6.15	Proto-1: open-loop experimental dataset used by VRFT.	84

List of Figures

6.16 Proto-1: setpoint tracking performance with VRFT tuning (experiment).	86
6.17 Proto-1: setpoint tracking performance with H_∞ tuning (experiment).	86
6.18 Proto-1: load disturbance rejection with VRFT tuning (experiment).	87
6.19 Proto-1: load disturbance rejection with H_∞ tuning (experiment).	88
6.20 ANT-1: open-loop experimental dataset used by data-driven methods.	89
6.21 ANT-1: pitch attitude model.	94
6.22 ANT-1: Bode diagram of the identified pitch attitude model.	95
6.23 ANT-1: comparison of the inner loop Bode diagrams considering VRFT, H_∞ and manual tunings (simulation).	96
6.24 ANT-1: comparison of the inner loop step responses considering VRFT, H_∞ and manual tunings (simulation)	96
6.25 ANT-1: comparison of the inner loop Bode diagrams considering VRFT, CbT and controller unfalsification tunings (simulation).	97
6.26 ANT-1: comparison of the inner loop step responses considering VRFT, CbT and controller unfalsification tunings (simulation).	97
6.27 ANT-1: impulse response of the inner and the outer reference models.	98
6.28 ANT-1: infinity norm discrepancy.	99
6.29 ANT-1: comparison of the inner loop Bode diagram considering the controller unfalsification tuning with unachievable reference model (simulation).	100
6.30 ANT-1: comparison of the inner loop step response considering the controller unfalsification tuning with unachievable reference model (simulation).	100
6.31 ANT-1: comparison of the outer loop Bode diagrams considering manual, VRFT and H_∞ tunings (simulation).	102

List of Figures

6.32 ANT-1: comparison of the outer loop step responses considering manual, VRFT and H_∞ tunings (simulation). 102

6.33 ANT-1: the outer loop Bode diagram considering the VRFT tuning (simulation). 103

6.34 ANT-1: the outer loop step response considering the VRFT tuning (simulation). 103

6.35 ANT-1: the outer loop Bode diagram considering the CbT tuning (simulation). 104

6.36 ANT-1: the outer loop step response considering the CbT tuning (simulation). 104

6.37 ANT-1: the outer loop Bode diagram considering the controller unfalsification tuning (simulation). 105

6.38 ANT-1: the outer loop step response considering the controller unfalsification tuning (simulation). 105

6.39 ANT-1: comparison of the outer loop Bode diagrams considering VRFT, CbT and controller unfalsification tunings (simulation). 106

6.40 ANT-1: comparison of the outer loop step responses considering VRFT, CbT and controller unfalsification tunings (simulation). 106

6.41 ANT-1: setpoint tracking with manual, H_∞ and VRFT tuning (experiment). 108

6.42 ANT-1: setpoint tracking (5 deg step) with manual, H_∞ and VRFT tuning (experiment). 108

6.43 ANT-1: setpoint tracking (10 deg step) with manual, H_∞ and VRFT tuning (experiment). 109

6.44 ANT-1: load disturbance rejection with manual, H_∞ and VRFT tuning (experiment). 109

6.45 ANT-1: setpoint tracking with VRFT, CbT and controller unfalsification tuning (experiment). 110

6.46 ANT-1: setpoint tracking (5 deg step) with VRFT, CbT and controller unfalsification tuning (experiment). 111

List of Figures

6.47 ANT-1: setpoint tracking (10 deg step) with VRFT, CbT and controller unfalsification tuning (experiment). 111

6.48 ANT-1: load disturbance rejection with VRFT, CbT and controller unfalsification tuning (experiment). 112

6.49 ANT-1: closed-loop experimental dataset used by data-driven methods. 113

6.50 ANT-1: comparison of the inner loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation). 115

6.51 ANT-1: comparison of the inner loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation). 116

6.52 ANT-1: comparison of the outer loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation). 116

6.53 ANT-1: comparison of the outer loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation). 117

6.54 ANT-1: setpoint tracking comparing VRFT with closed-loop data and VRFT with open-loop data (experiment). 118

6.55 ANT-1: setpoint tracking (5 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment). 118

6.56 ANT-1: setpoint tracking (10 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment). 119

6.57 ANT-1: load disturbance rejection comparing VRFT with closed-loop data and VRFT with open-loop data (experiment). 120

6.58 ANT-1: closed-loop pitch experimental dataset used by MIMO data-driven method. 122

6.59 ANT-1: closed-loop roll experimental dataset used by MIMO data-driven methods. 122

6.60 ANT-1: open-loop experimental dataset used by data-driven methods. 125

List of Figures

6.61 ANT-1: longitudinal position plant model.	127
6.62 ANT-1: comparison of the position inner loop Bode diagram considering the VRFT tuning (simulation).	128
6.63 ANT-1: comparison of the position inner loop step response considering the VRFT tuning (simulation).	128
6.64 ANT-1: comparison of the position outer loop Bode diagram considering the VRFT tuning (simulation).	129
6.65 ANT-1: comparison of the position outer loop step response considering the VRFT tuning (simulation).	129
6.66 ANT-1: position setpoint tracking with VRFT tuning and ideal reference model response (experiment).	130
6.67 ANT-1: setpoint tracking (1 m step) with VRFT tuning and ideal reference model response (experiment).	130

List of Tables

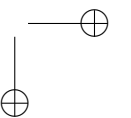
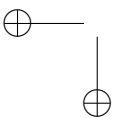
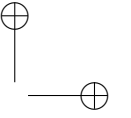
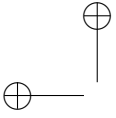
6.1	Aermatica P2-A1: optimal controller parameters considering manual, VRFT and H_∞ methods.	74
6.2	Aermatica P2-A1: optimal controller parameters considering VRFT and CbT.	80
6.3	Aermatica P2-A1: MSE of CbT and VRFT considering three different datasets.	82
6.4	Proto-1: optimal controller parameters considering H_∞ and VRFT methods.	85
6.5	Proto-1: average MSE of H_∞ and VRFT methods computed on 10 setpoint tracking tests.	85
6.6	Proto-1: average MSE of H_∞ and VRFT methods computed on 10 load disturbance rejection tests.	88
6.7	ANT-1: inner and outer reference models for VRFT, CbT and controller unfalsification algorithms.	92
6.8	ANT-1: optimal controller parameters considering manual, VRFT and H_∞ methods.	93
6.9	ANT-1: optimal controller parameters considering VRFT, CbT and controller unfalsification methods.	93

List of Tables

6.10 ANT-1: optimal controller parameters for outer and inner controllers considering the VRFT method with open-loop and closed-loop experimental data.	114
6.11 ANT-1: parameters for the initial controller exploited in the test to collect data for the MIMO controller tuning procedure.	121
6.12 ANT-1: optimal controller parameters for outer and inner controllers considering the VRFT MIMO method with closed-loop experimental data.	123
6.13 ANT-1: optimal controller parameters for outer and inner position controllers considering the VRFT method.	126

List of Algorithms

1	The VRFT algorithm.	15
2	The VRFT method for cascade control systems.	17
3	The CbT algorithm.	21
4	Control unfalsification tuning method.	28
5	The control system with closed-loop excitation data.	50
6	VRFT algorithm with the new controller architectures.	56
7	The CbT method for cascade control systems.	58



Glossary

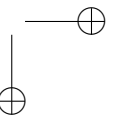
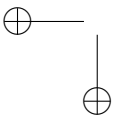
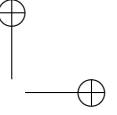
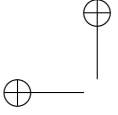
CbT	Correlation based Tuning. 10, 11, 17–19, 21, 23, 39, 54, 55, 60, 61, 67–69, 76, 78–81, 89–92, 94–96, 101–106, 108–110, 132–137, 139, 141
DoF	Degree of Freedom. 32, 33, 36, 68, 81, 118, 119, 121, 133
FCU	Flight Control Unit. 31, 32, 35, 61, 65, 68, 81, 87, 89, 122
IFT	Iterative Feedback Tuning. 10
IMU	Inertial Measurement Unit. 61, 68, 87
LAN	Local Area Network. 65
LTI	Linear Time Invariant. 43
MAV	Micro Aerial Vehicle. 64, 66

Glossary

- MIMO Multiple Input Multiple Output. 15, 43, 87, 119, 120, 140
- MISO Multiple Input Single Output. 48
- MSE Mean Square Error. 80, 81, 83, 85, 86, 139
- OGE Out of Ground Effect. 58
- PBSID Predictor Based System Identification. 43, 44, 47, 48, 92, 112
- PEM Prediction Error Method. 13, 17
- PRBS Pseudo Random Binary Sequence. 60, 68, 80, 81, 87, 111, 119, 122, 123
- PWM Pulse Width Modulation. 61
- R2P Rapid Robot Prototyping. 61, 63
- ROS Robotic Operating System. 63
- SISO Single Input Single Output. 6, 15, 29, 33, 48, 89, 91, 101, 119
- SMI Subspace Model Identification. 43
- SNR Signal to Noise Ratio. 21, 61, 67, 78, 80, 81, 94
- TOW Take-Off Weight. 5
- UAV Unmanned Aerial Vehicle. I, 2, 5, 7, 8, 10, 29–32, 34, 35, 39–41, 48, 57, 58, 60, 61, 63–68, 81, 87, 105, 111, 122, 123, 131, 132

Glossary

VRFT Virtual Reference Feedback Tuning. 10–15, 17–20, 23, 24, 39–41, 47–49, 52–54, 60, 61, 63, 64, 66–72, 74–76, 78–87, 89–97, 99–102, 104–116, 118, 119, 121, 122, 124–129, 131–141



Bibliography

- [1] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Isa Research Triangle Park, NC, 1995.
- [2] Federal Aviation Authority. <https://registermyuas.faa.gov/>.
- [3] G. Battistelli, D. Mari, D. Selvi, and P. Tesi. Direct control design via controller unfalsification. *International Journal of Robust and Nonlinear Control*, 2017. mc.3778.
- [4] M. Bergamasco and M. Lovera. Continuous-time predictor-based subspace identification using Laguerre filters. *IET Control Theory and Applications*, 5(7):856–867, 2011. Special issue on Continuous-time Model Identification.
- [5] M. Bergamasco and M. Lovera. Identification of linear models for the dynamics of a hovering quadrotor. *IEEE Transactions on Control Systems Technology*, 22(5):1696–1707, Sept 2014.
- [6] A. Bonarini, M. Matteucci, M. Migliavacca, and D. Rizzi. R2p: An open source hardware and software modular approach to robot prototyping. *Robotics and Autonomous Systems*, 62(7):1073 – 1084, 2014. Reconfigurable Modular Robotics.
- [7] C. L. Bottasso, D. Leonello, and B. Savini. Path planning for autonomous vehicles by trajectory smoothing using motion primitives. *IEEE Transactions on Control Systems Technology*, 16(6):1152–1168, 2008.
- [8] S. Bouabdallah, A. Noth, and R. Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2451–2456. IEEE, 2004.
- [9] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2247–2252. IEEE, 2005.

Bibliography

- [10] M.C. Campi, A. Lecchini, and S.M. Savaresi. Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8):1337 – 1346, 2002.
- [11] E. Capello, A. Scola, G. Guglieri, and F. Quagliotti. Mini quadrotor uav: design and experiment. *Journal of Aerospace Engineering*, 25(4):559–573, 2012.
- [12] D. Chevallard. Design, identification and control of a micro aerial vehicle. Master’s thesis, Politecnico di Milano, 2017.
- [13] A. Chiuso. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 43(6):1034 – 1048, 2007.
- [14] T. Chupin. Data-driven attitude control design for multirotor uavs. Master’s thesis, Politecnico di Milano, 2017.
- [15] M. Cutler, N. Kemal Ure, B. Michini, and P. How. Comparison of fixed and variable pitch actuators for agile quadrotors. In *AIAA Guidance, Navigation, and Control Conference (GNC)*, volume 2, 2011.
- [16] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory Applications*, 3(3):303–314, March 2009.
- [17] L. Derafa, T. Madani, and A. Benallegue. Dynamic modelling and experimental identification of four rotors helicopter parameters. In *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pages 1834–1839. IEEE, 2006.
- [18] ENAC. Remotely piloted aerial vehicle regulation. https://www.enac.gov.it/la_normativa/normativa_enac/.
- [19] S. Formentin, A. Cologni, D. Belloli, F. Previdi, and S. M. Savaresi. Fast tuning of cascade control systems. *IFAC Proceedings Volumes*, 44(1):10243–10248, 2011.
- [20] S. Formentin and M. Lovera. Flatness-based control of a quadrotor helicopter via feedforward linearization. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 6171–6176. IEEE, 2011.
- [21] 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*, 28(10):882–897, 2014.
- [22] M. Giurato. Design, integration and control of a multirotor uav platform. Master’s thesis, Politecnico di Milano, 2015.
- [23] 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*, 45(5):954–959, May 2000.
- [24] P. G. Hamel and J. Kaletka. Advances in rotorcraft system identification. *Progress in Aerospace Sciences*, 33(3):259 – 284, 1997.
- [25] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Systems*, 18(4):26–41, Aug 1998.

Bibliography

- [26] H. Hjalmarsson, S. Gunnarsson, and M. Gevers. A convergent iterative restricted complexity control design scheme. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 2, pages 1735–1740. IEEE, 1994.
- [27] Z. S. Hou and Z. Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235(Supplement C):3 – 35, 2013. Data-based Control, Decision, Scheduling and Fault Diagnostics.
- [28] I. Houtzager, J. W. van Wingerden, and M. Verhaegen. Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology*, 20(4):934–949, 2012.
- [29] R. Jategaonkar. *Flight vehicle system identification: a time domain methodology*, volume 216. AIAA, Reston, VA, USA, 2006.
- [30] A. Karimi, K. Van Heusden, and D. Bonvin. Non-iterative data-driven controller tuning using the correlation approach. In *Control Conference (ECC), 2007 European*, pages 5189–5195. IEEE, 2007.
- [31] M. La Civita. *Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters*. PhD thesis, Carnegie Mellon University, 2003.
- [32] A. Lecchini, M. Campi, and S. M. Savaresi. Virtual reference feedback tuning for two degree of freedom controllers. 16:355 – 371, 06 2002.
- [33] T. Lee, M. Leoky, and H. McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5420–5425. IEEE, 2010.
- [34] Leishman, J. G. *Principles of Helicopter Aerodynamics, 2nd ed.* Cambridge University Press, New York, NY, 2006.
- [35] P. Li, I. Postlethwaite, and M. C. Turner. Subspace-based system identification for helicopter dynamic modelling. 3, 01 2007.
- [36] L. Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.
- [37] M.A. Lotufo, L. Colangelo, C. Perez-Montenegro, C. Novara, and E. Canuto. Embedded model control for uav quadrotor via feedback linearization. *IFAC-PapersOnLine*, 49(17):266 – 271, 2016. 20th IFAC Symposium on Automatic Control in Aerospace ACA 2016.
- [38] R. Lozano, P. Castillo, and A. Dzul. Modeling and control of mini-flying machines, 2005.
- [39] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept 2012.
- [40] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.
- [41] M. Nakamoto. An application of the virtual reference feedback tuning for an mimo process. In *SICE 2004 Annual Conference*, volume 3, pages 2208–2213 vol. 3, Aug 2004.

Bibliography

- [42] P. Panizza, F. Riccardi, and M. Lovera. Black-box and grey-box identification of the attitude dynamics for a variable-pitch quadrotor. In *1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles ACNAAV*, 2015.
- [43] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010.
- [44] F. Riccardi. *Model Identification and Control of Variable Pitch Quadrotor UAVs*. PhD thesis, Politecnico di Milano, 2015.
- [45] F. Riccardi, M. F. Haydar, S. Formentin, and M. Lovera. Control of variable-pitch quadrotors. *IFAC Proceedings Volumes*, 46(19):206–211, 2013.
- [46] F. Riccardi, P. Panizza, and M. Lovera. Identification of the attitude dynamics for a variable-pitch quadrotor uav. In *40th European Rotorcraft Forum, Southampton, UK*, 2014.
- [47] Riccardi, F. and Lovera, M. Robust attitude control for a variable-pitch quadrotor. In *IEEE Conference on Control Applications, Antibes, France*, pages 730–735, 2014.
- [48] A. Sala and A. Esparza. Extensions to virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica*, 41(8):1473–1476, 2005.
- [49] M. B. Tischler and R. K. Remple. Aircraft and rotorcraft system identification. *AIAA education series*, 2006.
- [50] G. van der Veen, J. W. van Wingerden, Bergamasco M., Lovera M., and M. Verhaegen. Closed-loop subspace identification methods: an overview. *IET Control Theory and Applications*, 7(10):1339–1358, 2013.
- [51] K. Van Heusden, A. Karimi, and D. Bonvin. Data-driven controller tuning with integrated stability constraint. In *2008 47th IEEE Conference on Decision and Control*, pages 2612–2617, Dec 2008.
- [52] 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*, 25(4):331–351, 2011.
- [53] J. Wartmann and S. Seher-Weiss. Application of the predictor-based subspace identification method to rotorcraft system identification. In *39th European Rotorcraft Forum (ERF)*, September 2013.