Politecnico di Milano

Scuola di Ingegneria Industriale e dell'Informazione

Corso di Laurea Magistrale in Ingegneria Aeronautica

Dipartimento di Scienze e Tecnologie Aerospaziali

Data-driven attitude and position control design for Micro Aerial Vehicles

Advisor: Prof. Marco LOVERA
Co-Advisor: Eng. Pietro PANIZZA

Thesis by:
Stefano CAPOCCHIANO Matr. 842067

Academic Year 2016–2017

Alla mia famiglia, che mi è sempre stata vicina.

Ringraziamenti

Desidero innanzitutto ringraziare il professor Marco Lovera per la disponibilità che mi ha sempre dimostrato e per la fiducia che ha riposto in me nell'affidarmi questo progetto. Lo ringrazio inoltre per aver stimolato il mio interesse nell'argomento, nonchè quella curiosità senza la quale questa tesi non sarebbe stata possibile.

Vorrei dedicare un ringraziamento particolare a Pietro Panizza che mi ha assistito fin dal primo giorno. In particolare, vorrei ringraziarlo per l'infinita disponibilità e pazienza che mi ha sempre riservato, nonché per l'altruismo dimostrato in più occasioni. Il suo preziosissimo aiuto e le conoscenze che mi ha trasmesso, più di ogni altra cosa, sono state essenziali per raggiungere questo risultato.

Infine, vorrei esprimere la mia profonda gratitudine alla mia famiglia, che in questi anni di studio mi ha sempre sostenuto, moralmente e materialmente. Il raggiungimento di questo mio traguardo lo devo anche a loro.

Abstract

The definition of flight control laws is one of the most demanding activities in the process of designing remote-controlled multirotor systems. Typically, the controller is synthesised by means of techniques that rely on a prior modelling of the real plant, which however frequently neglects some significant dynamic effects. To ensure robustness with respect to model uncertainties, a clear degradation of controlled system performance in favour of stability is necessary. Data-driven tuning methods, in contrast, do not require an accurate knowledge of the model, thus eliminating the problem of under-modelling and therefore representing an efficient alternative in the design of flight control laws. The purpose of this thesis is to extend the data-driven methods available in the literature to tune both the attitude and the position PID controllers of a small-scale quadrotor of the Micro Aerial Vehicles (MAVs) class. In particular, the data-driven methods discussed in the literature present some limitations that hinder their diffusion in the multirotor design framework and make them totally unsuitable for any potential application in the helicopter industry. As part of the present work, it has been shown that a data-driven method guaranteeing a-priori closed-loop stability can be effectively implemented, thereby making the subsequent experimental validation test virtually risk-free. In addition, by extending the Virtual Reference Feedback Tuning (VRFT) algorithm, it has been possible to apply it to closed-loop collected data. The resulting controller showed similar performance to that obtained with open-loop collected data, while ensuring a significant simplification of the tuning process. Finally, experimental tests have shown that data-driven methods can provide performance comparable to model-based methods such as the H_∞, but requiring only one experimental test. These results pave the way for the potential adoption of the proposed tuning approaches in the helicopter industry.

Sommario

La definizione delle leggi di controllo è una delle attività più onerose nel progetto di sistemi multirotore a pilotaggio remoto. La sintesi del controllore è tipicamente affidata a tecniche basate su una preventiva modellazione del sistema reale, la quale però spesso trascura importanti dinamiche. Dovendo garantire robustezza rispetto a tali incertezze di modello, si impone un decisivo degrado delle prestazioni del sistema controllato a favore della stabilità. Le tecniche di taratura data-driven, al contrario, non richiedono una conoscenza accurata del modello, eliminando dunque il problema delle dinamiche non modellate e rappresentando perciò una valida alternativa nella progettazione delle leggi di controllo. Lo scopo del presente lavoro di tesi è quello di estendere le tecniche data-driven presenti in letteratura, al fine di tarare i controllori PID di assetto e posizione di un quadrirotore appartenente alla categoria dei Micro Aerial Vehicles (MAVs). In particolare, i metodi data-driven discussi in letteratura presentano limitazioni che ne ostacolano la diffusione nel progetto di multirotori e li rendono totalmente inadatti a un'eventuale applicazione in ambito elicotteristico. Nell'ambito del lavoro svolto, è stato dimostrato come sia possibile realizzare un controllore mediante tecnica data-driven che garantisca, a priori, la stabilità del sistema controllato, rendendo virtualmente priva di rischi la prova sperimentale di validazione. Inoltre, mediante estensione dell'algoritmo VRFT (Virtual Reference Feedback Tuning), è stato possibile applicarlo, per la prima volta, a dati raccolti in anello chiuso. Il controllore così realizzato ha dimostrato prestazioni paragonabili alla controparte ottenuta con dati raccolti in anello aperto, con notevole semplificazione del processo di taratura. Infine, le prove sperimentali effettuate hanno dimostrato che, mediante metodi data-driven, è possibile ottenere prestazioni paragonabili a quelle proprie dei metodi model-based come l'H_∞, richiedendo però una sola prova sperimentale. Tutto ciò apre di fatto la strada al potenziale impiego del metodo di taratura proposto in campo elicotteristico.

Contents

List of Figures

List of Tables

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 setpoint signal or to satisfy a design target.

In the *model-based* approach a mathematical model of the plant is required 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 [1, 2], 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 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 true 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, this methods is inherently less safe and less robust due to the unmodelled dynamics. *Robust control theory* was born to deal with this kind of problems including additive and multiplicative descriptions and the assumption of bound 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 convergence and robustness of the closed-loop system are not always valuable.

Since in this work control theory is applied to an UAV, 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.*, [3, 4, 5] 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 as 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 one. 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.

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-modelling and to facilitate the design of fixed-order controllers. Data-driven algorithms skip the modelling phase almost entirely and instead reformulate the controller tuning 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 [6]). Several data-driven control design algorithms have been proposed recently. Compared to the work in [7] 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 in this work are also computationally 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 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 unmodelled dynamics, the traditional definition of robustness is no longer valid.

At this point where the data-driven framework is introduced, the reader might wonder if data-driven methods perform better than model-based methods. Recalling the results in [8], 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 [8] and the references therein). Nevertheless, the previous criterion represents only an intermediate step toward the evaluation of the methods. As will be explained in the next chapter, 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, 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-modelling can not be avoided with a low-order model, the data-driven approach may give better results in real applications.

The previous considerations are the conclusion of [8] 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 parametrised. Finally, the data-driven approach leads to a convex optimisation problem if the controller is linearly parametrised whereas the model-based approach requires that both the controller and the plant model are linearly parametrised.

In recent the literature, the validity of data-driven methods for UAV control law tuning has been verified. In particular, in [9, 10] the data-driven tuning approach has been applied to a large multirotor platform (5 kg Take-Off Weight (TOW)) characterised by attitude control via collective blade pitch. In [11] similar analyses have been carried out on a medium-sized quadrotor (1.5 kg TOW) with variable-speed, fixed-pitch rotors. The main purpose of this thesis is to extend the analyses made so far by considering a Micro Aerial Vehicle (MAV), *i.e.* a small UAV with a TOW of about 200 g. Besides validating previous results, the following objectives have been pursued and successfully achieved:

- validate a data-driven algorithm with a stability constraint for attitude control;
- exploit the attitude controller to achieve position tracking via a cascade control paradigm and tune the outer-loop position controller by means of a data-driven algorithm;
- extend a data-driven algorithm to deal with data obtained from flight tests performed in closed-loop conditions.

Structure of the thesis

This thesis is organised as follows:

- **Chapter 2: Overview of data-driven tuning methods.** An overview of model reference control and data-driven methods is provided and the most promising algorithms are illustrated in detail.

- **Chapter 3: UAV control architectures.** After providing an explanation of how the overall control system of a generic multirotor platform works, the control architectures of the quadrotor adopted in this thesis are shown. In particular, the unknown controller parameters that need to be tuned are illustrated for each control architecture.

- **Chapter 4: Data-driven algorithms for multirotor systems.** All the extensions of data-driven methods, made as part of this thesis, are discussed. The main extension involves the possibility to tune the controller with experimental data obtained from closed-loop flight tests. Furthermore, the methods presented in Chapter 2 are extended to deal with the control problem illustrated in Chapter 3.

- **Chapter 5: The ANT-1 MAV platform.** After a brief overview of MAVs, the considerd multirotor platform is presented in detail. In the last part of this chapter, the test-bed is described.

- **Chapter 6: Results.** In this chapter all the results are thoroughly discussed. First, all the data-driven tunings of the attitude controller are compared with each other and with pre-existing model-based and manual tunings. Then, the results obtained with closed-loop collected data are compared with that of the open-loop ones. Finally, the tracking performance of the data-driven tuned position controller is shown.

- **Chapter 7: Conclusions and future developments.** Some considerations on the results are drawn and suggestions for future developments are provided.

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 it recently 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-structure controller, others implicitly involve the plant model structure and lead to a controller structure that is not fixed *a priori* (see [6]). The first class of data-driven methods is considered in this work since the main goal is to tune controllers with a fixed structure. Indeed, in most practical cases, changing the controller structure is not feasible (*e.g.*, the controller source code is not available).

Furthermore, in the UAV considered in this work, pre-existing controller tunings obtained with model-based and manual methods were available, so the new data-driven tunings have been compared with the previously defined ones without changing the controller architecture. The determination of the control structure goes beyond the scope of this work, but an alternative PID structure had to be implemented in order to execute and compare all the algorithms described below.

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 [12, 13] 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. Due to all these significant drawbacks the IFT method was not considered in this thesis.

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). Virtual Reference Feedback Tuning (VRFT) and Correlation-based Tuning (CbT) belong to this class and they are presented in detail in the next sections. Furthermore, in [14], 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 the UAV exploits cascade control architectures, as will be illustrated in Chapter 3.

Since the data-driven methods do not explicitly involve the plant model, 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 an the actual input-output behaviour. CbT, as presented in [15], 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 [15] 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 [16]). It is a non-iterative data-driven control design approach that incorporates stability tests originally introduced for the unfalsified control framework (see again [16]). 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 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 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.

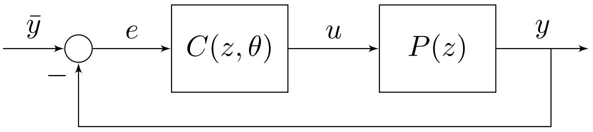


Figure 2.1: The control system.

Consider the closed-loop system shown in Figure 2.1 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 function is formulated with the control cost criterion in the following:

$$J_{MR}(\theta) = \left\| \left(\frac{P(z)C(z,\theta)}{1+P(z)C(z,\theta)} - M(z) \right) W(z) \right\|_2^2 \tag{2.1}$$

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 (2.1) exists and is given by:

$$\bar{C}(z) = \frac{M(z)}{P(z)\,(1-M(z))}. \tag{2.2}$$

In order to obtain the optimal controller in (2.2), 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 (2.1) 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 Chapter 1, this model may be derived exploiting different identification techniques but unmodelled dynamics always exist. Note that the choice of a high order model is not a viable solution. Indeed, the order of the ideal controller in (2.2) 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 model $M(z)$.

2.2 Virtual Reference Feedback Tuning

In this section the VRFT is introduced. This method was presented in [17, 18] 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 (2.1) can be reformulated as a parameter identification problem on the controller.

The main features of VRFT are that the model-reference problem (2.1) 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 used to generate $y(t)$ and it can be computed from the output data as

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

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:

$$J_{VR}^{N}(\theta) = \frac{1}{N}\sum_{t=1}^{N}\left(u_L(t) - C(z,\theta)e_L(t)\right)^2, \tag{2.3}$$

where $u_L(t)$ and $e_L(t)$ are suitably filtered versions of $u(t)$ and $e(t)$ such that the cost function (2.3) is a local approximation of the criterion (2.1) in the neighbourhood of the minimum point. As explained in [17], the optimal choice of the filter $L(z)$ is

$$|L|^2 = |1 - M|^2|M|^2|W|^2\frac{1}{\Phi_u}, \quad \forall\omega \in [-\pi, \pi], \tag{2.4}$$

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 model matching problem (this controller is defined in (2.2)).

Note that if the user considers a linearly parametrised controller class $\mathcal{C}(\theta) = \left\{C(z,\theta) = \beta^T(z)\theta\ ,\ \theta \in R^n\right\}$, the criterion in (2.3) can be rewritten as

$$J_{VR}^{N}(\theta) = \frac{1}{N}\sum_{t=1}^{N}\left(u_L(t) - \varphi_L^T(t)\theta\right)^2, \tag{2.5}$$

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

$$\hat{\theta}_N = \arg\min_{\theta} J_{VR}^{N}(\theta). \tag{2.6}$$

The closed form solution of the problem in (2.6) exists and it is:

$$\hat{\theta}_N = \left[\sum_{t=1}^{N}\varphi_L(t)\varphi_L^T(t)\right]^{-1}\sum_{t=1}^{N}\varphi_L(t)u_L(t). \tag{2.7}$$

Since VRFT exploits a Predictor Error Method (PEM) identification procedure

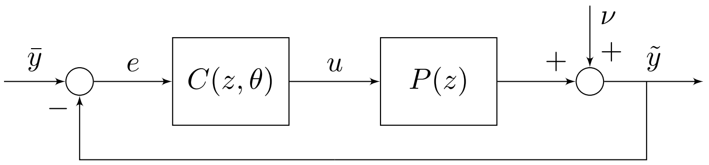


Figure 2.2: The control system with measurement noise.

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 2.2):

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

with the assumption that $u(t)$ and $\nu(t)$ are uncorrelated. In this case the 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 [17], 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

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

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

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

> 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 in the creation of the instrumental variable.

When the instrumental variable is selected, it can be used to solve the problem in (2.5) and the optimal solution is:

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

The VRFT procedure is summarised in Algorithm 1.

Algorithm 1 The VRFT algorithm.

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

2.2.1 Cascade control systems

It has been shown in [14] 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 2.3 where only two loops are shown without loss of generality and a noiseless environment is considered (see [14] for a deeper presentation of the methodology). Given two reference models $M_i(z)$ and

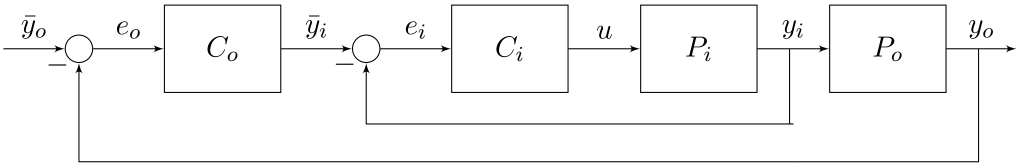


Figure 2.3: Cascade control scheme with two nested loops.

$M_o(z)$, for the inner loop and the outer loop respectively, consider two families of linear proper controllers $\mathcal{C}_i(\theta_i) = \{C_i(z, \theta_i)\ ,\ \theta_i \in R_i^n\}$ and $\mathcal{C}_o(\theta_o) = \{C_o(z, \theta_o)\ ,\ \theta_o \in R_o^n\}$ 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 2.2. 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 2.3), that is not available in the dataset, since measurements are collected during an open-loop experiment.

Nevertheless, in [14] 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 a 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 of

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

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 [14], following the rationale of [17].

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

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

```
1: Compute $U(z)$ such that $\left|U\left(e^{j\omega}\right)\right|^2 = \Phi_u(\omega)$.
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 $e_{iL}(t) = L_i(z)\left(M_i^{-1}(z) - 1\right) y_i(t)$.
6:     Compute $\hat{\theta}_i = \arg\min_{\theta_i} \frac{1}{N} \sum_{t=1}^{N} \left(u_L(t) - C_i(z,\theta_i) e_{iL}(t)\right)^2$.
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 $\left|U_o\left(e^{j\omega}\right)\right|^2 = \Phi_{\bar{y}_i}(\omega)$ where $\Phi_{\bar{y}_i}(\omega)$ is the power 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 $e_{oL}(t) = L_o(z)\left(M_o^{-1}(z) - 1\right) y_o(t)$.
13: Compute $\hat{\theta}_o = \arg\min_{\theta_o} \frac{1}{N} \sum_{t=1}^{N} \left(\bar{y}_{iL}(t) - C_o(z,\theta_o) e_{oL}(t)\right)^2$.
```

2.3 Correlation-based Tuning

As explained in Section 2.2, 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 [15, 19, 20]). 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 (2.1). 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 (2.1) to a constrained optimisation problem. CbT is also able to deal with nonminimum-phase or unstable plants (see again [15]).

In the CbT approach, the optimal controller is computed exploiting the error $\varepsilon(t, \theta)$ as depicted in Figure 2.4. Indeed, the minimisation of (2.1) is equivalent to minimising 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 parametrised controller, an approximation of the cost criterion

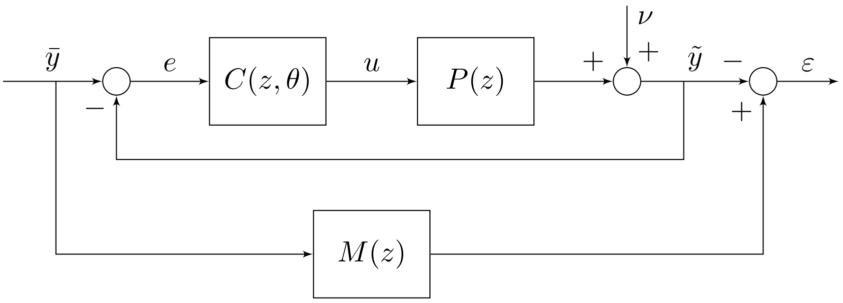


Figure 2.4: The tuning scheme for the CbT method.

defined in (2.1) can be made in order to make it convex. In particular the approximation considers the sensitivity function: the ideal sensitivity function is given by

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

where $\bar{C}(z)$ is the optimal controller that realises $M(z)$ as defined in (2.2). 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 (2.1) becomes convex and the tuning scheme for the CbT method can be redrawn as in Figure 2.5 (see [15] 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 [17]).

Considering the Figure 2.5 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 $\nu(t)$:

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

If the optimal controller defined in (2.2) is considered, the error in (2.8) becomes:

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

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 [15] for

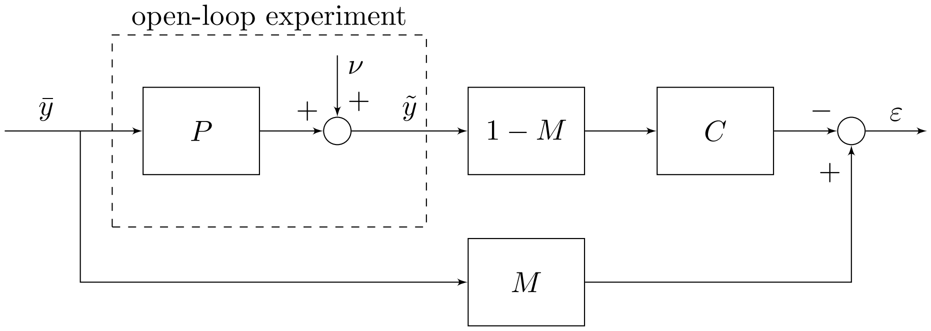


Figure 2.5: The approximate tuning scheme for the CbT method.

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:

$$\varsigma(t) = \begin{bmatrix} \bar{y}_F(t+l) & \dots & \bar{y}_F(t) & \dots & \bar{y}_F(t-l) \end{bmatrix}^T \tag{2.9}$$

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

$$|F|^2 = |1-M|^2|W|^2\frac{1}{\Phi_{\bar{y}}}, \quad \forall\omega \in [-\pi,\pi]$$

where $\Phi_{\bar{y}}$ is the spectral density of the reference signal $\bar{y}(t)$. See [15] 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

$$\hat{\theta}_N = \arg\min_{\theta} J_{N,l}(\theta). \tag{2.10}$$

The optimal parameters in (2.10) asymptotically converge to the optimiser of (2.1) (the proof is provided in [15]).

The choice of the parameter l in (2.9) is now discussed. This parameter represents a trade-off between accuracy and bias: l must be large enough to minimise (2.1) using (2.10) but the bias due to noise increases with l (see [15]). As proposed in [8], 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 (2.1) as the bias due to noise is negligible thanks to the high SNR.

The CbT method is summarised in Algorithm 3.

Algorithm 3 The CbT algorithm.

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

2.4 Controller unfalsification

As already discussed, data-driven methods are based on the assumption that the model of the plant is not known, or at least not accurately enough. 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 *controller unfalsification*, has been presented in [16]. The proposed method is based on the unfalsified control theory. As suggested in [16], 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.

2.4.1 Problem formulation

As in any 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 [16], 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 input sensitivity function. As will be shown below, one can ensure internal stability of the closed-loop by minimising the discrepancy between input 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 [16]. This allows one to define a single objective problem based on the minimisation of a cost function of the form

$$J(\theta) = (1-\delta)J_n(\theta) + \delta J_v(\theta), \tag{2.11}$$

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

Controller structure

The unfalsified control, as well as the 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

$$C(z,\theta) = \frac{\bar{N}(z,\theta)\ N^*(z)}{\bar{D}(z,\theta)\ D^*(z)}, \tag{2.12}$$

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 [16], 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 input 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 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

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

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.

2.4.2 Optimization 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:

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

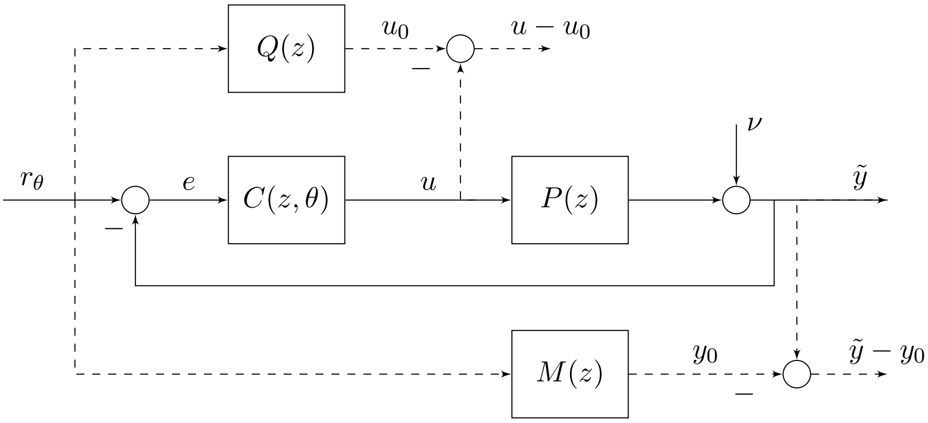


Figure 2.6: The tuning scheme for the controller unfalsification method.

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 [21], which is based on the inverse controller identification.

The fictitious reference $r_\theta(t)$ can be adopted as the input of the reference models to obtain the ideal control variable and the ideal output:

$$u_0(t) = Q(z)r_\theta(t) \tag{2.13}$$

$$y_0(t) = M(z)r_\theta(t). \tag{2.14}$$

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 2.6. By replacing the definition of fictitious reference $r_\theta(t)$ in equations (2.13) and (2.14), the input-output discrepancies can be written as

$$u(t) - u_0(t) = u(t) - Q(z)\tilde{y}(t) - C^{-1}(z,\theta)Q(z)u(t) \tag{2.15}$$

$$\tilde{y}(t) - y_0(t) = (1 - M(z))\tilde{y}(t) - C^{-1}(z,\theta)M(z)u(t). \tag{2.16}$$

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 (2.15) and (2.16) 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 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 (2.15) and (2.16):

$$\begin{aligned} J_n &= \|(u - u_0)\|^2 \\ J_v &= \|(\tilde{y} - y_0)\|^2. \end{aligned}$$

In conclusion, to obtain a single objective minimisation problem, these two cost functions have to be combined together, as previously disclosed in (2.11), resulting in the following design criterion:

$$\hat{\theta}(\delta) = \arg\min_{\theta \in \Theta_S} \{(1 - \delta)J_n(\theta) + \delta J_v(\theta)\} \tag{2.17}$$

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.

2.4.3 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 input 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

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

guarantees stability of the closed-loop system. Consequently, when $\delta = 0$ the minimisation of criterion (2.17) 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*

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

where

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

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 [16].

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

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

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

Finally, as suggested in [16], the H_∞ norm can be effectively estimated relying only on the gathered dataset and input discrepancy. So, it can be rewritten as

$$\|Q(z)\Delta_Q(z,\theta)\|_\infty \simeq \sup \frac{|\hat{u}(\omega) - \hat{u}_0(\omega)|}{|\hat{u}(\omega)|}, \tag{2.18}$$

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 (2.17) 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 (2.17) 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 error related to a finite dataset and the presence of the disturbance.

To sum up, the tuning procedure developed in [16] is shown in Algorithm 4 in a synthetic way.

Algorithm 4 Controller unfalsification tuning method.

```
repeat
    Compute $\hat{\theta}(\delta) = \arg\min_{\theta\in\Theta_S}\{(1-\delta)J_n(\theta) + \delta J_v(\theta)\}$, with $\delta = 0$.
    Compute $u - u_0$ and estimate $\|Q(z)\Delta_Q(z,\hat{\theta}(0))\|_\infty$.
until $\|Q(z)\Delta_Q(z,\hat{\theta}(0))\|_\infty < 1-\alpha$ otherwise change $M(z)$.
repeat
    Set $\delta = 1/2^{iter-1}$.
    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;
    Compute $u - u_0$ and estimate $\|Q(z)\Delta_Q(z,\hat{\theta}(\delta))\|_\infty$.
until $\|Q(z)\Delta_Q(z,\hat{\theta}(0))\|_\infty < 1-\alpha$.
```

Chapter 3

UAV control architectures

In this chapter, the control architectures implemented on the under examination multirotor system are presented. In particular, two different control structures have been considered so that all the data-driven algorithms described in Chapter 2 can be used. Indeed, the standard VRFT, CbT and controller unfalsification methods require a SISO-type controller, but the VRFT algorithm has been extended to deal with a Multiple-Input Single-Output (MISO) control structures (see Chapter 4) in order to compare the results with pre-existing H_∞ and manual tunings. For each control architecture the controller transfer function and the related unknown controller parameter vector will be outlined in the next sections. All the results of the controller tunings will be widely discussed in Chapter 6.

The chapter is organised as follows. Section 3.1 outlines the overall control architecture of the quadrotor that includes both the position and the attitude controllers. Then, the MISO control structure is described, focusing on the attitude control loop (Section 3.2) and on the position control loop (Section 3.3). Finally, the second control structure, consisting in a SISO PID cascade controller, is illustrated in Section 3.4. Notice that, for this control architecture, only the attitude controller is provided.

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 strong 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 [22], against the more modern LQ technique, as discussed in, *e.g.*, [23]. The state-of-the-art in linear control for small scale helicopters is given by approaches such as, *e.g.*, [24], in which modern robust control design is coupled with identification of linear rotorcraft models. A nonlinear control method with proven effectiveness for underactuated system is the backstepping, see for example [25], 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 [26], adaptive control [27, 28], dynamic inversion [29] and feedback linearisation [30]. Of particular interest are methods for planning and tracking based on the flatness property of quadrotor dynamics (see, *e.g.*, [31, 32]) as well as procedures based on smoothing of a given trajectory using motion primitives (see for example [33]).

With regard to the design criteria of multirotor platforms, the most common 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. This solution is adopted on the considered quadrotor (see Chapter 5), whose very small size does not allow to rely on the more complex and uncommon solution which consists in variable rotor blade collective pitch and fixed rotors angular rates. This latter design scheme allows extremely fast changes in thrust, even though the larger weight and complexity makes it a viable solution only for large multirotors. For this reason, almost all Micro Aerial Vehicles (MAVs) adopt a fixed-pitch solution so, from here on, only this design scheme will be considered.

The overall control architecture of the UAV considered in this work consists of a hierarchical control approach with an outer loop that controls the position of the UAV and generates the attitude setpoint for an inner loop, which is responsible for the attitude control of the quadrotor (see Figure 3.1). The position and the attitude controllers are not independent of each other because the system is underactuated, which means for example that, to move the UAV forward, a pitch attitude change has to take place. Furthermore, the position and the attitude measurements come from a Kalman-based estimator already implemented in the Flight Control Unint (FCU) installed on board.

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

$$X(t) = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T$$

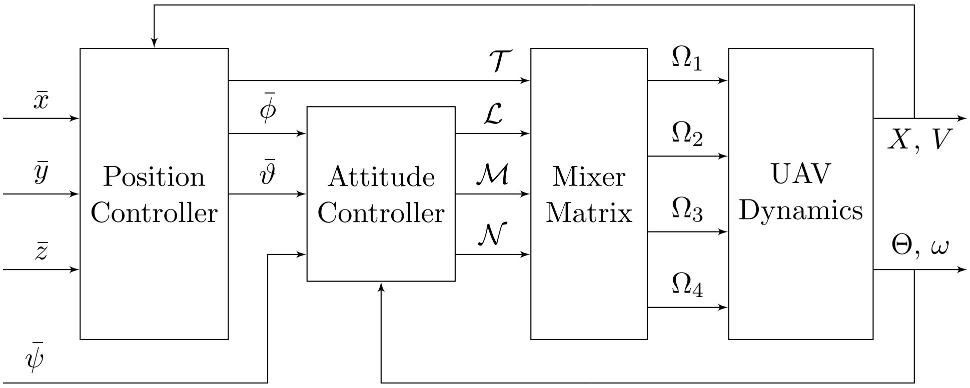


Figure 3.1: The overall control architecture for a fixed-pitch multirotor system.

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

$$V(t) = \begin{bmatrix} v_x(t) & v_y(t) & v_z(t) \end{bmatrix}^T$$

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 trust $\mathcal{T}(t)$ 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 is a matrix that relates the required thrust and moments about each axis to the control inputs of the UAV. In case of fixed blade pitch multirotors, the outputs of this block, so the control inputs, consist of the four rotational speeds of the propellers (Ω_i, $i = 1, \ldots, 4$). Whereas, in variable-pitch UAVs, the control inputs consist of the collective pitch angles of each propeller.

The control architecture was already implemented in the FCU of the tested UAV.

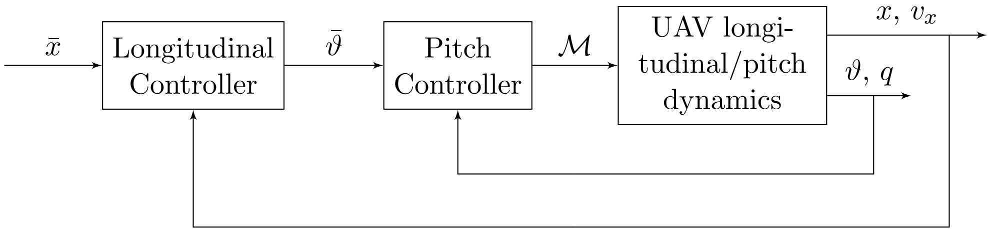


Figure 3.2: The controller architecture for the longitudinal/pitch DoFs.

The structure of the controllers was designed assuming decoupled dynamics between the DoFs. This assumption is valid only if the UAV is considered in near-hovering conditions, and falls progressively as the translational speed 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 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 have been 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 MISO pitch attitude controller

The pitch attitude controller implemented on the FCU of the tested UAV is not a simple SISO PID controller, but rather it is an alternative architecture more commonly adopted in the industrial field. The standard VRFT algorithm, described in Section 2.2, is not intended to tune multiple-input controllers. Nevertheless, instead of simplifying the control structure in order to match the architecture presented in Section 3.4, the VRFT method has been extended (see Section 4.2) to deal with the pre-existing MISO control structure. By doing so, the dynamic response of the system tuned with the VRFT method can be directly compared with that of pre-existing manual and H_∞ tunings. Hereafter, the control architecture actually implemented on the UAV is illustrated.

The attitude control architecture is based on two cascaded PID loops. The

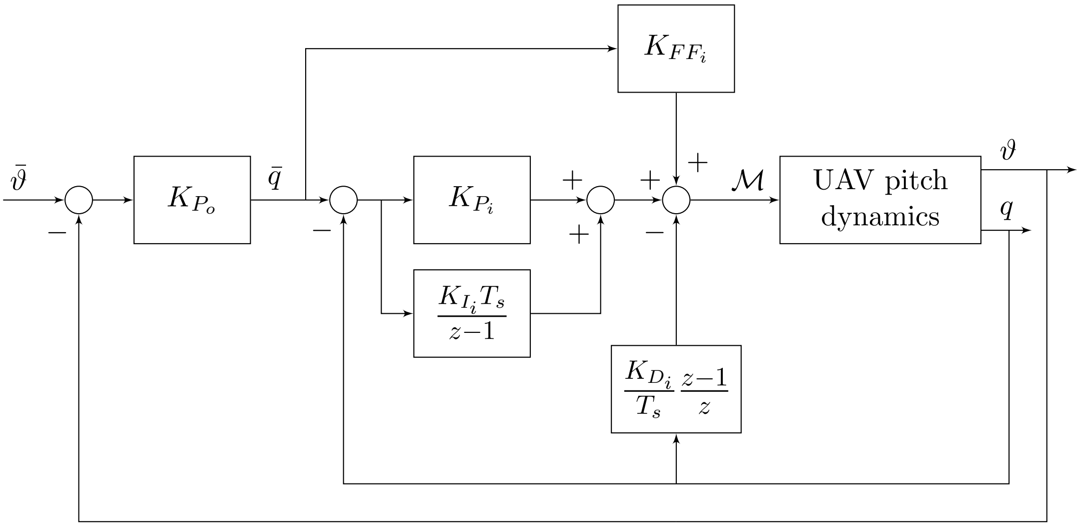


Figure 3.3: The pitch attitude controller with a feed-forward gain and a derivative action based on plant measurements.

outer loop on the pitch attitude feedback is a classical proportional controller, while the inner controller is a complete PID with an additional *feed-forward* term. More specifically, the *feed-forward* gain is directly computed on the pitch angle setpoint and the derivative action of the inner loop is computed starting from the plant output $q(t)$ and not from the pitch angular rate error (see Figure 3.3). This last expedient avoids to generate an impulse on the control action $\mathcal{M}(t)$ when a step is applied to the reference signal $\bar{q}(t)$ of the inner loop.

The output of the outer controller can be computed as:

$$\bar{q}(t) = K_{P_o}\left(\bar{\vartheta}(t) - \vartheta(t)\right)$$

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

$$\mathcal{M}(t) = K_{FF_i}\bar{q}(t) + \left(K_{P_i} + K_{I_i}\frac{T_s}{z-1}\right)(\bar{q}(t) - q(t)) - K_{D_i}\frac{z-1}{zT_s}q(t),$$

where T_s is the sampling time and $\mathcal{M}(t)$ is the pitching moment. The unknown controller parameters that will be tuned with the data-driven approach are:

$$\theta = \begin{bmatrix} K_{P_o} & K_{FF_i} & K_{P_i} & K_{D_i} & K_{I_i} \end{bmatrix}^T. \tag{3.1}$$

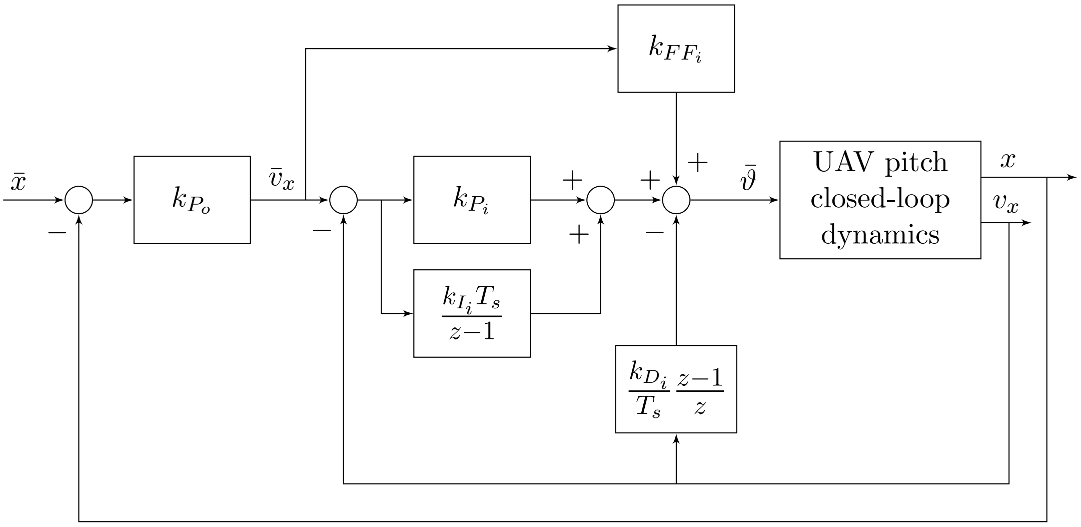


Figure 3.4: The longitudinal position controller with a feed-forward gain and a derivative action based on plant measurements.

3.3 MISO longitudinal position controller

As will be explained in Chapter 5, where the considered UAV is introduced, the very small size of the quadrotor as well as the high flexibility and customisation of its FCU allow to perform in-flight identification experiments. This feature opens the data-driven approach to the design of the position controller. 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, which is indeed coupled with the pitch attitude regulator, but independent from the controllers of other DoFs, at least if small deviations from hover are considered.

The control architecture is analogous to that of the attitude controller shown in Figure 3.3. In this case, however, the proportional controller on the outer loop is based on the longitudinal position, receiving the position error as input and returning the longitudinal linear velocity reference as output. Instead, the inner loop is associated with the longitudinal velocity and presents a derivative term computed on the plant output $v_x(t)$, while a proportional and an integral term are calculated on the velocity error (see Figure 3.4). The inner controller computes the pitch angle setpoint that feeds the attitude controller schematically shown in Figure 3.3.

Since the control architecture is the same as for the attitude regulator described in Section 3.2, the number of the unknown controller parameters are the same and they are:

$$\theta = \begin{bmatrix} k_{P_o} & k_{FF_i} & k_{P_i} & k_{D_i} & k_{I_i} \end{bmatrix}^T .$$

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.

3.4 SISO pitch attitude controller

As already mentioned, the CbT method as well as the controller unfalsification algorithm cannot be used to tune the MISO controllers described in the previous sections. For this reason, a simplified controller structure has been implemented on the quadrotor FCU, ultimately allowing all the data-driven algorithms described in Chapter 2 to be compared (see Section 6.1). For simplicity, the modified control structure and the related data-driven tunings have been implemented and assessed only for the attitude controller.

The new control architecture is based on cascaded SISO PID loops and it is schematically shown in Figure 3.5. Just as in the previous case, the outer regulator $C_o(z)$ is defined as a simple proportional gain based on attitude feedback (estimated angle $\vartheta(t)$ and setpoint $\bar{\vartheta}(t)$). On the contrary, the inner regulator $C_i(z)$ presents a different architecture with respect to that defined in Section 3.2. Specifically, it is a PID controller with SISO structure, thus having only the angular rate feedback (measured angular velocity $q(t)$ and setpoint $\bar{q}(t)$) as input. The output of the outer controller is defined exactly as in the MISO case, while the control input provided by the inner controller can be computed as:

$$\mathcal{M}(t) = \left(K_{P_i} + K_{I_i} \frac{T_s}{z-1} + K_{D_i} \frac{z-1}{zT_s} \right) (\bar{q}(t) - q(t)) .$$

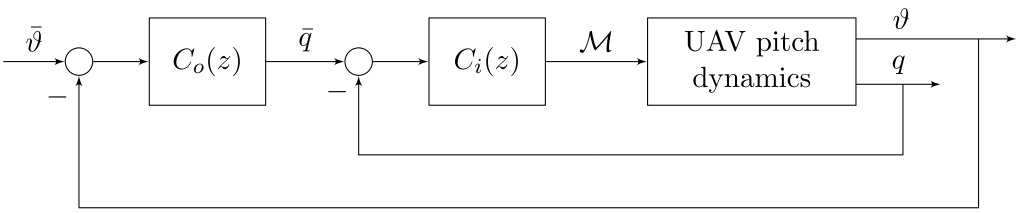


Figure 3.5: The pitch attitude controller based on SISO PID architecture.

Therefore, the transfer functions of the outer and the inner controllers are defined as:

$$\begin{aligned} C_o(z) &= K_{P_o} \\ C_i(z) &= K_{P_i} + K_{I_i} \frac{T_s}{z-1} + K_{D_i} \frac{z-1}{zT_s} \end{aligned}$$

Eventually, the unknown controller parameters that will be tuned with the data-driven approaches are:

$$\theta = \begin{bmatrix} K_{P_o} & K_{P_i} & K_{D_i} & K_{I_i} \end{bmatrix}^T .$$

Chapter 4

Data-driven algorithms for multirotor systems

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

This chapter is organised as follows. In Section 4.1 the VRFT method is adapted to deal with experimental data gathered by means of closed-loop flight tests. Subsequently, Section 4.2 provides an extension to the standard VRFT method in order to tune a controller with a feed-forward term and a derivative action related to the measured output (for further details on control structure see Chapter 3). 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.

4.1 Closed-loop experiment

The standard VRFT algorithm presented in Section 2.2 exploits experimental data that come from an open-loop test. Performing such a test is not always possible, for obvious reasons. Some of the potential applications are indeed unstable by nature, and thus have to operate in closed-loop conditions. Sometimes, also stable systems must operate in closed-loop during the experiment to satisfy some conditions related to the experiment itself, or due to safety reasons. In this work the VRFT method is employed to tune the controllers of a small quadrotor. 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 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 to get this controller goes beyond the scope of this work, but, since this preliminary controller is only used to collect closed-circuit data, the reader should focus on the fastest way to obtain it, as no closed-loop performance requirement is demanded except for system stability.

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 2.2, cannot be applied to obtain a new controller exploiting the measurements $D_N = \{u(t), \tilde{y}(t)\}_{t=1..N}$: 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 setpoints $\bar{y}(t)$ and $\bar{u}(t)$, and the input of the plant is now affected by this action:

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

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:

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

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

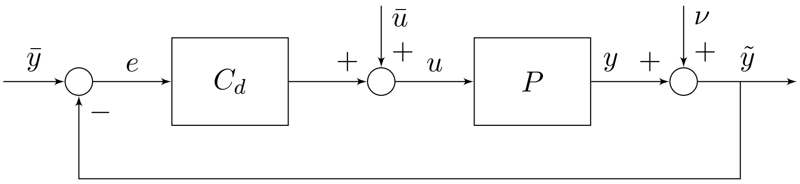


Figure 4.1: VRFT experiment in closed-loop operation.

the input of the plant $u(t)$ will be the same in the two experiments, violating the assumption made in Section 2.2. Recalling that the VRFT method is employed in this work to tune the controller of an UAV and it has been here extended to perform the tuning experiment in flight, it is obvious that a lot of environment uncertainties arises 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 built using the second choice presented in Section 2.2: 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

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

and using (4.2):

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

Following Algorithm 1, the instrumental variable is:

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

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

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

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:

$$\begin{aligned}\zeta(t) &= \beta(z)L(z)\left(M(z)^{-1}-1\right)\hat{y}^{OL}_{\bar{u}}(t)\\ &= \beta(z)L(z)\left(M(z)^{-1}-1\right)\hat{P}(z)\bar{u}(t).\end{aligned} \tag{4.4}$$

The second option for the instrumental variable is:

$$\zeta(t) = \beta(z)L(z)\left(M(z)^{-1}-1\right)\hat{y}^{CL}_{\bar{u}}(t) \tag{4.5}$$

where

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

is the simulated output of the plant when it is fed only by the excitation signal $\bar{u}(t)$ in closed-loop conditions. The choices expressed in (4.4) and (4.5) 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 (4.4) 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 conditions 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 will be presented in the next subsection (see [34] for an overview of closed-loop SMI methods).

4.1.1 The PBSID algorithm

As illustrated in [35], 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 the 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 the state-space models of MIMO systems in a number of industrial applications.

The PBSID algorithm, which is briefly described in the following, considers the finite dimensional, LTI state space model class

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k) \\ \tilde{y}(k) &= Cx(k) + Du(k) + \nu(k) \end{aligned} \tag{4.6}$$

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

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

where K is the Kalman gain associated with (4.6), and note that system (4.6) can be written as

$$\begin{aligned} x(k+1) &= \bar{A}x(k) + \widetilde{B}z(k) \\ \tilde{y}(k) &= Cx(k) + Du(k) + e(k), \end{aligned} \tag{4.7}$$

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 (4.7), where p is the so-called past window length, one gets

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

where

$$\mathcal{K}^p = \begin{bmatrix} \bar{A}^{p-1}\widetilde{B}_0 & \dots & \widetilde{B} \end{bmatrix}$$

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 behaviour of the system is approximately given by

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

so that, introducing the matrix notation defined in the previous subsection, the data equations are given by

$$\begin{aligned}
X^{p,f} &\simeq \mathcal{K}^p \bar{Z}^{p,f} \\
Y^{p,f} &\simeq C\mathcal{K}^p \bar{Z}^{p,f} + DU^{p,f} + E^{p,f}.
\end{aligned}$$

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

$$\min_{C\mathcal{K}^p, D} \|Y^{p,p} - C\mathcal{K}^p \bar{Z}^{p,p} - DU^{p,p}\|_F. \tag{4.8}$$

Defining now the extended observability matrix Γ^p as

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

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

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

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 (4.8). Recalling now that

$$X^{p,p} \simeq \mathcal{K}^p \bar{Z}^{p,p}$$

it also holds that

$$\Gamma^p X^{p,p} \simeq \Gamma^p \mathcal{K}^p \bar{Z}^{p,p}.$$

Therefore, computing the Singular-Value Decomposition (SVD)

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

an estimate of the state sequence can be obtained as

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

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

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

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

$$E_N^{p,f} = Y^{p,p} - \widehat{C} \widehat{X}^{p,p} - \widehat{D} U^{p,p}$$

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

$$\min_{A,B,K} \|\widehat{X}^{p+1,p} - A \widehat{X}^{p,p-1} - B U^{p,p-1} - K E^{p,p-1}\|_F.$$

In [36] 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.

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

```
1: Compute $U(z)$ such that $\left|U\left(e^{j\omega}\right)\right|^2 = \Phi_u(\omega)$.
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 $\varphi(t) = \beta(z) L(z) \left(M^{-1}(z) - 1\right) \tilde{y}(t)$.
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 $\zeta(t) = \beta(z) L(z) \left(M(z)^{-1} - 1\right) \hat{y}_{\bar{u}}(t)$.
8: Compute $\hat{\theta}_N^{IV} = \left[\sum_{t=1}^N \zeta(t) \varphi_L^T(t)\right]^{-1} \sum_{t=1}^N \zeta(t) u_L(t)$.
```

4.2 Controller with feed-forward and action on plant output

As the reader will have already understood, the design of a new controller architecture goes beyond the scope of this work, whose main goal is to use data-driven methods to tune or re-tune a controller that is already implemented. Nevertheless, the pre-existing control architecture of the considered UAV cannot be tuned with the standard data-driven methods illustrated in Chapter 2, so two approaches have been followed: the VRFT algorithm has been extended to deal with a multiple-input control structure and, on the other hand, the control structure has been modified in order to assess all the data-driven tunings and compare their results. In particular, the new controller is a classical SISO PID controller (see Section 3.4) that can be represented as in Scheme 2.1 and it can be tuned with the classical VRFT algorithm, as presented in Section 2.2 as well as with the CbT and the controller unfalsification methods. As regards the pre-existing control architecture, already described in Chapter 3, it consists of a PID whose derivative action is computed on the plant output, combined with a feed-forward gain (see Figure 3.3 and Figure 3.4). This controller can be considered as a MISO system with two input, the setpoint $\bar{y}(t)$ and the plant output $\tilde{y}(t)$, and one output, the control action $u(t)$. In this section, the VRFT algorithm is extended in order to deal with this controller architecture, which is more commonly adopted in industrial applications than the SISO version.

A generalisation of the scheme in Figure 3.3 and Figure 3.4 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:

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

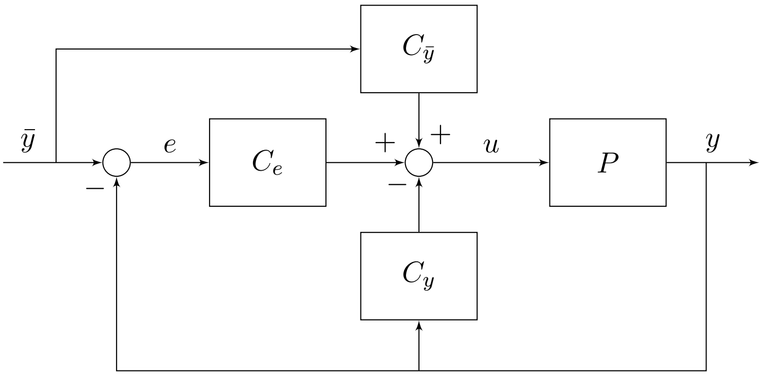


Figure 4.2: Controller with a feed-forward term and an action on plant measurements.

Considering the new controller architecture, also the criterion minimised by the VRFT algorithm in (2.3) changes as:

$$\begin{aligned} J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}}) = \frac{1}{N} \sum_{t=1}^{N} L(z) \big[& u(t) - C_e(z, \theta_e) e(t) \\ & - C_{\bar{y}}(z, \theta_{\bar{y}}) \bar{y}(t) + C_y(z, \theta_y) y(t) \big]^2 . \end{aligned} \tag{4.9}$$

If the controller class is linearly parametrised, the minimum of the criterion (4.9) can be found exploiting the closed form solution as in (2.7). 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

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

the criterion in (4.9) can be rewritten as:

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

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 (2.7).

Considering the new controller architecture, a different selection of the filter $L(z)$ must be made. The filter is shaped starting from the derivation in [37] where the filter is built for a two degree of freedom controller.

Let

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

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

$$M(z) = \frac{\left(\bar{C}_{\bar{y}}(z) + \bar{C}_e(z)\right) P(z)}{1 + \left(\bar{C}_e(z) + \bar{C}_y(z)\right) P(z)} \tag{4.10}$$

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. The filter is defined as

$$|L|^2 = |M|^2 |S|^2 |W|^2 \frac{1}{\Phi_u}, \quad \forall \omega \in [-\pi, \pi]. \tag{4.11}$$

Proof. Note that the criterion stated in (2.1), for this controller architecture is defined as

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

and it can be written as

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

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

$$J_{MR}(\theta_e, \theta_y, \theta_{\bar{y}}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \frac{P\left(C_e(\theta_e) + C_{\bar{y}}(\theta_{\bar{y}})\right)}{1 + P\left(C_e(\theta_e) + C_y(\theta_y)\right)} - M \right|^2 |W|^2 d\omega. \tag{4.12}$$

After some manipulations and exploiting the definition of $M(z)$ in (4.10), the criterion (4.12) can be rewritten as:

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

If the involved signals in (4.9) are realisations of stationary and ergodic stochastic processes and $N \to \infty$, the analysis of $J_{VR}^N(\theta_e, \theta_y, \theta_{\bar{y}})$ is based on asymptotic results:

$$J_{VR}^{N}(\theta_e,\theta_y,\theta_{\bar{y}}) \to J_{VR}(\theta_e,\theta_y,\theta_{\bar{y}}). \tag{4.13}$$

Exploiting the definition of $M(z)$ in (4.10) and the Parseval theorem (see [38]), the asymptotic criterion (4.13) is written as:

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

In the following, $J_{VR}(\theta_e,\theta_y,\theta_{\bar{y}})$ is used instead of $J_{VR}^{N}(\theta_e,\theta_y,\theta_{\bar{y}})$. The main idea of the VRFT algorithm is to minimise $J_{VR}(\theta_e,\theta_y,\theta_{\bar{y}})$ rather than $J_{MR}(\theta_e,\theta_y,\theta_{\bar{y}})$, as described in Section 2.2. If the ideal controllers belong to the class of the available controllers, that is

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

regardless of how the plant, the filters and the reference model are selected, the VRFT method leads to the ideal controller (see [37]). 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. As a matter of fact, if

$$|L|^2 = \frac{|M|^2|W|^2}{\left|1+P\left(C_e(\theta_e)+C_y(\theta_y)\right)\right|^2}\frac{1}{\Phi_u}, \quad \forall\omega\in[-\pi,\pi], \tag{4.14}$$

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 (4.14) depends on $P(z)$, that is unknown, the choice in (4.14) is not feasible. As described in [37], where an exhaustive analysis is proposed, the following assumption can be made:

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

and the filter in (4.14) can be rewritten as in (4.11) 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 variable method must be implemented as in the standard VRFT algorithm presented in Section 2.2.

Algorithm 6 VRFT algorithm with the new controller architectures.

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

4.3 Correlation-based Tuning for cascade control systems

Although the VRFT algorithm was already extended in [14] to deal with cascade control systems with a single set of experimental data, the same extension was not available in the literature for the CbT method. This makes the latter unsuitable for tuning 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 2.2.1.

Also in this case two loops are considered without loss of generality (Figure 2.3). 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 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 2.3, setting $\tilde{y}(t) = \tilde{y}_i(t)$ and $\bar{y}(t) = u(t)$.

Considering the outer regulator, the same problem of the VRFT approach arises, *i.e.* the input of the controlled system is not available in the data. This signal must be built starting from the available information. The procedure to obtain this signal follows the same steps presented in Section 2.2.1 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 with the CbT algorithm as presented in Section 2.3, 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.

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

```
Compute $U_i(z)$ such that $\left|U_i\left(e^{j\omega}\right)\right|^2 = \Phi_u(\omega)$.
**repeat**
    Compute $F_i(z) = \left(1 - M_i(z)\right) W_i(z) U_i^{-1}(z)$.
    Compute $\bar{u}_F(t)$ as $\bar{u}_F(t) = F_i(z)u(t)$.
    Choose $l_i$ close to the impulse response of $M_i(z)$.
    Compute $\varsigma_i(t) = \begin{bmatrix} u_F(t+l) & \dots & u_F(t) & \dots & u_F(t-l) \end{bmatrix}^T$.
    Compute the error $\varepsilon_i(t, \theta_i) = M_i(z)u(t) - C_i(z, \theta_i)(1 - M_i(z))\tilde{y}_i(t)$.
    Compute $f_i(\theta_i) = \frac{1}{N}\sum_{t=1}^{N} \varsigma_i(t)\varepsilon_i(t, \theta)$.
    Compute $J_i(\theta_i) = f_i^T(\theta) f_i(\theta_i)$.
    Compute $\hat{\theta}_i = \arg\min_{\theta_i} J_i(\theta_i)$.
**until** $C_i(z, \hat{\theta}_i)$ is a minimum phase system, otherwise change $M_i(z)$.
Compute $\bar{y}_i(t) = C_i^{-1}(z, \hat{\theta}_i)u(t) + \tilde{y}_i(t)$.
Compute $U_o(z)$ such that $\left|U_o\left(e^{j\omega}\right)\right|^2 = \Phi_{\bar{y}_i}(\omega)$ where $\Phi_{\bar{y}_i}(\omega)$ is the spectral
    density of $\bar{y}_i(t)$.
Compute $F_o(z) = \left(1 - M_o(z)\right) W_o(z) U_o^{-1}(z)$.
Compute $\bar{y}_{iF}(t)$ as $\bar{y}_{iF}(t) = F_o(z)\bar{y}_i(t)$.
Choose $l_o$ close to the impulse response of $M_o(z)$.
Compute $\varsigma_o(t) = \begin{bmatrix} \bar{y}_{iF}(t+l) & \dots & \bar{y}_{iF}(t) & \dots & \bar{y}_{iF}(t-l) \end{bmatrix}^T$.
Compute the error $\varepsilon_o(t, \theta_o) = M_o(z)\bar{y}_i(t) - C_o(z, \theta_o)(1 - M_o(z))\tilde{y}_o(t)$.
Compute $f_o(\theta_o) = \frac{1}{N}\sum_{t=1}^{N} \varsigma_o(t)\varepsilon_o(t, \theta)$.
Compute $J_o(\theta_o) = f_o^T(\theta) f_o(\theta_o)$.
Compute $\hat{\theta}_o = \arg\min_{\theta_o} J_o(\theta_o)$.
```

Chapter 5

The ANT-1 MAV platform

The data-driven methods presented in the previous chapters have been used to design attitude and position controllers of a pre-existing multirotor system.

The tested multirotor is a quadrotor belonging to the Micro Aerial Vehicle (MAV) class, which includes all the Remotely Piloted Aerial Vehicles (RPAV) characterised by limited size and weight. The definition of MAV is quite arbitrary, but referring to the Italian Civil Aviation Authority (ENAC) regulation [39], the RPAVs are rigorously divided into different categories according to the Maximum Take-off Weight (MTOW). More specifically, all the aerial platform weighting less than 300 g (250 g according to the Federal Aviation Autorization (FAA), see [40]) are considered not dangerous for persons or things and therefore exempted from most of the regulations, including: the need for an RPAV flight license and restrictions on overflying of persons. For simplicity, from here on, the flying vehicles belonging to this category will be called MAV.

Over the last few years, these drones are becoming more and more widespread thanks to permissive rules, as well as the great flexibility and thus the wide variety of their uses. In particular, the development of MAVs 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 the insides of 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 the computing, sensing and communication workloads so that they become faster in completing a given task 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 applications, such as monitoring of toxic clouds and meteorological conditions, security and artistic shows.

The small size of the considered UAV simplifies the execution of indoor flight tests, easing the gathering of the in-flight data necessary to design the attitude control system with closed-loop experiments (see Section 4.1), as well as to tune the position controller. All the results related to attitude and position dynamics will be presented in Chapter 6. Hereafter, in Section 5.1, the overall design of the considered quadrotor platform, namely the ANT-1, will be presented and, in Section 5.2, its main components will be briefly described. Finally, an overview of the performed tests and a brief description of the test-bed will be discussed in Section 5.3.

5.1 ANT-1 design overview

The ANT-1 (Figure 5.1) is a small fixed pitch quadrotor developed as a part of a previous master thesis work in [41]. It has been assembled with off-the-shelf components, chosen with the help of a widely tested numerical simulation tool, namely eCalc. This tool allows for an accurate estimation of the main quadrotor performance characteristics by inserting component parameters available on manufacturer data sheets. The goal was to design a MAV that would comply with the following requirements:

- Maximum Take-Off Weight (MTOW): not exceeding 300 grams (to satisfy ENAC standards for RPAVs [39]).
- Flight time: at least 10 minutes.
- Frame dimensions (footprint): within 200 mm (including rotors).

After simulating several different configurations, the choice of the components has been the result of a compromise between hovering efficiency and overall dimensions. The chosen components are listed in Table 5.1, while Table 5.2 summarizes the main results provided by eCalc.

As can be seen, according to the online tool, all requirements have been met. Better performance in terms of specific thrust, and thus flight time, could have been obtained by choosing larger propellers. Indeed, the specific thrust provides an idea of the hovering efficiency of a rotorcraft, being defined as the ratio between propeller thrust and electric motor absorbed power. Nevertheless, rotors with a larger diameter would have required a wider frame, eventually resulting in a significant increase in the overall weight. For this reason, the propellers selected in [41] have been considered the best compromise with respect to the initial design requirements.

Component type	**Component name**	**Weight** [g]
FCU	Pixfalcon	9
Companion Computer	Raspberry Pi Zero W	15
Battery	Turnigy Nano-tech 950 mAh Li-Po Pac	69
Motors	QAV1306-3100KV brushless motor	12
ESC	ZTW Spider Series 18A	5.6
Propeller	Gemfan Bullnose 3035, 3 blades	4
Frame	Self-made (150 mm excluding rotors)	80

Table 5.1: List of components.

Weight	**233** g
Hovering flight time	**11.9** min
Thrust-to-weight ratio	**2.4**
Specific thrust	**5.35** g/W
Efficiency (in hovering)	**85.3** %
Estimated rate of climb	**16.7** m/s
Maximum speed	**92** km/h

Table 5.2: eCalc results.

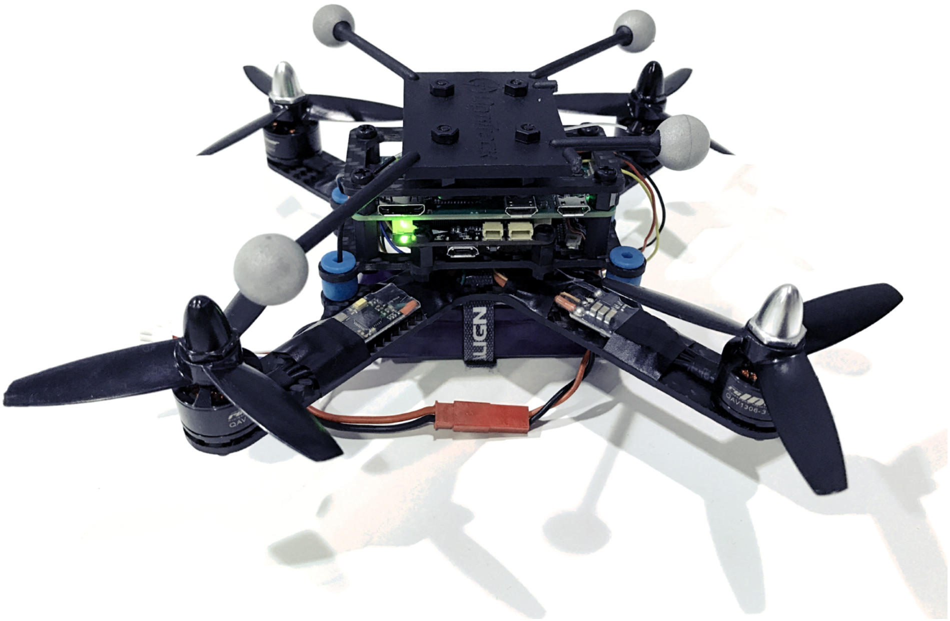


Figure 5.1: The ANT-1 quadrotor.

5.2 Hardware description

In the following section the main components will be described in detail. For further information on hardware choice and integration see [41].

5.2.1 Flight control unit and companion computer

Flight control unit

The Flight Control Unit (FCU) function is performed by the Pixfalcon board (Figure 5.2), which is a small and light autopilot kit commercially available off-the-shelf and suitable for a wide variety of remote-controlled systems including, but not limited to, airplanes, multirotors and helicopters. The sensing unit includes:

- 3 orthogonally placed rate-gyroscopes;
- 3-axis linear accelerometer;
- 3-axis magnetometer;
- Barometric pressure sensor;
- GPS receiver (optional).

Information collected by the Inertial Measurement Unit (IMU) sensors provides an accurate estimation of the state variables, while real-time processing through the controller allow for the computation of the appropriate control action. More specifically, the firmware chosen and installed on the ANT-1 is the PX4 Pro Autopilot, which is an open-source software downloadable from GitHub. The Pixfalcon is characterized by 8 PWM output channels that must be connected to the Electronic Speed Controls (ESCs), to allow for adjusting motors rotational speed and, ultimately, controlling vehicle attitude.

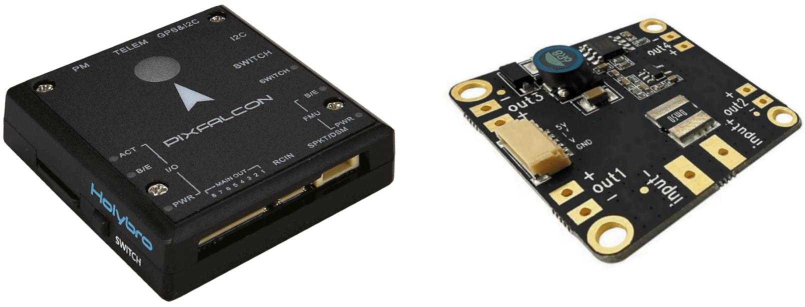


Figure 5.2: Pixfalcon.

Companion computer

Pixfalcon is connected to a Raspberry Pi Zero W (Figure 5.3), which is a small single-board computer equipped with a 1 GHz single-core ARM processor and 512 MB of SDRAM. This companion computer has the main function of communicating with the laboratory LAN through a Wi-Fi connection, enabling the drone to be controlled by a ground station that provides input commands to the quadrotor via a MATLAB interface. Moreover, due to its computational power, it is going to have a primary role in collaborative flight when ANT-1 will be involved in MAV swarming.

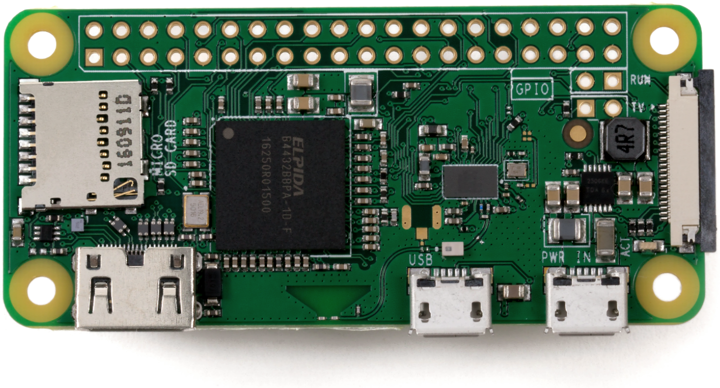
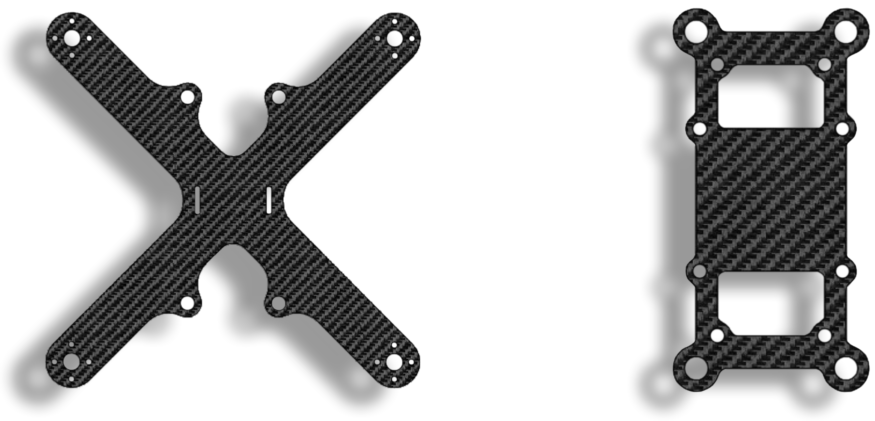


Figure 5.3: Raspberry Pi Zero W.

5.2.2 Frame

Although several off-the-shelf frames were readily available, a self-designed one has been preferred. Indeed, a customized frame ensures compatibility with the chosen components, such as battery, motors and logic cards. In this way, the frame does not represent a constraint in terms of components selection except for geometric dimensions, since the frame itself is limited in size by design requirements. Indeed, to meet the size and weight requirements a carbon fiber cross plate of 160 mm has been manufactured (Figure 5.4a). The choice of a composite material provides both lightness (only 30 gr) and high stiffness to the vehicle structure. Finally, a specific support plate has been built and assembled onto the upper part of the frame to accommodate the FCU and the companion computer (Figure 5.4b). These two elements are joined together through damping balls which are intended to mitigate the effects of rotors vibrations on the IMU sensors.



(a) Cross plate.

(b) FCU and Raspberry support.

Figure 5.4: ANT-1 frame.

5.2.3 Battery, motors and propellers

Battery

A 950 mAh Li-Po battery (Figure 5.5) is located underneath the quadrotor and powers all the on-board devices through a suitable power module. Battery capacity suggested by eCalc has been the result of an inherent compromise between weight and flight time. According to eCalc the maximum current required by the four motors operating at maximum power is equal to 16.8 A, which is compatible with the continuous discharge rate ensured by the chosen battery (950 mAh capacity with 25C continuous discharge rate means the battery can supply up to 23.75 A continuously).

TURNIGY
nano-tech
950mAh 25-50C DISCHARGE
0.95
HIGH DISCHARGE Li-PO BATTERY
VOLTAGE 1 Cell 3.7V 2 Cell 7.4 3 Cell 11.1V

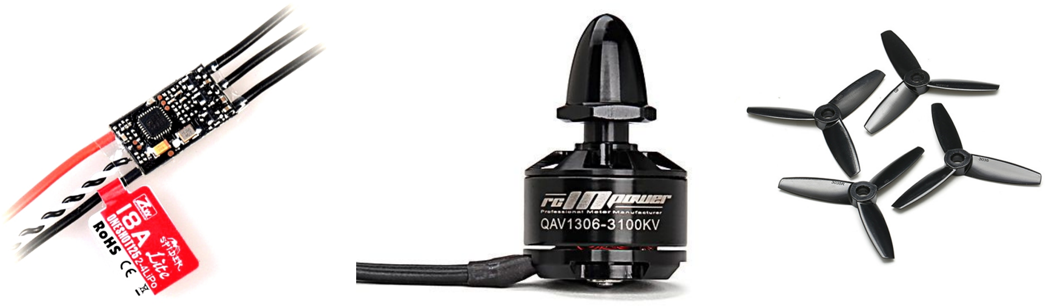
Figure 5.5: 950 mAh Li-Po battery.

Motors and ESCs

The battery powers the four motors via their respective ESCs (Figure 5.6a). These devices have the basic task of converting DC voltage provided by the battery into a three-phase alternating current suitable for synchronous motors. This supply voltage depends on the Pulse Width Modulation (PWM) servo signal generated by the FCU, which is proportionally related to the desired rotational speed of the motors. As mentioned, ANT-1 is powered by four synchronous Brushless DC motors (Figure 5.6b) that ensure high power-to-weight ratio. The selected motors are characterized by a KV parameter equal to 3100 rpm/V. This parameter represents the number of revolutions per minute for each Volt applied to the motor (*e.g.*, if the battery guarantees 11.1 V, $KV = 3100$ rpm/V means a speed of 34410 rpm is ideally reached with no external loads applied).

Propellers

Finally, each motor drives a three-blade propeller to generate the aerodynamic thrust (Figure 5.6c). The chosen propellers are characterized by a diameter of 76 mm and a (fixed) pitch of 88 mm. As already mentioned, the small diameter of the propellers represents a mandatory choice due to the reduced overall size of the quadrotor. The required thrust being fixed, smaller propeller generates lower air flow rate and consequently higher variation in air velocity is needed. This results in higher rotational speed and therefore higher blade pitch. As a negative aspect, this solution leads to reduced hovering efficiency due to a higher disc loading (ratio between gross weight and thrust area), but ensures faster motor dynamics in terms of rpm variations and, consequently, enhanced responsiveness of the closed-loop attitude dynamics of the whole quadrotor.



(a) ESC. **(b)** Brushless DC motor. **(c)** Propellers.

Figure 5.6: ANT-1 propulsion system.

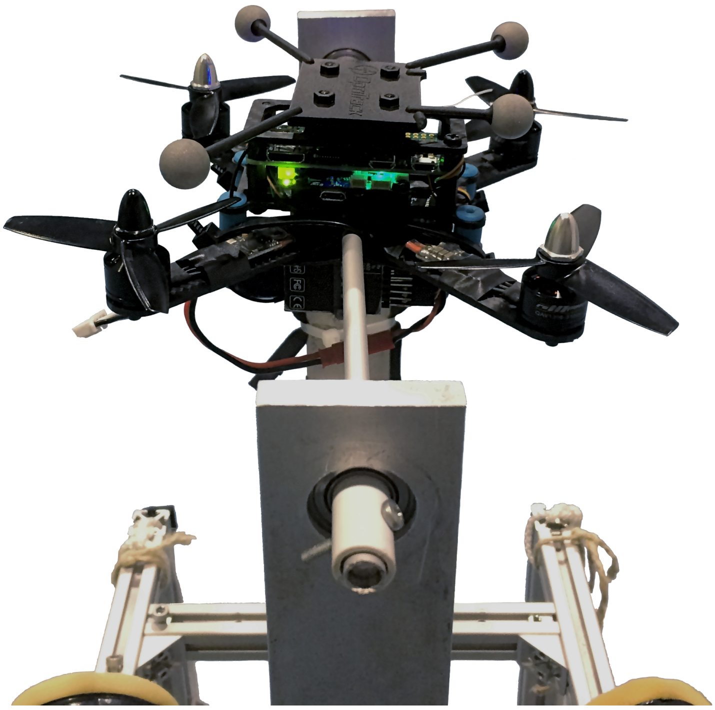


Figure 5.7: The ANT-1 quadrotor on the test-bed.

5.3 Test-bed and flight tests overview

The highly customised design and the wise selection of the FCU ensure high flexibility in terms of flight control law implementations, allowing the experimental testing of all the data-driven algorithms proposed in this work. Some experiments required the use of a test-bed (see Figure 5.7), such as the open-loop tuning experiment and the attitude validation experiments. The test-bed is built out of x-frame aluminium rods and weighted with sacks of concrete. The upper part of the frame carries a beam with circular cross section, resting on ball bearings at both ends for frictionless rotation. The quadrotor is then securely fastened to this shaft. In the current mounting scheme, the shaft axis passes as close as possible to the centre of mass of the system in order to interfere as little as possible with the rotational dynamics of the quadrotor. Nevertheless, because of the physical configuration of the system, a small distance between the shaft and the actual centre of mass is still present. In turn, this causes the system to act like a pendulum, adding a small damping effect that is negligible for small oscillations. Furthermore, the test-bed holds the quadrotor high enough to prevent any ground effect disturbance. However, small interferences still occur as a consequence of the interaction between rotor wakes and the test-bed frame. This represents an additional discrepancy with respect to the real in-flight behaviour, where the rotor wakes develop free from obstacles. Despite these non-idealities, it has been shown in [42] that such a test-bed is representative of the actual attitude dynamics in flight.

First, the attitude controller has been tuned with all the data-driven methods presented in the previous chapters; *i.e.* VRFT (with open-loop and closed-loop gathered data), CbT and controller unfalsification. For each one, a validating experiment has been carried out on the test-bed, and all the results are shown in Section 6.1 and 6.2. After that, the position controller has been considered, so flight tests have been performed either for collecting the tuning data and for validating the position dynamic response, as will be widely discussed in Section 6.3.

Chapter 6

Results

In this chapter the simulation and the experimental results obtained applying the data-driven methods to the ANT-1 quadrotor is illustrated. Section 6.1 contains the results obtained by tuning the controllers with open-loop experimental data; in particular, VRFT is compared with the H_∞ and manual controllers and with the results provided by the CbT and controller unfalsification methods. In Section 6.2, the VRFT method applied to closed-loop gathered data is compared with that based on open-loop experiments. Finally, in Section 6.3 the results related to the position controller tuned with the VRFT algorithm are discussed.

6.1 Pitch attitude controller with open-loop experiments

In the following sections, the results of the attitude controller tuning is discussed in detail. In particular, in Subsection 6.1.1 an overview of the tuning experiment is provided. Then, the reference models of inner and outer loops is outlined in Subsection 6.1.2 and all the simulated MATLAB results are shown in Subsection 6.1.3. Finally, the experimental results obtained through suitable tests, carried out on the laboratory test-bed, are discussed in the last Subsection (6.1.4).

6.1.1 Tuning experiment

One of the key element of any data-driven algorithms is the so-called tuning experiment. That is, the collection of a sufficiently long input-output dataset that excites the dominant system dynamics. The tuning experiment has been carried out with the quadrotor mounted on the test-bed since only the pitch dynamics have to be excited. Figure 6.1 shows the input and output signals used for data-driven tuning. The entire dataset has been obtained by combining three subsequent open-loop tests

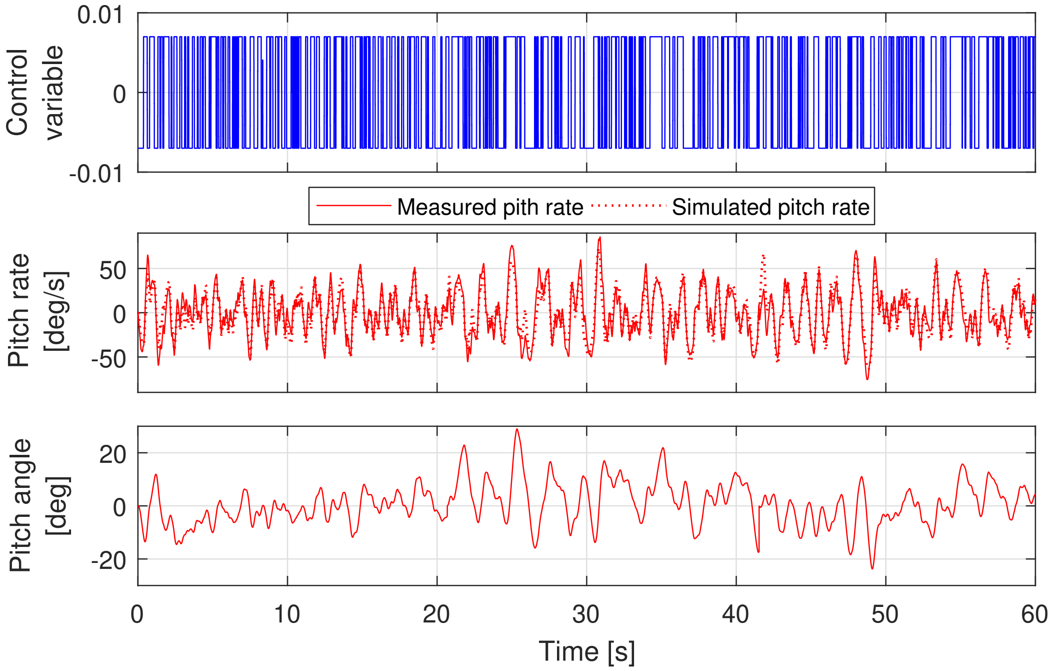


Figure 6.1: Open-loop experimental dataset used by data-driven methods. Dotted pitch rate represents the simulated open-loop response of the quadrotor.

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 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 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 using only on-board sensors (*i.e.* accelerometer and gyroscope).

6.1.2 Reference models

In all data-driven algorithms, a wise choice of the reference model is an essential step in achieving satisfactory closed-loop performance. In principle, the reference model should have the highest possible cut-off frequency to ensure a sufficiently fast dynamic response, but without being physically unachievable. In order to better understand which is the physical behaviour we are asking for, it is advisable to rely on a simple second-order model characterized 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 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 time delay, but sometimes it is also necessary to change the numbers of poles and zeros, as will be shown below.

Furthermore, both the VRFT and the CbT methods provide for user-defined weighting function $W(z)$, which allows forcing the matching problem within a certain frequency range. Typically, one wants the closed-loop behaviour to be as close as possible to the desired one at least at low frequency, so the weighting functions will be of low-pass filter type.

MISO PID controller

As already mentioned, we always start from a second order model characterized by two complex poles. In order to verify the suitability of a simple second order reference model, consider the controller structure defined in Section 3.2 and approximate the quadrotor attitude dynamics with a second order transfer function. The latter is a fairly strong assumption but allows for a significant simplification of the analytical calculations that will follow, without undermining the generality of the discussion. Hence, by assuming a simple non-filtered derivative term,

$$C_{PI}(s) = \frac{K_{P_i}s + K_{I_i}}{s} \quad \text{and} \quad C_D(s) = K_{D_i}s\,,$$

and the simplified pitch rate dynamics of the form

$$P_i(s) = \frac{a}{bs^2 + cs + d}\,,$$

the closed-loop transfer function can be computed as follow:

$$\begin{aligned} F_i(s) &= \frac{C_{PI}(s)P_i(s)}{1 + C_D(s)P_i(s) + C_{PI}(s)P_i(s)} = \\ &= \frac{\dfrac{aK_{P_i}s + aK_{I_i}}{bs^3 + cs^2 + ds}}{1 + \dfrac{aK_{D_i}s}{bs^2 + cs + d} + \dfrac{aK_{P_i}s + aK_{I_i}}{bs^3 + cs^2 + ds}} = \\ &= \frac{aK_{P_i}s + aK_{I_i}}{bs^3 + (c + aK_{D_i})s^2 + (d + aK_{P_i})s + aK_{I_i}}\,. \end{aligned} \tag{6.1}$$

As can be seen from (6.1), the closed-loop transfer function has one real zero and three poles, which roughly results in a -40 dB/decade gradient of the Bode diagram. For this reason, the choice of a second order reference model for the inner loop might well approximate the actual behaviour of the closed-loop system, at least within the desired bandwidth.

The resulting continuous-time inner loop reference model is of the form

$$M_i(s) = \frac{\omega_{ni}^2}{s^2 + 2\xi_i \omega_{ni} s + \omega_{ni}^2} \, ,$$

where ω_{ni} is the desired natural frequency and ξ_i is the damping ratio.

So, the system dynamics will not be changed apart from adding the 5 samples time delay of the plant. As far as the choice of ω_{ni} and ξ_i is concerned, the maximisation of the closed-loop performance has been pursued, taking as goal a pre-existing H_∞ controller. The selected parameters are collected in Table 6.1 and the resulting discrete time transfer function is the following:

$$M_i(z) = z^{-5} \frac{0.0238z + 0.02265}{z^2 - 1.816z + 0.8626} \, . \tag{6.2}$$

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

The reference model of the outer control loop can be defined in a similar fashion. Recall that the outer controller consists of a simple proportional term, $C_o(s) = K_{P_o}$, while the system to be controlled will be given by the multiplication of the inner loop and an integrator, that is

$$P_o(s) = \frac{aK_{P_i}s + aK_{I_i}}{bs^3 + (c + aK_{D_i})s^2 + (d + aK_{P_i})s + aK_{I_i}} \cdot \frac{1}{s} \, .$$

Hence, the closed-loop transfer function is the following.

$$\begin{aligned} F_o(s) &= \frac{C_o(s)P_o(s)}{1 + C_o(s)P_o(s)} = \\ &= \frac{aK_{P_i}K_{P_o}s + aK_{I_i}K_{P_o}}{bs^4 + (c + aK_{D_i})s^3 + (d + aK_{P_i})s^2 + a(K_{I_i} + K_{P_i}K_{P_o})s + aK_{I_i}K_{P_o}} \end{aligned}$$

Unlike the inner loop, an additional root of the denominator has now appeared, which means it is advisable to augment the complexity of the reference model by adding a suitable pole. Therefore, the resulting continuous time reference model is of the form

$$M_o(s) = \frac{\omega_{no}^2}{s^2 + 2\xi_o\omega_{no}s + \omega_{no}^2}\,\frac{p_o}{s+p_o}\,,$$

where ω_{no} and ξ_o respectively are the natural frequency and the damping ratio of the outer reference model and p_o is the position of the additional pole. As before, a time delay of 5 samples has been added to the final discrete time transfer function,

$$M_o(z) = z^{-5}\frac{0.0008861z^2 + 0.003231z + 0.0007374}{z^3 - 2.635z^2 + 2.333z - 0.6927}\,. \tag{6.3}$$

Finally, in this case, 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 lowpass digital ninth-order *Butterworth filter*, with a cut-off frequency equal to 200 rad/s. All the characteristics of the outer reference model are summarized in Table 6.1

	ω [rad/s]	ξ	Added poles/zeros	Weighting function
Inner loop	**56**	**0.33**	-	-
Outer loop	**45**	**0.52**	Pole in **45** rad/s	Low-pass filter (200 rad/s)

Table 6.1: Inner and outer reference models for the VRFT algorithm (MISO PID control structure).

SISO PID controller

If a SISO PID architecture is taken into account, *i.e.* a control structure in which the derivative action is applied to the error (see Section 3.4), the reference models just defined are no longer suitable. Indeed, by considering the ideal PID controller of the form

$$C_{PID}(s) = \frac{K_{D_i}s^2 + K_{P_i}s + K_{I_i}}{s}\,,$$

it is straightforward to show that the inner loop transfer function is the following:

$$F_i(s) = \frac{aK_{D_i}s^2 + aK_{P_i}s + aK_{I_i}}{bs^3 + (c + aK_{D_i})s^2 + (d + aK_{P_i})s + aK_{I_i}}\,. \tag{6.4}$$

It should be noted that the controller transfer function written above is not feasible in practice, but since a merely preliminary analysis is being carried out, this simplification does not affect its validity. As in the previous case, it can be shown that the transfer function of the outer loop turns out to be:

$$F_o(s) =$$

$$= \frac{aK_{D_i}K_{P_o}s^2 + aK_{P_i}K_{P_o}s + aK_{I_i}K_{P_o}}{bs^4 + (c + aK_{D_i})s^3 + (d + a(K_{P_i} + K_{D_i}K_{P_o}))s^2 + a(K_{I_i} + K_{P_i}K_{P_o})s + aK_{I_i}K_{P_o}} .$$

As can be seen from equation (6.4), a second order model will probably not accurately represent the behaviour of the inner closed-loop system. For this reason a zero has been added, resulting in an actual improvement of such designed control systems. Table 6.2 summarizes 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 5 time delay samples.

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 unfalsification method does not provide any weighting function.

		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 6.2: Inner and outer reference models for VRFT, CbT and controller unfalsification algorithms (SISO PID control structure).

Looking to Table 6.2, different outer reference models for different data-driven methods should surprise the reader. It is straightforward 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 2.4, in order to apply the controller unfalsification method, the user must define the reference model also for the desired input sensitivity function. This is mandatory to ensure the internal stability of the closed-loop system. Following the conditions in Section 2.4, the model is defined as:

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

6.1.3 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, *i.e.* the choice of the reference models, is intended to significantly outperform the manual tuning and, in principle, achieve performance comparable with the pre-existing H_∞ controller.

Plant model

The major advantage of data-driven methods (*i.e.*, not to require an accurate knowledge of the plant model) turns out to be, at the same time, a drawback of such approaches. Indeed, the choice of an adequate reference model that guarantees good closed-loop performance is not so trivial. For this reason, the lack of an identified model involves the need for some experimental tests in order to achieve the desired behaviour. By contrast, if a sufficiently detailed mathematical model is available,

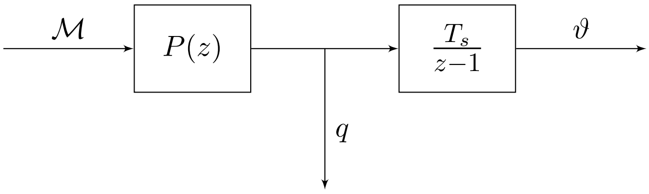


Figure 6.2: Quadrotor attitude dynamics model.

the controller tuning procedure can be considerably speeded up, ultimately achieving better output performance. Nevertheless, the tuning method is still model-free, but faster iterations between controller tunings allows one to more rapidly define an optimal model reference. The resulting data-driven tuning will be eventually validated on the real plant.

An accurate ANT-1 pitch attitude model has been identified in [41] 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 6.2, the attitude dynamics has been modelled by means of the cascade of two dynamic systems:

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

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 6.3.

Since the quadrotor is inherently non-linear, the validity of the identified model is necessarily restricted to the vicinity of the hovering conditions. Furthermore, the identified model completely neglects the electrical dynamics as well as the rotor dynamics. Basically, it is as if the dynamics of the electric motors had been statically approximated, which means that a direct relationship exists between demanded pitching moment and applied rotational speed. This approximation is reasonable as the natural frequencies of the rigid modes are much smaller than those of the electric motors and the rotors.

In order to validate the model, the experimental results can be compared with numerical simulation based on the identified transfer function. In particular, the same input signal used for the open-loop test can be exploited as input signal for the identified model, so that simulated and experimental output data can be compared. For further details about the open-loop test and input and output signals, see Subsection 6.1.1. As can be seen from Figure 6.1, the identified system response (pitch rate) accurately reproduces that of the real one. Therefore, due to the proven reliability of the identified model, it is expected that the tuning achieved via numerical simulations will provide comparable performance on the test-bed (see Subsection 6.1.4).

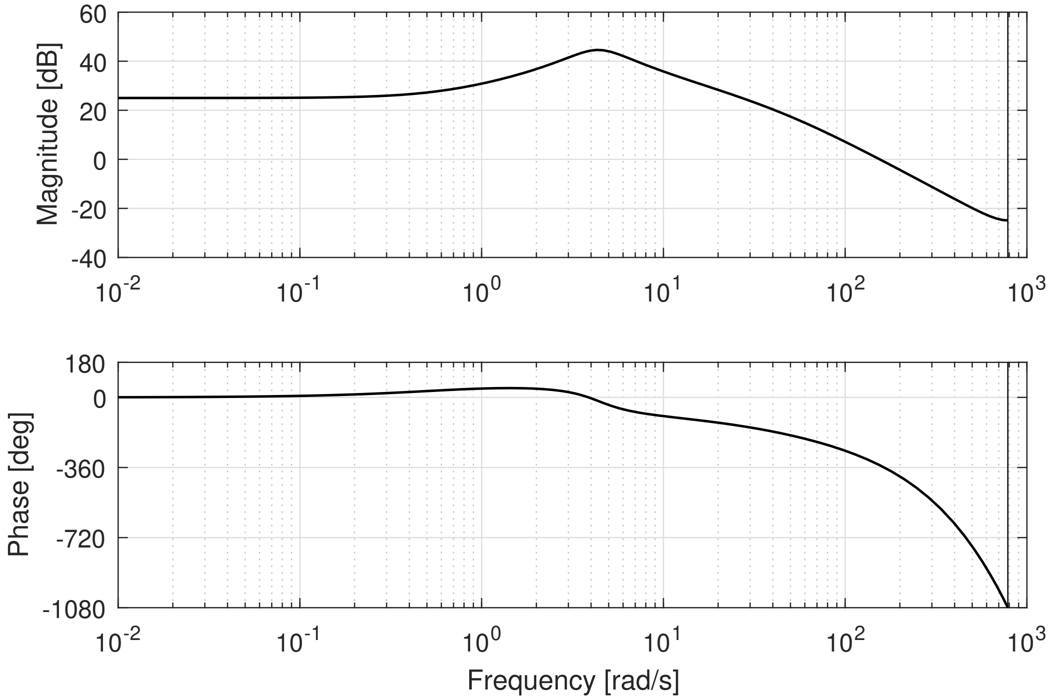


Figure 6.3: Bode diagram of the identified pitch attitude model.

Inner loop results

The main results achieved by closing the inner control loop will be shown below. More specifically, both frequency domain and time domain results will be discussed. The inner control loop is characterized by a PID controller that has been implemented with two different control architectures in order to validate the VRFT as well as the CbT and the controller unfalsification methods. Furthermore, besides being compared with each other, the data-driven tuning results will be compared with the manual tuning and the H_∞ controller. All the results that will follow have been obtained with the reference models widely discussed in Subsection 6.1.2. Before presenting the results of the inner loop, it is convenient to recall the controller transfer functions mentioned in Chapter 3.

Controller transfer functions. As already said, two different PID control structures have been implemented for the inner control loop. The first applies the derivative action on the measurement, as typically happens in industrial controllers, and also includes a feedforward term (see Figure 3.3). On the other hand, a second control structure has been implemented to allow the controller design via the CbT and the controller unfalsification algorithms (see Figure 3.5). Indeed, unlike the VRFT that has been suitably modified to adapt to the new control structure, these tuning methods necessarily require a simple SISO PID controller, or more generally a controller defined with a single transfer function.

In the former case, in fact, the following three different controller transfer function vectors are defined:

$$\beta_{\bar{y}}(z) = 1\,, \qquad \beta_e(z) = \begin{bmatrix} 1 & \dfrac{T_s}{z-1} \end{bmatrix}^T, \qquad \beta_y(z) = \begin{bmatrix} \dfrac{1-z}{zT_s} \end{bmatrix}. \tag{6.6}$$

Where T_s is the sampling time, equal to 0.004 s. Hence, from (6.6), the linear controllers respectively depending only on the reference signal, the error and the plant output can be derived.

$$\begin{aligned} C_{\bar{y}}(z) &= \beta_{\bar{y}}^T(z)\,\theta_{\bar{y}} = K_{FF_i} \\ C_e(z) &= \beta_e^T(z)\,\theta_e = K_{P_i} + K_{I_i}\frac{T_s}{z-1} \\ C_y(z) &= \beta_y^T(z)\,\theta_y = K_{D_i}\frac{1-z}{zT_s} \end{aligned}$$

Where the parameter vector is a four elements vector defined as $\theta = [\theta_{\bar{y}} \quad \theta_e^T \quad \theta_y]^T$. Note that the VRFT requirement to rely on linear controller families is still met.

In the latter case, the controller linearly dependent on parameters is of the form

$$C(z,\theta) = \beta^T(z)\,\theta\,, \tag{6.7}$$

where β is the vector of linear discrete time transfer functions defined as

$$\beta(z) = \begin{bmatrix} 1 & \dfrac{T_s}{z-1} & \dfrac{z-1}{zT_s} \end{bmatrix}^T.$$

In this case, since the vector β is unique, both the standard VRFT and the CbT algorithms can be applied. Regarding the controller unfalsification method, the controller transfer function must be reformulated in order to define a different parameter vector.

In particular, by rewriting the PID transfer function (6.7) in an extended form one obtains:

$$\begin{aligned} C(z,\theta) &= K_{P_i} + K_{I_i}\frac{T_s}{z-1} + K_{D_i}\frac{1-z}{zT_s} \\ &= \frac{z^2(K_{P_i}T_s + K_{D_i}) + z(K_{I_i}T_s^2 - 2K_{D_i} - K_{P_i}T_s) + K_{D_i}}{(z-1)T_s z}\,. \end{aligned}$$

Then, by recalling the parametric controller family (2.12) defined in Section 2.4, one obtains

$$\begin{aligned} D^*(z) &= T_s z^2 + T_s z \\ \bar{N}(z,\theta) &= z^2(K_{P_i}T_s + K_{D_i}) + z(K_{I_i}T_s^2 - 2K_{D_i} - K_{P_i}T_s) + K_{D_i} \end{aligned}$$

where $D^*(z)$ is the denominator polynomial with fixed coefficients and characterized by an unstable root, while $\bar{N}(z,\theta)$ is the polynomial whose coefficients form the unknown parameter vector. So, the controller unfalsification parameter vector is defined as

$$\rho = [K_{P_i}T_s + K_{D_i}\,,\quad K_{I_i}T_s^2 - 2K_{D_i} - K_{P_i}T_s\,,\quad K_{D_i}]^T\,.$$

Finally, the PID parameters θ are related to the parameter vector ρ by means of the following relationship:

$$\theta = \begin{bmatrix} T_s & 0 & 1 \\ -T_s & T_s^2 & -2 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \rho\,.$$

VRFT tuning results with MISO PID controller. Figure 6.4 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 6.5). Furthermore, it is clearly visible that the the manual tuning is significantly slower than both data-driven and H_∞ controller tunings.

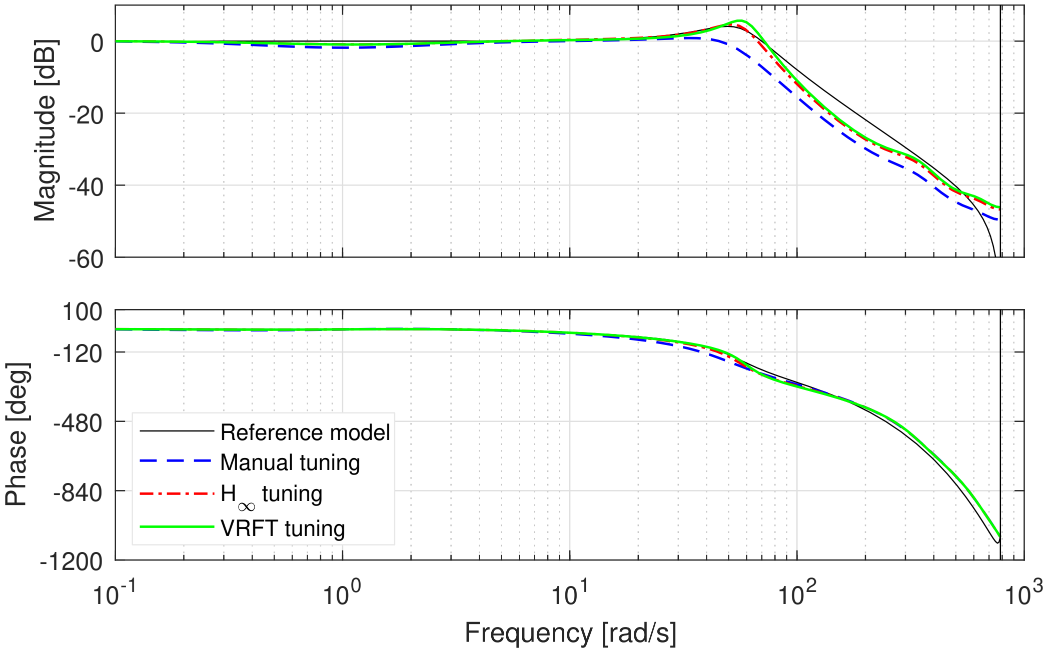


Figure 6.4: Comparison of the inner loop Bode diagrams considering VRFT, H_∞ and manual tunings (simulation).

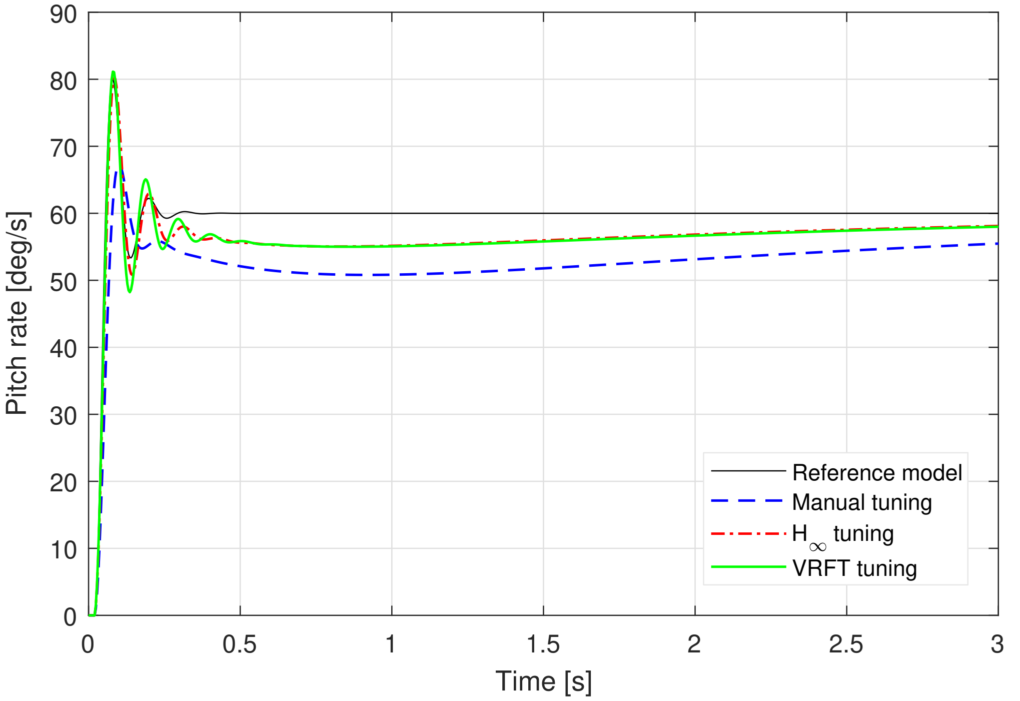


Figure 6.5: Comparison of the inner loop step responses considering VRFT, H_∞ and manual tunings (simulation).

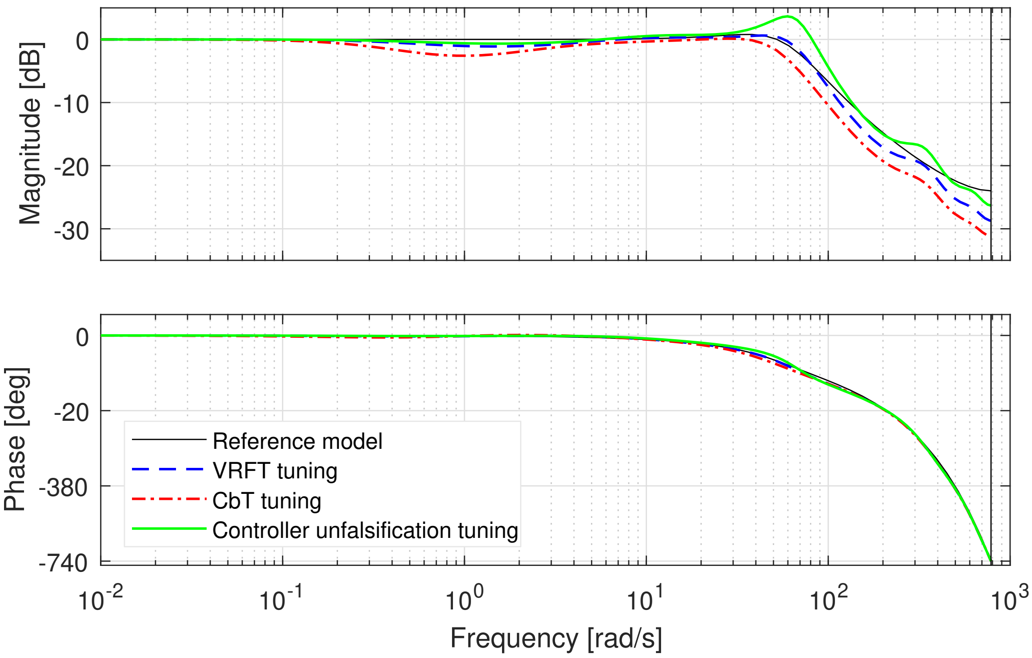


Figure 6.6: Comparison of the inner loop Bode diagrams considering VRFT, CbT and controller unfalsification tunings (simulation).

VRFT, CbT and controller unfalsification tunings results with SISO PID controller. Moving on the simplest control architecture, *i.e.* the SISO PID, the internal controller tuning has been carried out using CbT algorithm and the unfalsification method. For completeness, a VRFT tuning has been also implemented, so that it can be compared with the other two tunings. By analysing Figure 6.6 and Figure 6.7, it is clearly evident that the VRFT tuning ensures better performance than the CbT counterpart, as already shown in [9].

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 a destabilising controller. On the contrary, the CbT method maintains roughly constant performance as data SNR decreases. As a drawback, for sufficiently 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. Indeed, in [9] it has been shown that a good choice for the l parameter is represented by the length of the reference model impulse response (Figure 6.8), and it can be increased the more the SNR value is large. Both for the

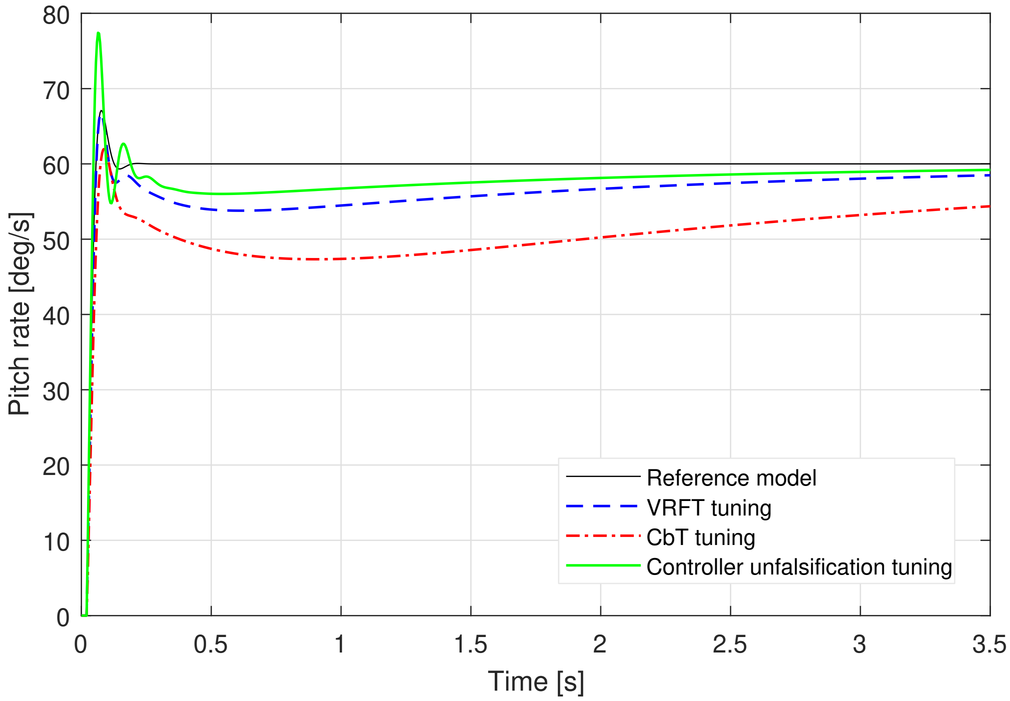


Figure 6.7: Comparison of the inner loop step responses considering VRFT, CbT and controller unfalsification tunings (simulation).

inner and outer loop, l has been set equal to 500.

Regarding the controller unfalsification, the results reported in Figure 6.6 and Figure 6.7 have been obtained by solving the optimisation problem (2.17) considering the reference model reported in Table 6.2. Thanks to the wise choice of the reference model 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 6.2 an 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 input sensitivity with respect to the output one, until the stability test has passed. Figure 6.9 shows the gradual reduction of the infinity norm discrepancy in (2.18) as δ is reduced. According to the small-gain theorem, the stability is ensured if the infinity norm is less than one. As suggested in [16], it is advisable to set a minimum stability margin to take into account the infinity norm estimation error in (2.18), that is why the stability test is passed only when the infinity norm is less than 0.95 (dotted line).

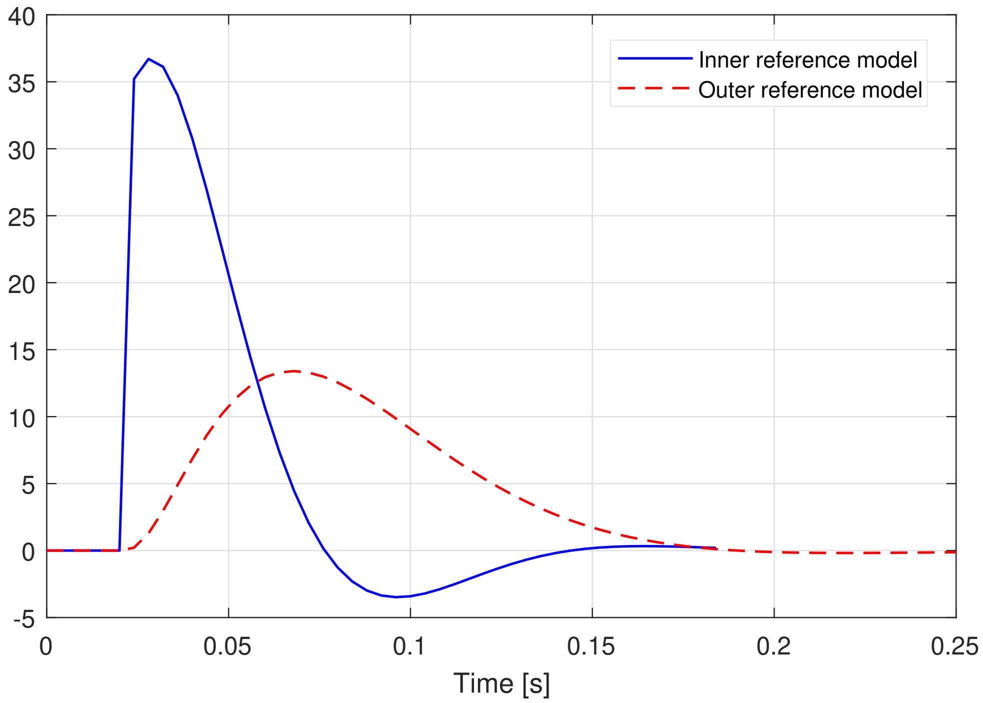


Figure 6.8: Impulse response of the inner and the outer reference models.

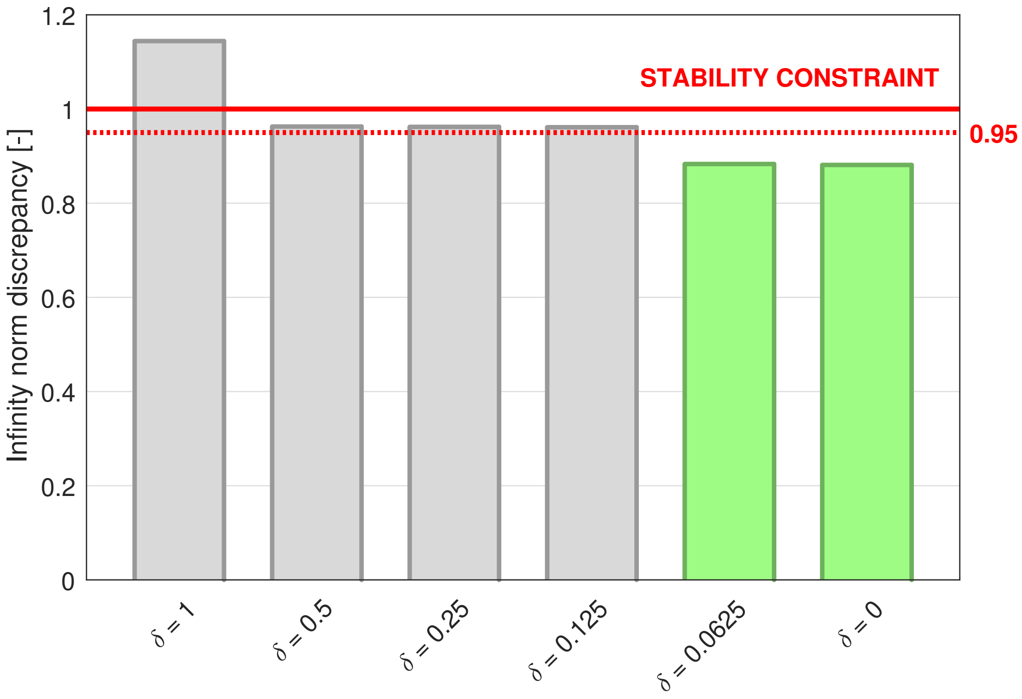


Figure 6.9: Infinity norm discrepancy.

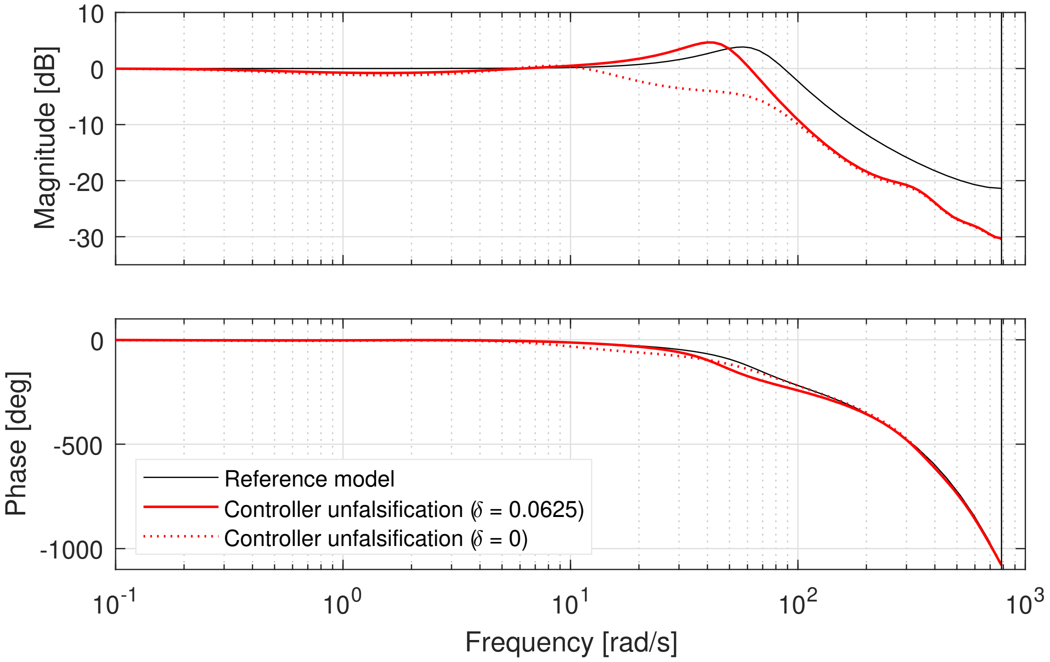


Figure 6.10: Inner loop Bode diagram considering the controller unfalsification tuning with unachievable reference model and the stabiliser controller ($\delta = 0$) (simulation).

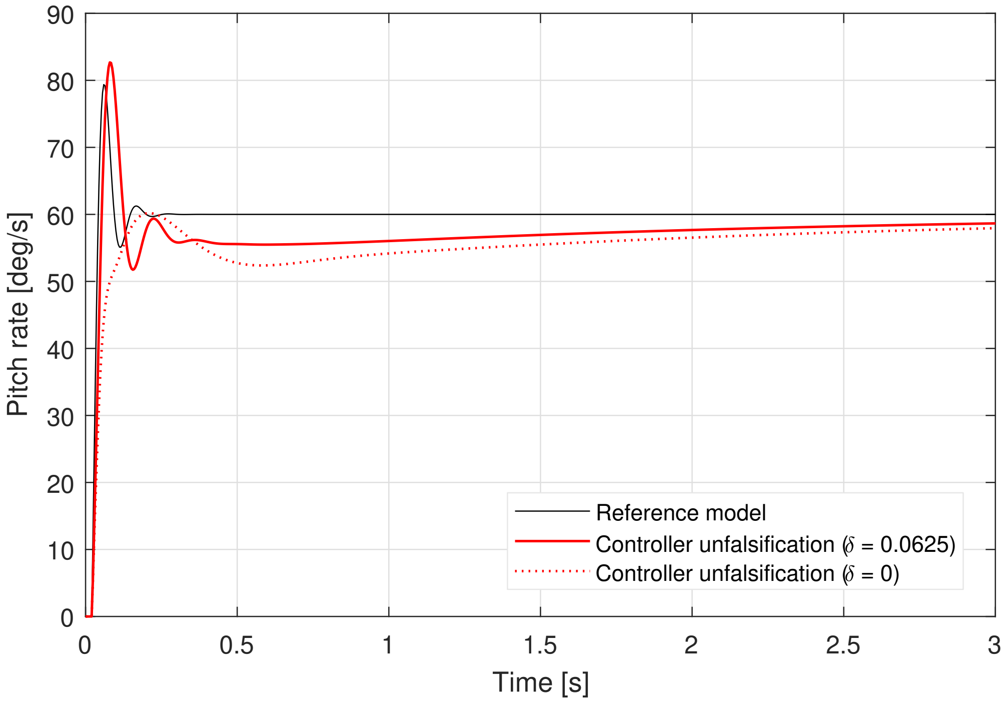


Figure 6.11: Inner loop step responses considering the controller unfalsification tuning with unachievable reference model and the stabiliser controller ($\delta = 0$) (simulation).

The Bode diagram and the step response obtained with this reference model are shown in Figure 6.10 and Figure 6.11. On the same graph, the curves associated to the stabilising controller have been plotted. This controller has been obtained by minimising the cost function (2.11) with $\delta = 0$. By doing so, the input sensitivity function is forced to be as close as possible to the desired one and thus the closed-loop stability is guaranteed (see Section 2.4). 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. Nevertheless, it has been proved that the yielded controller is actually stabilising, contrary to what it would have been achieved with the VRFT.

Controller parameter values. The controller parameters obtained with the VRFT tuning applied to the MISO PID architecture are summarized in Table 6.3. Table 6.4 shows the tunings generated by VRFT, CbT and controller unfalsification algorithms by adopting the SISO PID control architecture. In particular, regarding the controller unfalsification method, the results of all three tunings described above have been reported.

	K_{FF_i}	K_{P_i}	K_{I_i}	K_{D_i}
VRFT	**0**	**0.09143**	**0.2067**	**0.0016**
H_∞	**0**	**0.0849**	**0.2138**	**0.0014**

Table 6.3: Inner controller parameters considering VRFT with MISO PID architecture.

	K_{P_i}	K_{I_i}	K_{D_i}
VRFT	**0.05146**	**0.1935**	**0.0012**
CbT	**0.04696**	**0.06792**	**0.00091**
Controller Unfalsification ($\delta = 0$)	**0.02936**	**0.1763**	**0.0010**
Controller Unfalsification	**0.06280**	**0.3303**	**0.0017**
Controller Unfalsification ($\omega_n = 65$ rad/s)	**0.06940**	**0.2741**	**0.0010**

Table 6.4: Inner controller parameters considering VRFT, CbT and controller unfalsification methods with SISO PID architecture.

Outer loop results

Once the controller parameters of the inner loop are defined and the inner closed-loop performance is suitably validated, attention can be turned to the outer control loop. In particular, by tuning the outer loop 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.

As with the inner loop, the Bode diagrams and the step responses of all the tunings will be shown in the following pages and the results are summarized in Table 6.5 at the end of this subsection.

Controller transfer function. The outer loop control structure is trivial because it consists only of a proportional controller. Therefore, the vector of linear transfer functions $\beta(z)$ is reduced to a simple unit transfer function and the parameter vector θ is simply K_{P_o}, as illustrated in Figure 3.3. Unlike the inner loop, the same control architecture has been adopted for both the VRFT and the CbT tuning algorithms.

Note that, for simplicity, the controller unfalsification method has been applied only in the inner loop, 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. Indeed, if the inner loop is stable and the bandwidth of the outer reference model is realistic, then the closed-loop stability is almost certainly achieved. 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 $K_{P_o} = 35 \div 40$), but such values have never been obtained with any reference model.

VRFT tuning results with MISO PID controller. As shown in Figure 6.12 and 6.13, 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 overshot 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.

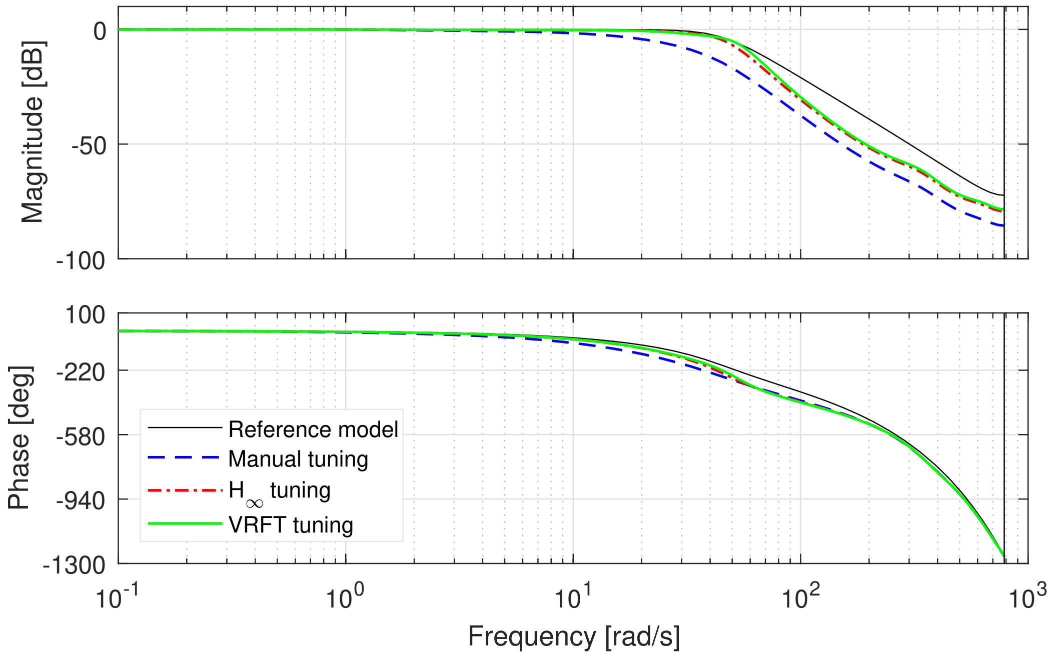


Figure 6.12: Comparison of the outer loop Bode diagrams considering manual, VRFT and H_∞ tunings (simulation).

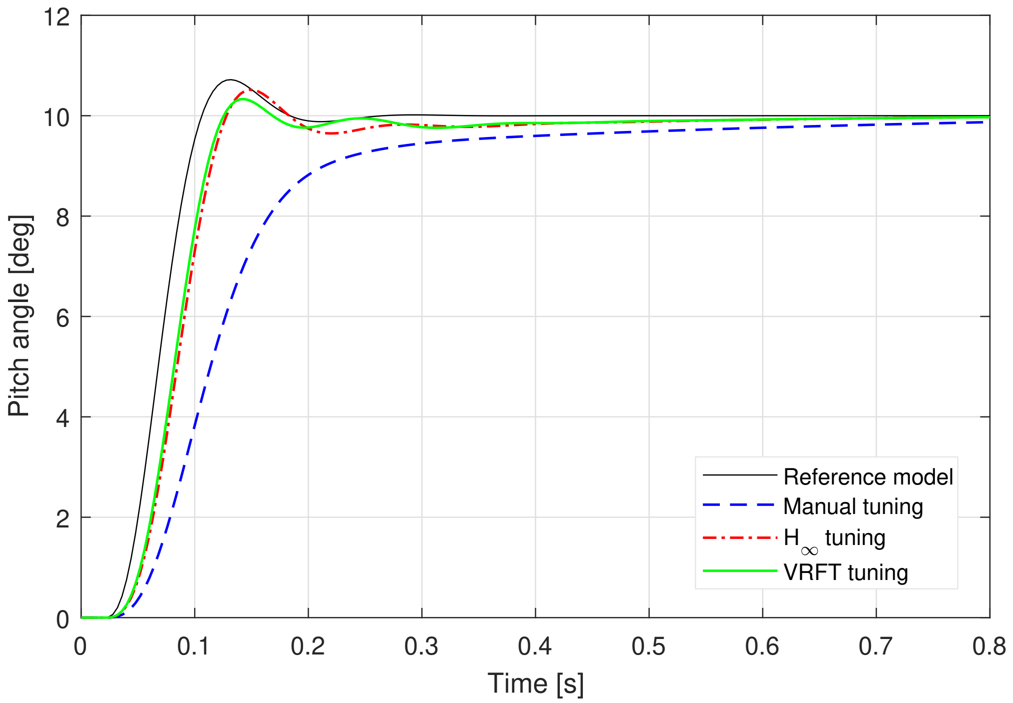


Figure 6.13: Comparison of the outer loop step responses considering manual, VRFT and H_∞ tunings (simulation).

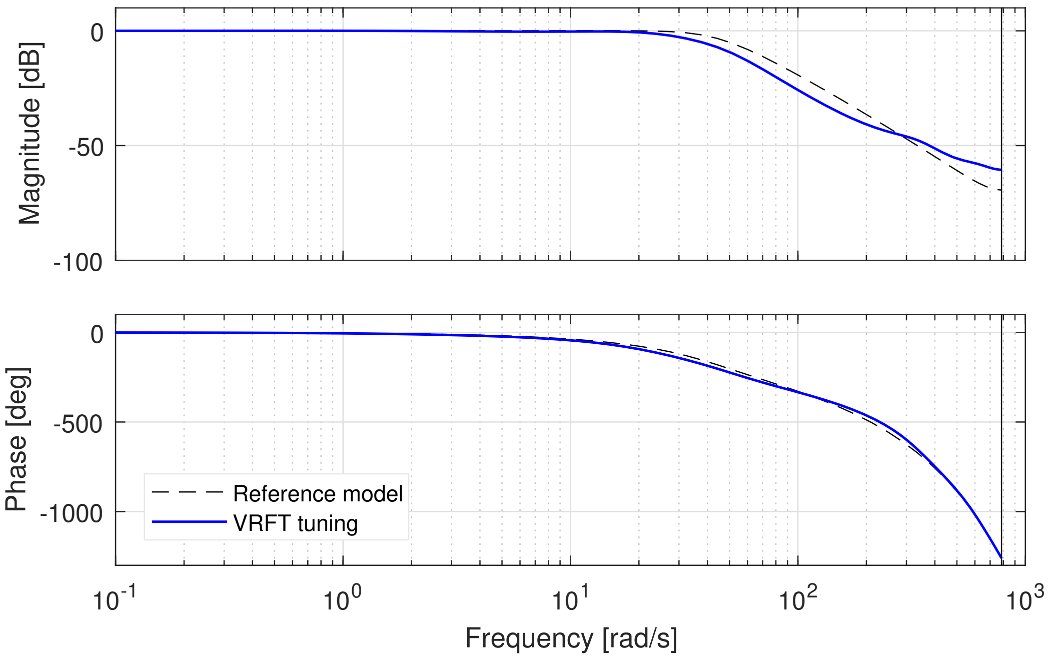


Figure 6.14: The outer loop Bode diagram considering the VRFT tuning (simulation).

VRFT, CbT and controller unfalsification tunings results with SISO PID controller. The VRFT, the CbT and the controller unfalsification methods exploit different reference models as illustrated in Table 6.2, so the results are displayed in separate figures for the sake of clarity. Figure 6.14 and Figure 6.15 show the closed-loop performance guaranteed by the VRFT tuning, while the CbT tuning results are displayed in Figures 6.16 and 6.17. The outer proportional gain, associated with the inner unfalsified controller, has been derived by means of the VRFT algorithm (Figures 6.18 and 6.19). The three methods are compared in Figures 6.20 and 6.21 without displaying the reference models. For comparison purposes, the results obtained by means of controller unfalsification tuning of the inner loop with unachievable reference model have been reported on the same graphs.

The outer loop confirms what has already been outlined by analysing the results of the inner loop. That is, the performance ensured by the CbT tuning method is overall lower than that obtained by the VRFT approach. Furthermore, by looking at the Figure 6.21 it can be noted that the unfalsified controller obtained assigning an achievable reference model has a very fast response with a minimal overshoot. On the contrary, assigning an unachievable reference model the closed-loop response tends to oscillate resulting in a long settling time. Nevertheless, thanks to the stability test, the resulting controller is stable.

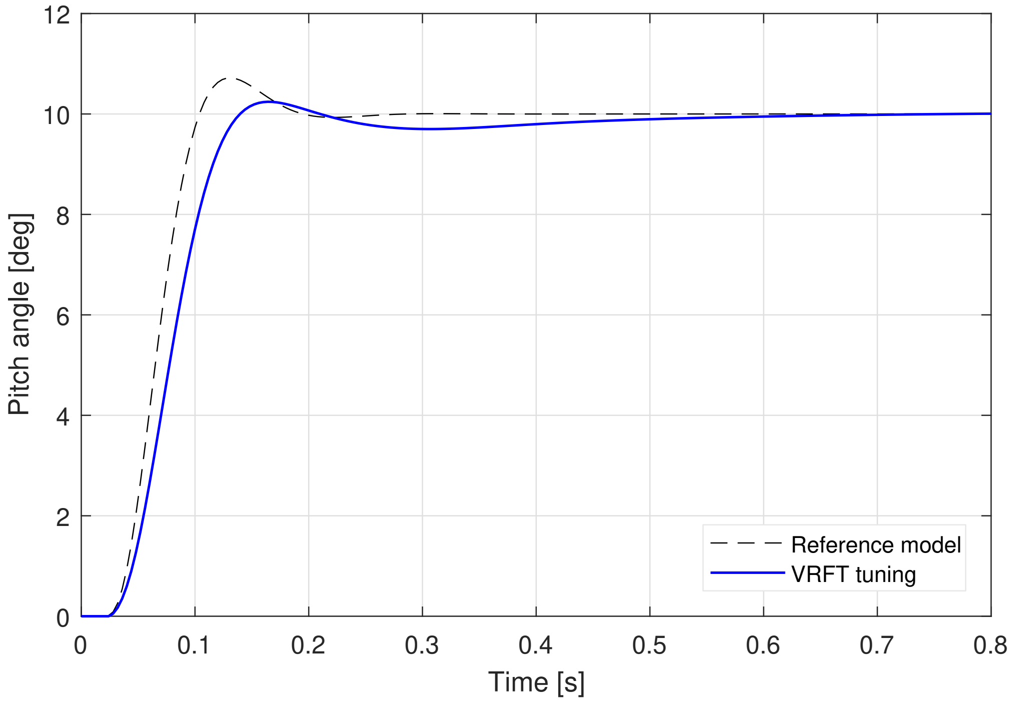


Figure 6.15: The outer loop step response considering the VRFT tuning (simulation).

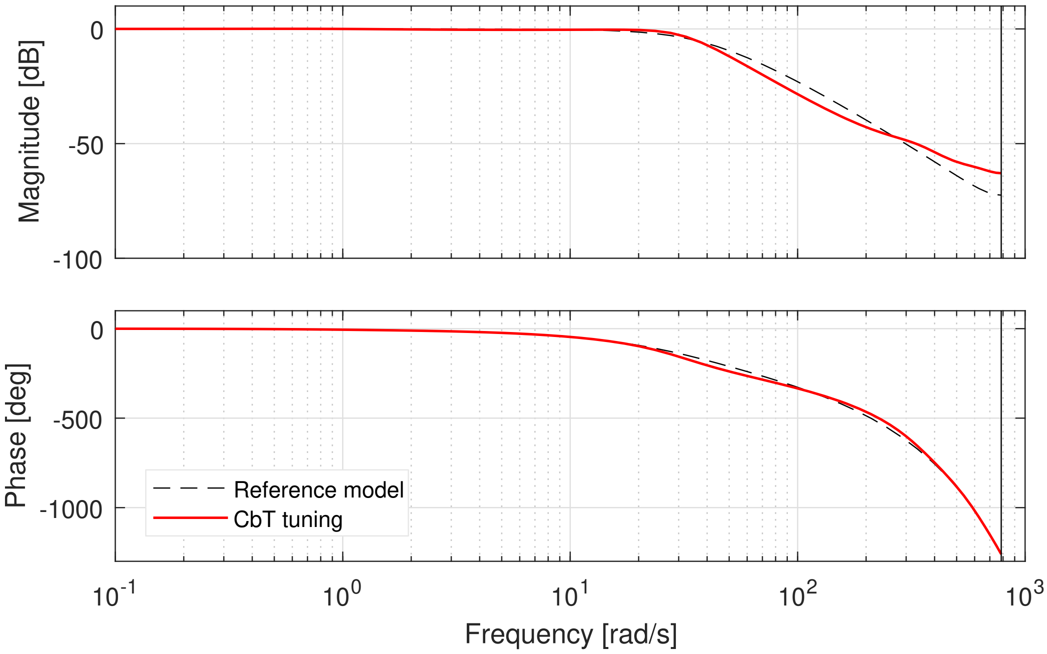


Figure 6.16: The outer loop Bode diagram considering the CbT tuning (simulation).

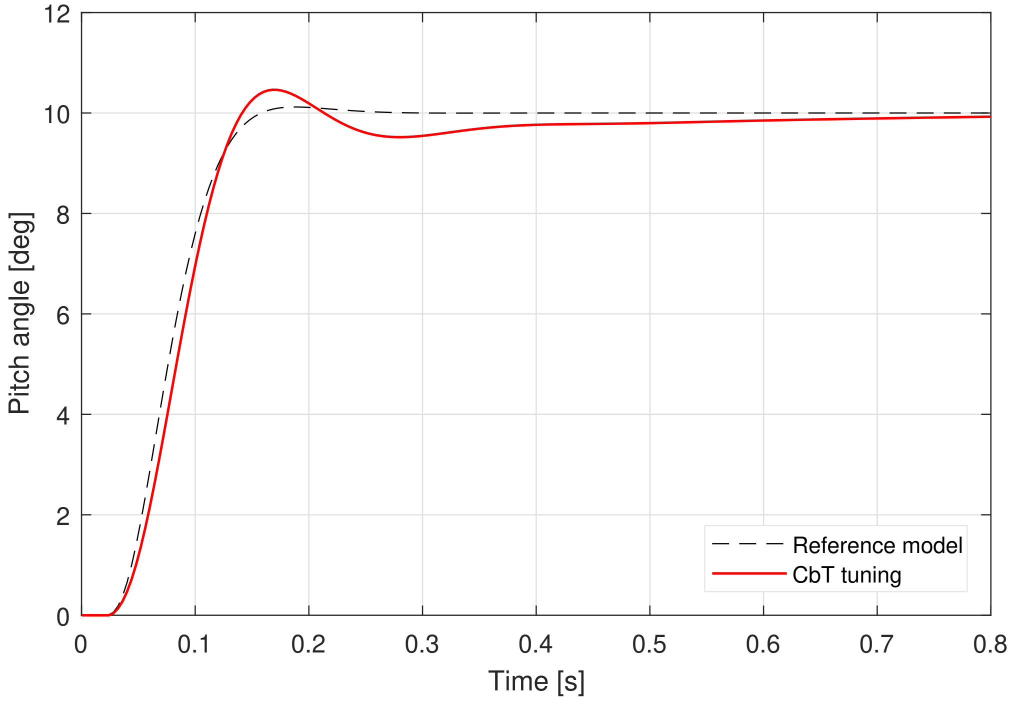


Figure 6.17: The outer loop step response considering the CbT tuning (simulation).

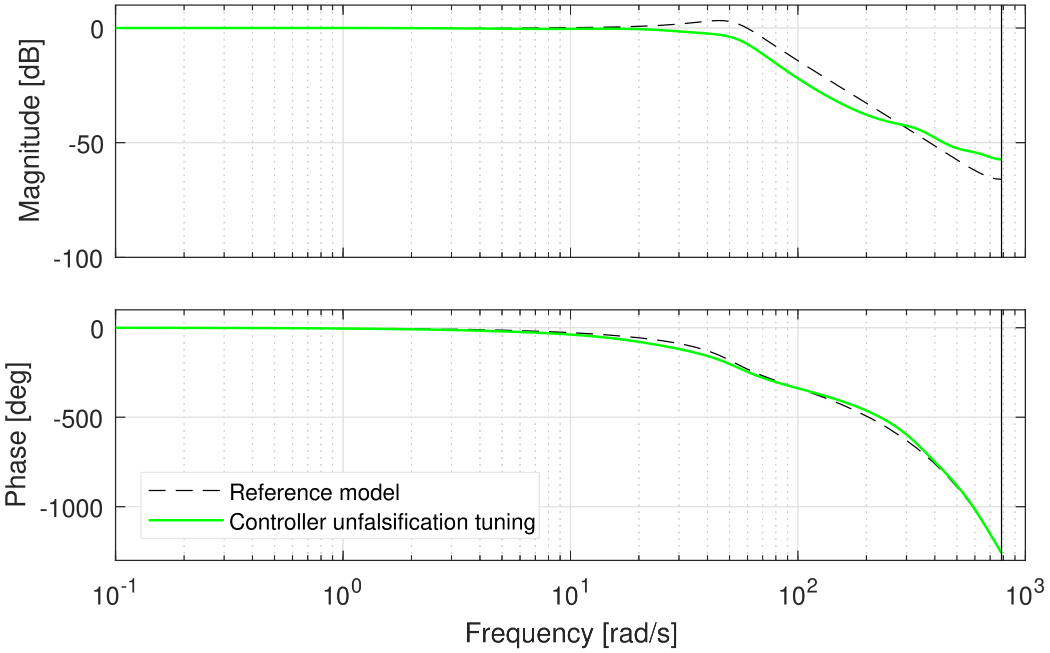


Figure 6.18: The outer loop Bode diagram considering the controller unfalsification tuning (simulation).

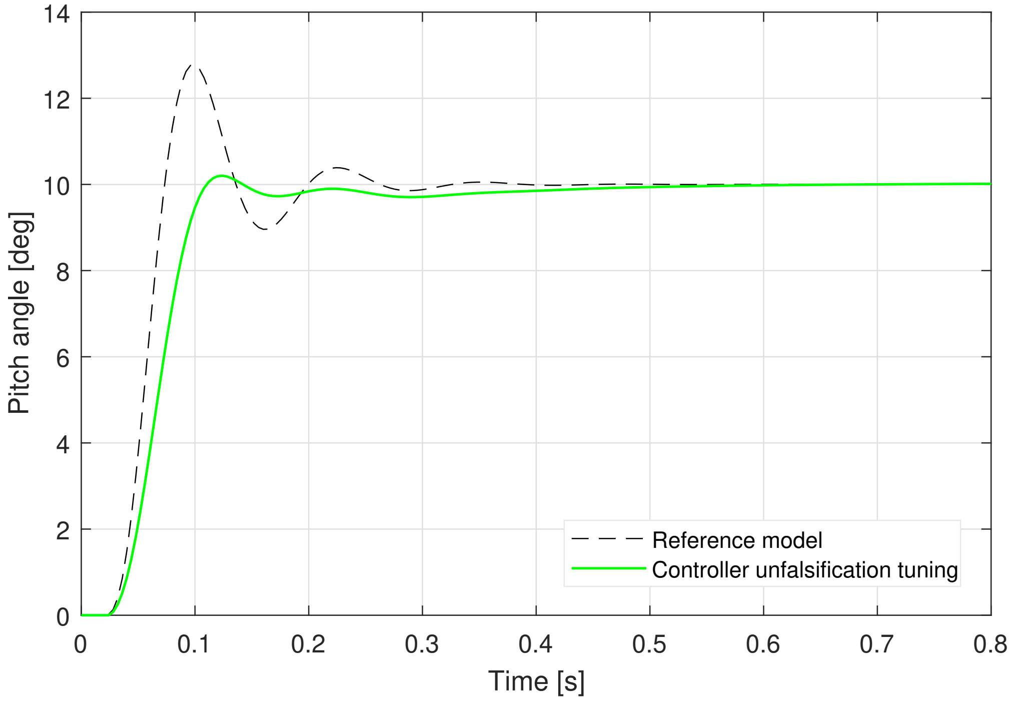


Figure 6.19: The outer loop step response considering the controller unfalsification tuning (simulation).

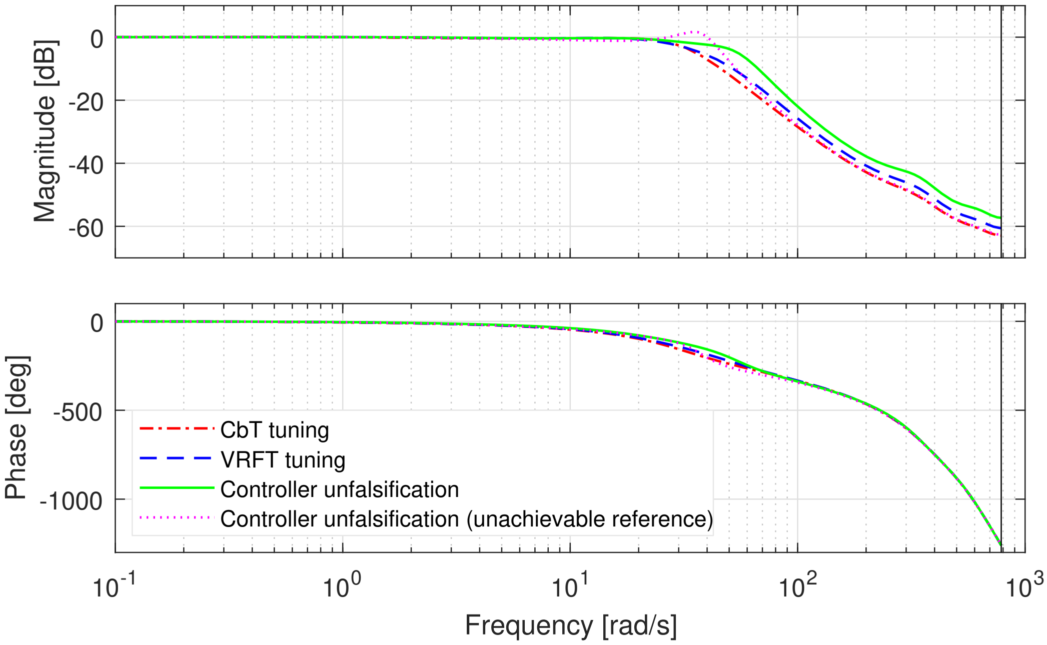


Figure 6.20: Comparison of the outer loop Bode diagrams considering VRFT and CbT tunings and controller unfalsification tunings both with achievable and unachievable reference model (simulation).

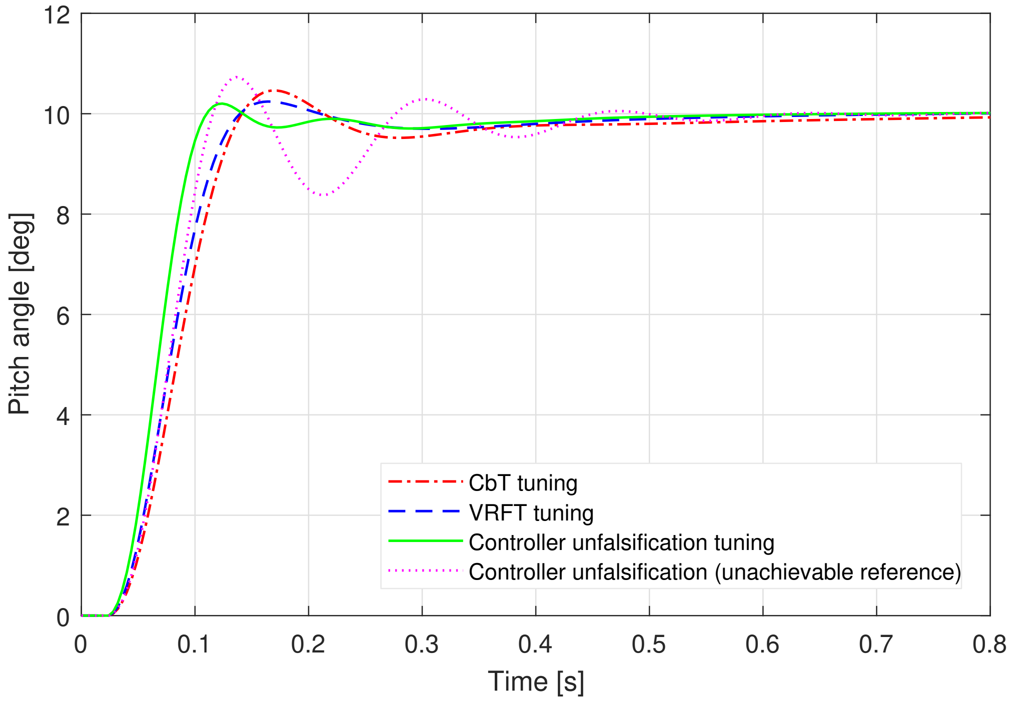


Figure 6.21: Comparison of the outer loop step responses considering VRFT and CbT tunings and controller unfalsification tunings both with achievable and unachievable reference model (simulation).

Controller parameter values. Table 6.5 summarises all the outer loop proportional gains associated with the different tunings. The parameter value associated with the unfalsification method is related to the inner loop tuned with achievable reference model.

	H_∞	VRFT (MISO PID)	VRFT (SISO PID)	CbT	VRFT (Unfalsification)
K_{P_o}	**11.75**	**12.11**	**12.78**	**12.92**	**14.10**

Table 6.5: Proportional gains of the outer controller considering VRFT, CbT and H_∞ methods.

6.1.4 Experimental results

As already stated in Subsection 6.1.3 the controllers design have been performed on a simulated system to have much faster iterations between the different reference models and thus between subsequent controller tunings. 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 a 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 characterization. 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.

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 tested data-driven controllers have been obtained with the reference models described in Tables 6.2 and 6.1.

Setpoint tracking and disturbance rejection with VRFT tuning (MISO PID controller)

The performance obtained with the VRFT algorithm has been compared with H_∞ and manual tunings. Figure 6.22 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 6.23) and 10 deg (Figure 6.24). 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 (Figures 6.25 and 6.26) the performance of the VRFT 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 ensures significantly better performance with respect to the manual tuning but, at the same time, require an higher control effort.

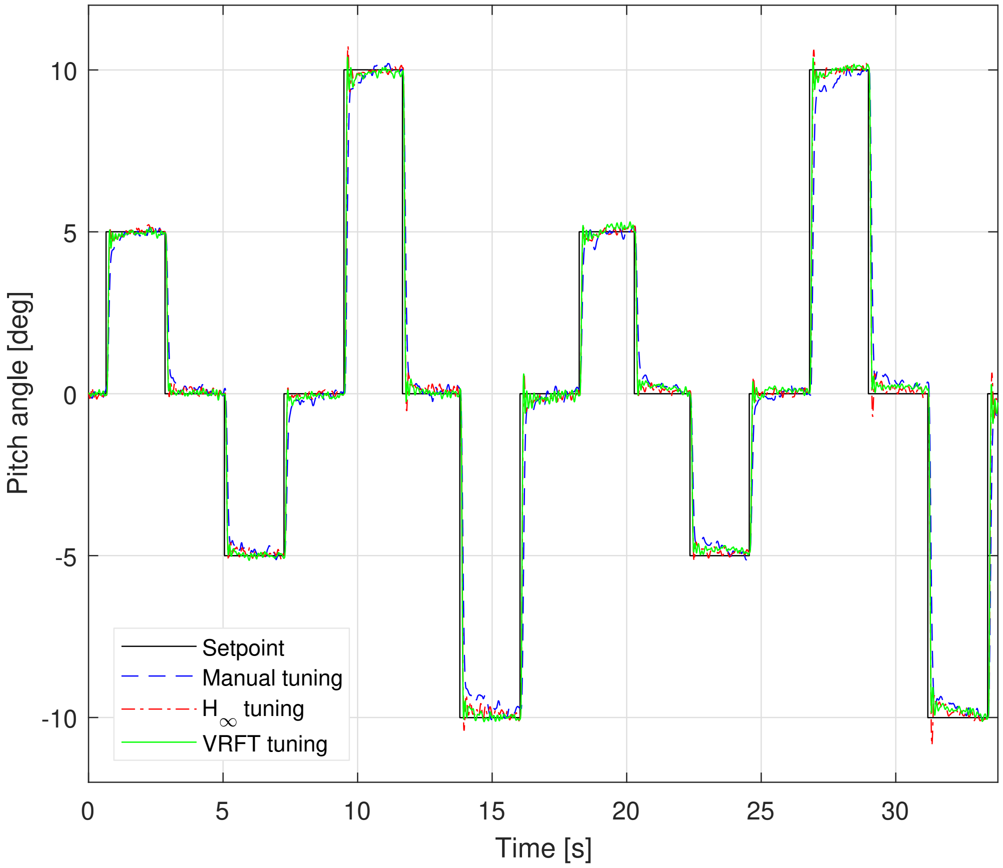


Figure 6.22: Setpoint tracking with manual, H_∞ and VRFT tunings (experiment).

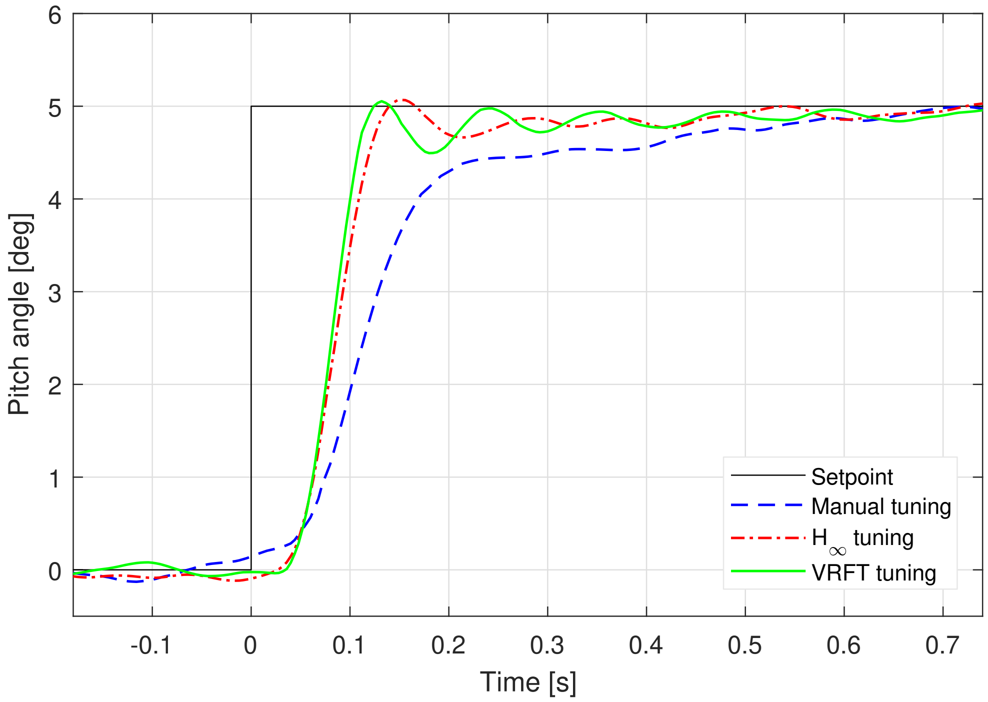


Figure 6.23: Setpoint tracking (5 deg step) with manual, H_∞ and VRFT tunings (experiment).

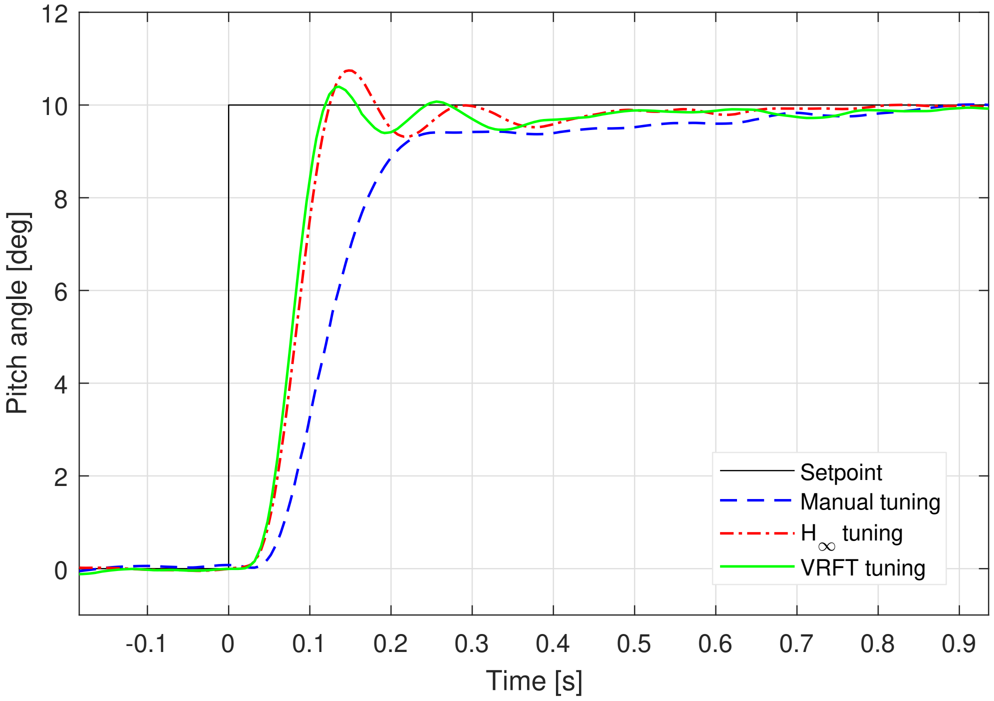


Figure 6.24: Setpoint tracking (10 deg step) with manual, H_∞ and VRFT tunings (experiment).

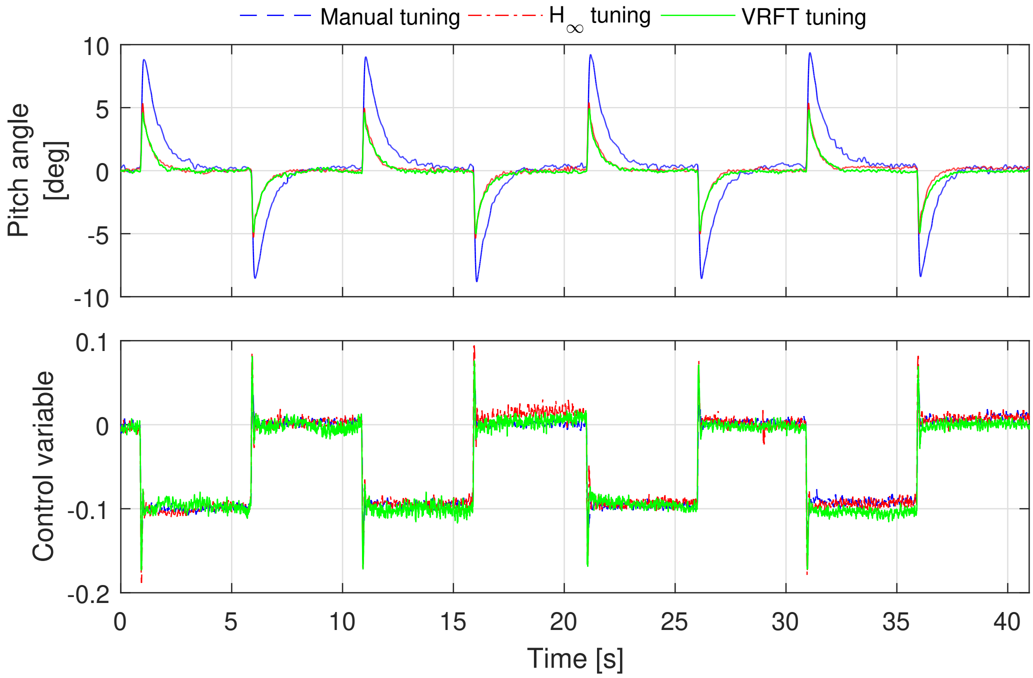


Figure 6.25: Load disturbance rejection with manual, H_∞ and VRFT tunings. (experiment)

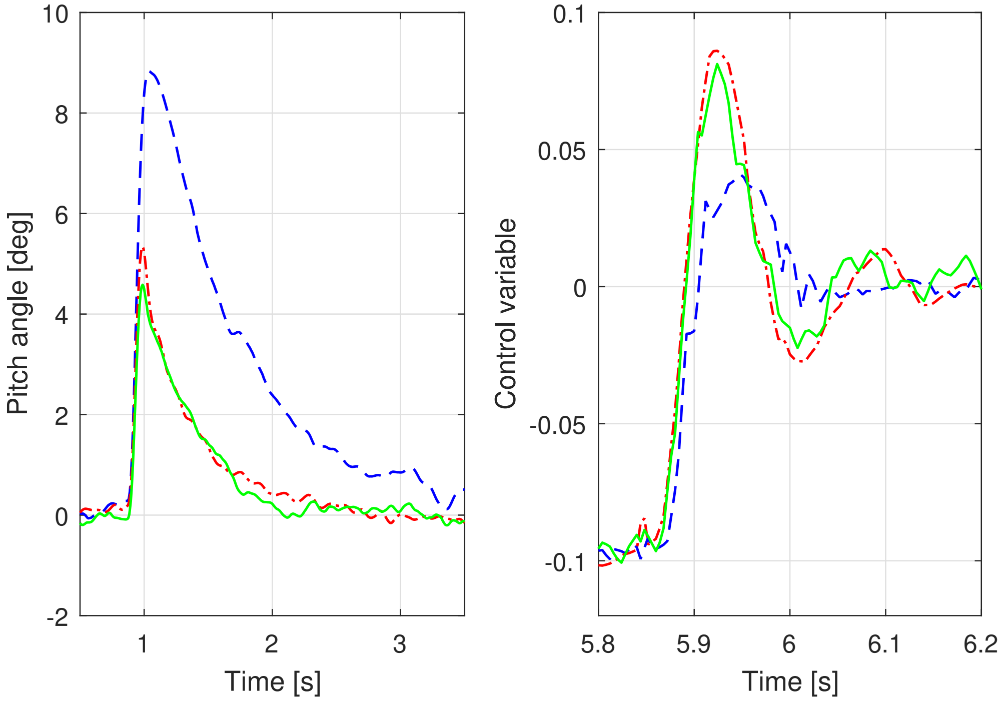


Figure 6.26: Zoomed-in view of the load disturbance rejection with manual, H_∞ and VRFT tunings.

Setpoint tracking and disturbance rejection with VRFT, CbT and controller unfalsification methods (SISO PID controller)

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 6.27, 6.28 and 6.29 show the setpoint tracking tests of VRFT, CbT and the controller unfalsification tunings.

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 (Figures 6.30 and 6.31) 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.

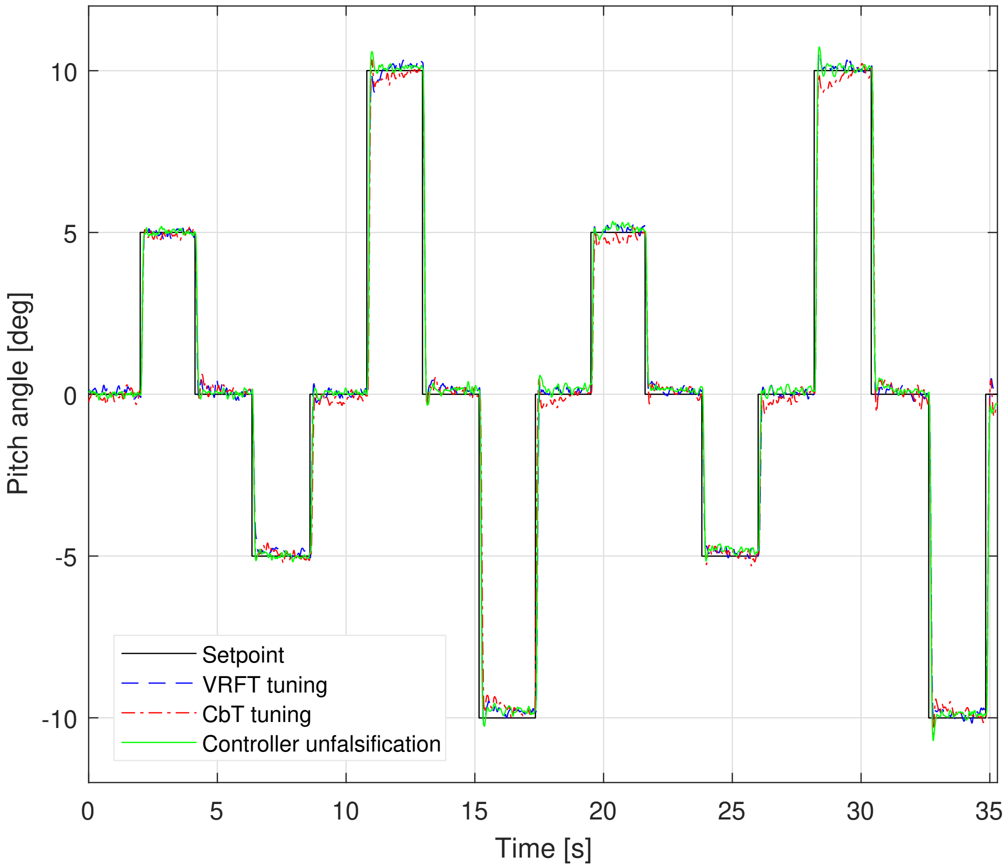


Figure 6.27: Setpoint tracking with VRFT, CbT and controller unfalsification tunings (experiment).

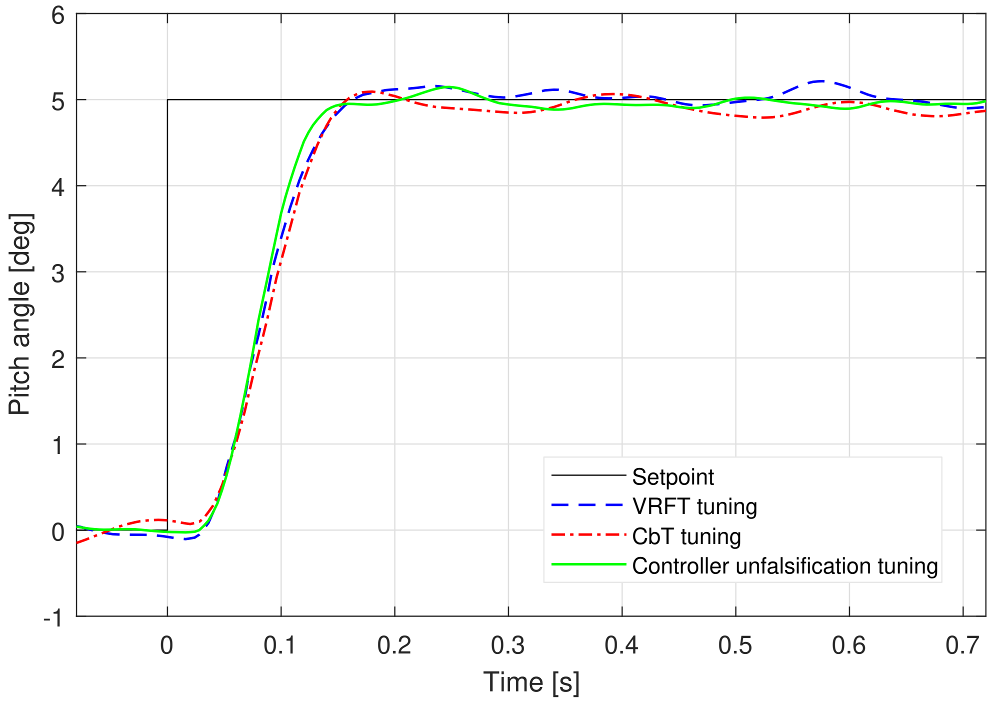


Figure 6.28: Setpoint tracking (5 deg step) with VRFT, CbT and controller unfalsification tunings (experiment).

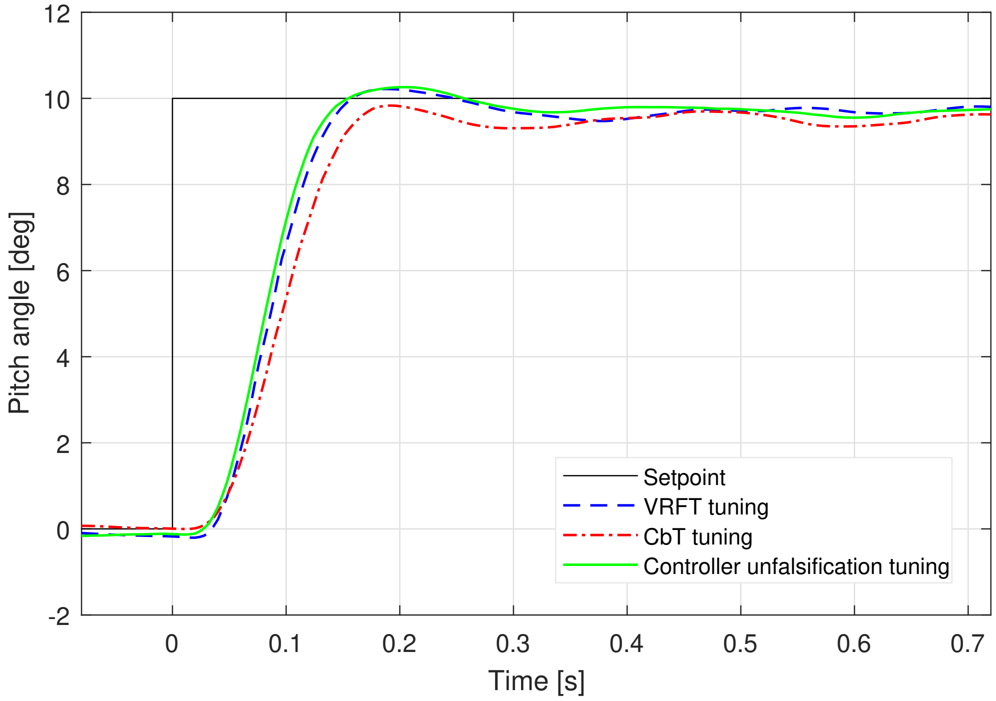


Figure 6.29: Setpoint tracking (10 deg step) with VRFT, CbT and controller unfalsification tunings (experiment).

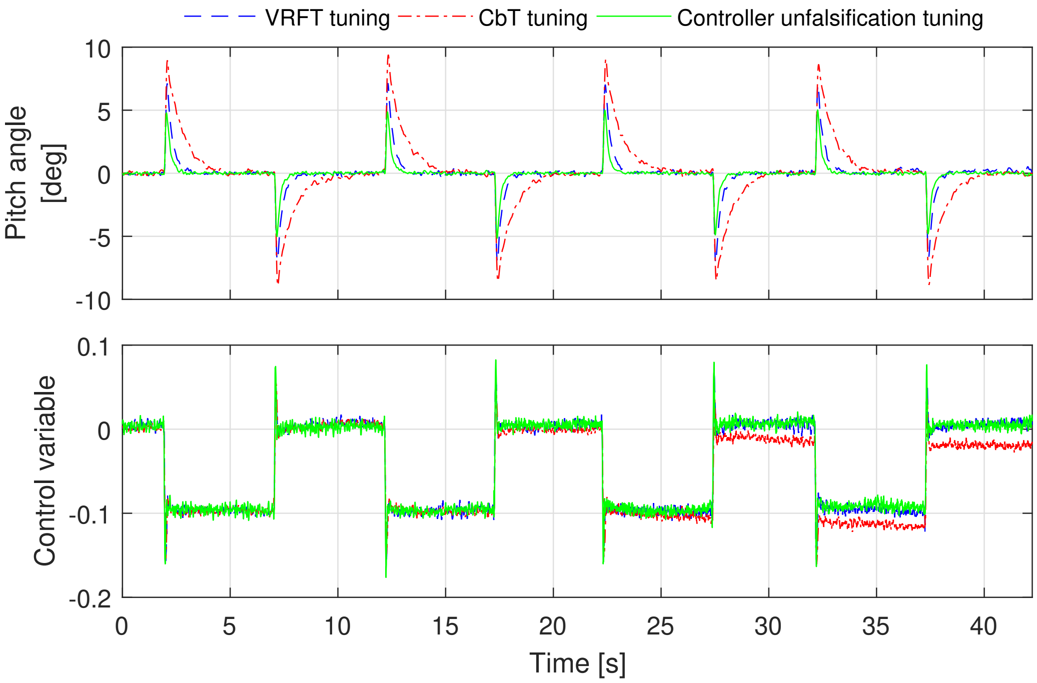


Figure 6.30: Load disturbance rejection with VRFT, CbT and controller unfalsification tunings (experiment).

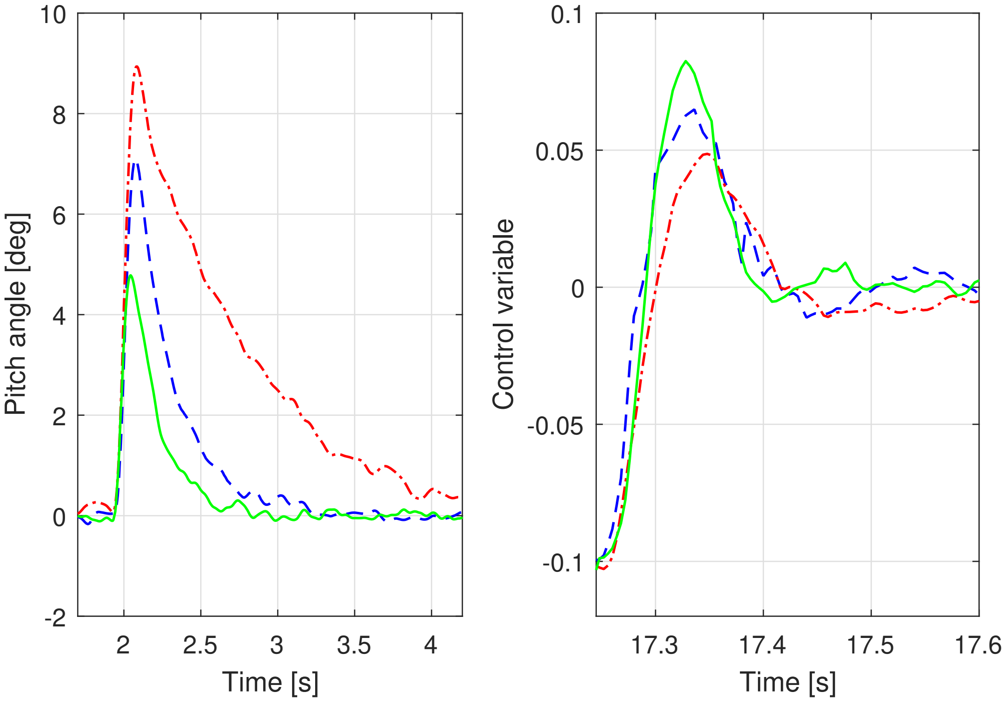


Figure 6.31: Zoomed-in view of the load disturbance rejection with VRFT, CbT and controller unfalsification tunings (experiment).

Controller parameter values

To conclude, the parameters of all the tested controllers are summarised in Tables 6.6 and 6.7. The VRFT results described above, and the related controller parameters shown in the table, have been obtained by building the instrumental variable through the identification of an ARX(5,5) model for the inner and the outer loops. As already stated in Section 2.2, this is a fundamental step in order to deal with noisy experimental data.

	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 6.6: Optimal controller parameters for MISO PID control architecture.

	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 6.7: Optimal controller parameters for SISO PID control architecture.

6.2 Pitch attitude controller with closed-loop experiments

In Section 6.1 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 3.3 is exploited. The results will be compared with the VRFT tuning obtained with open-loop experimental data as presented in Section 6.1.

6.2.1 Tuning experiment

As illustrated in Section 4.1, an initial controller $C_d(z)$ that stabilises the system must be available, in order to conduct the tuning experiment in closed-loop. This controller has the same structure of the one to be tuned and the manual tuning, as presented in Table 6.6, emerges as the most obvious choice, since it is the first available controller and it was obtained with a simple trial and error procedure.

As illustrated in Figure 4.1, the excitation signal $\bar{u}(t)$ is added to the output of the controller $C_d(z)$. The time histories of all the involved signals in the data-driven tuning procedure are illustrated in the Figure 6.32. In this case, the control variable that is the pitch moment applied to the UAV (it is illustrated in the second plot of Figure 6.32) 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.

6.2.2 Reference models

Comparing this section and Section 6.1, 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 models for the inner and outer

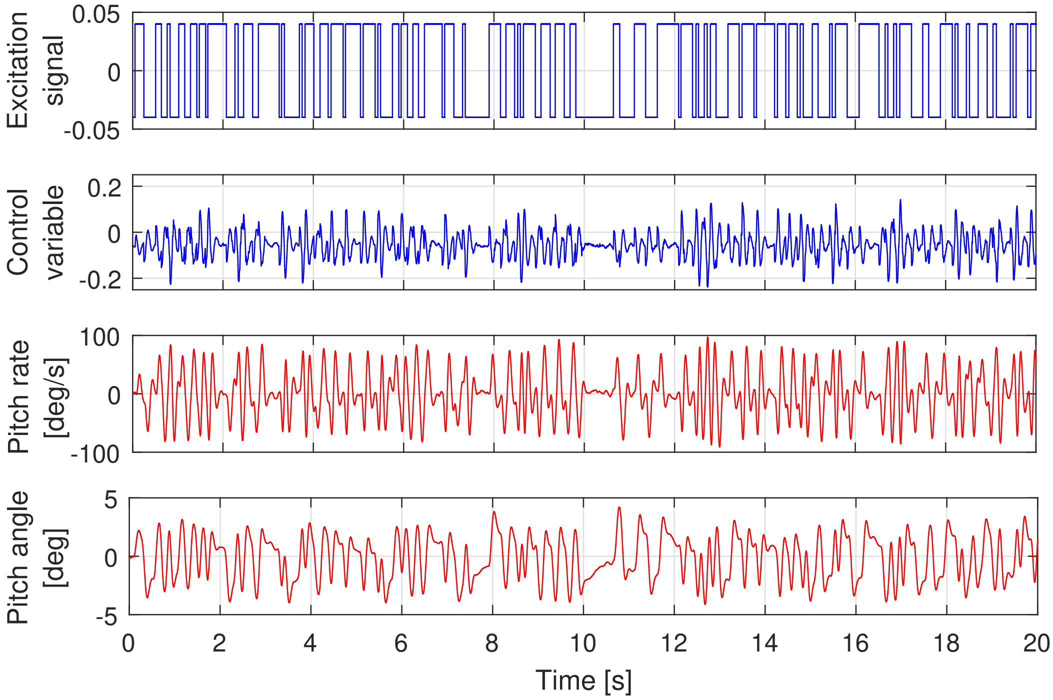


Figure 6.32: Closed-loop experimental dataset used by data-driven method.

loop are expressed in (6.2) and (6.3) respectively. 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 6.1.

6.2.3 Controller parameter values

Exploiting the reference models and closed-loop experimental data the VRFT method leads to the parameter values reported in Table 6.8. Both the parameters for the inner and the outer controllers are displayed. For comparative purposes also the tuning obtained with open-loop data (see Section 6.1) 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.

6.2.4 Simulation results

Also in this case the plant model in (6.5) and the scheme in Figure 6.2 are exploited to better show the results.

	K_{FF_i}	K_{P_i}	K_{I_i}	K_{D_i}	K_{P_o}
VRFT with open-loop data	**0**	**0.09143**	**0.2067**	**0.0016**	**12.11**
VRFT with closed-loop data	**0**	**0.1359**	**0.1008**	**0.0019**	**12.47**

Table 6.8: Optimal controller parameters for outer and inner controllers considering the VRFT method with open-loop and closed-loop experimental data.

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

Just to recall, in Section 6.1 it has been shown that 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 have been done on the step responses of the two tunings (Figure 6.5).

Concerning the VRFT tuning obtained with closed-loop gathered data, the frequency response is quite similar to that the VRFT based on open-loop data, as shown in the Bode diagram in Figure 6.33. 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 6.34).

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 as output (see again Figure 3.3). As illustrated in Figures 6.35 and 6.36, 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 method is slightly more oscillating.

6.2.5 Experimental results

As explained in Section 6.1, 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 6.1: 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 6.37 shows the entire time history of the setpoint tracking test

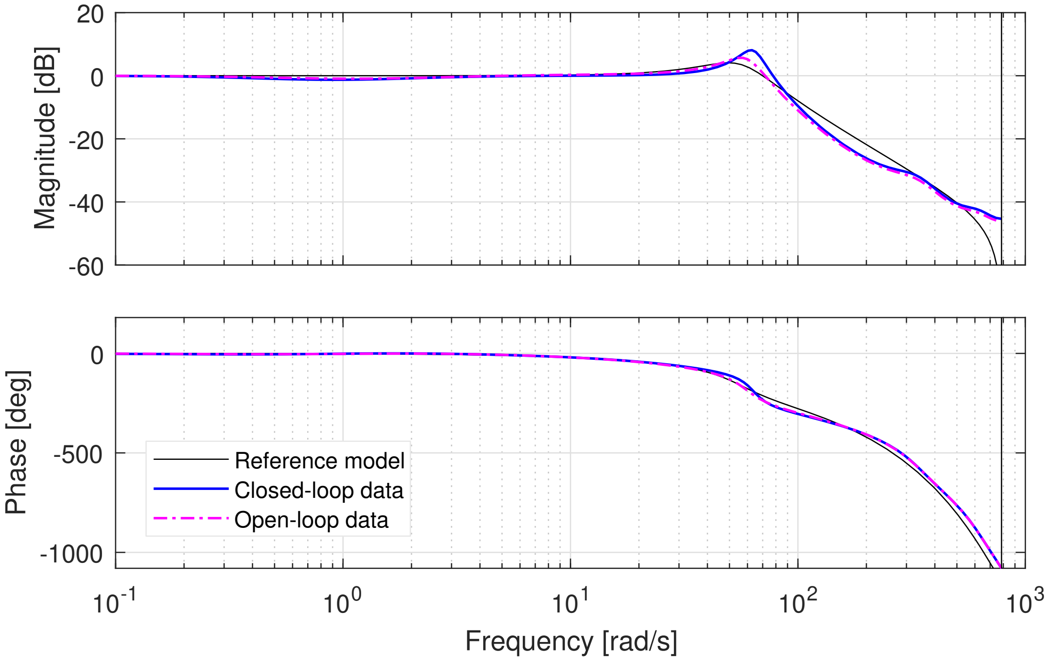


Figure 6.33: Comparison of the inner loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

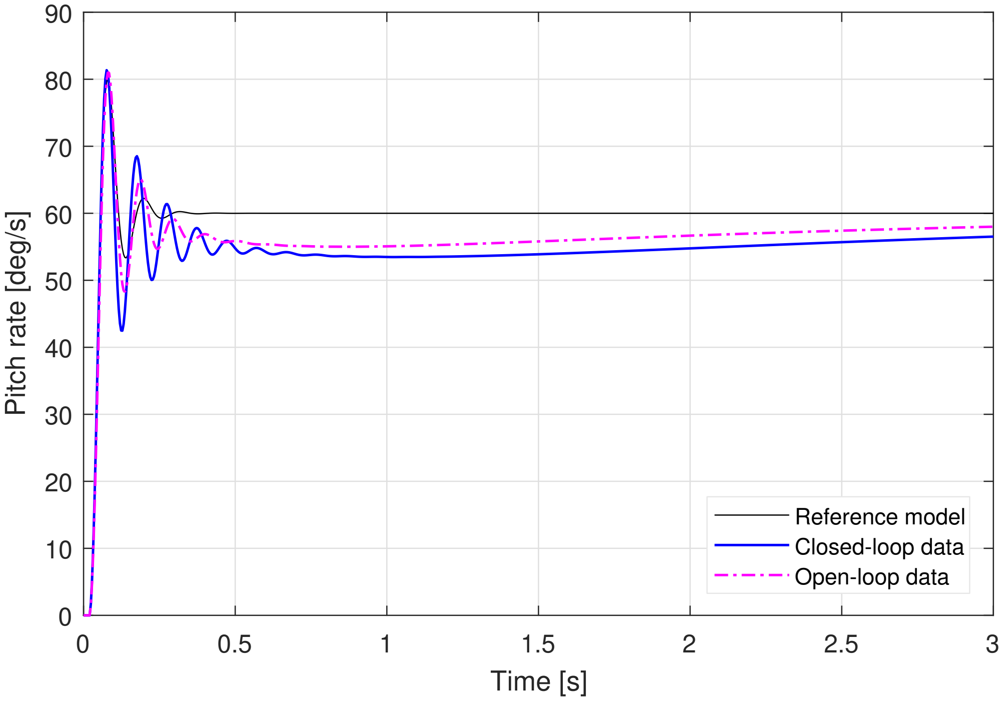


Figure 6.34: Comparison of the inner loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

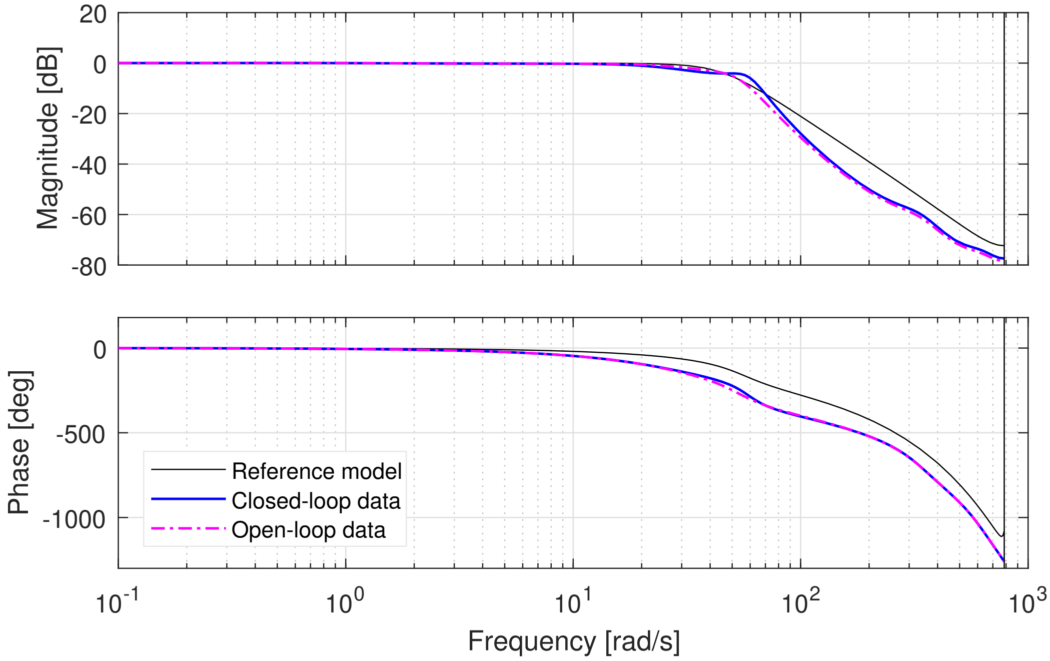


Figure 6.35: Comparison of the outer loop Bode diagrams considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

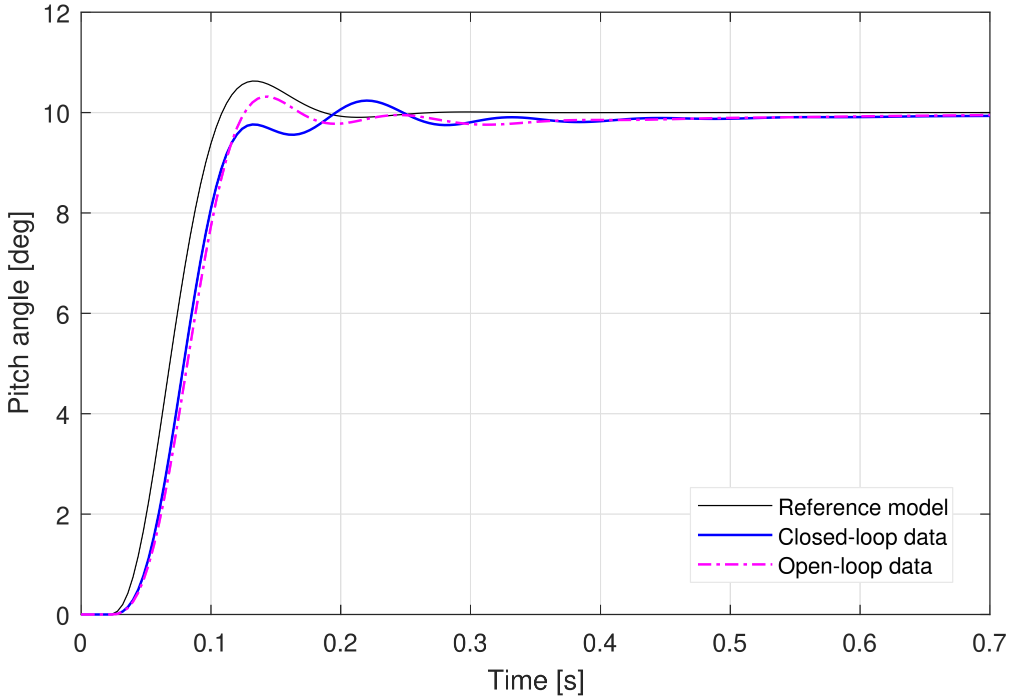


Figure 6.36: Comparison of the outer loop step responses considering VRFT with closed-loop data and VRFT with open-loop data (simulation).

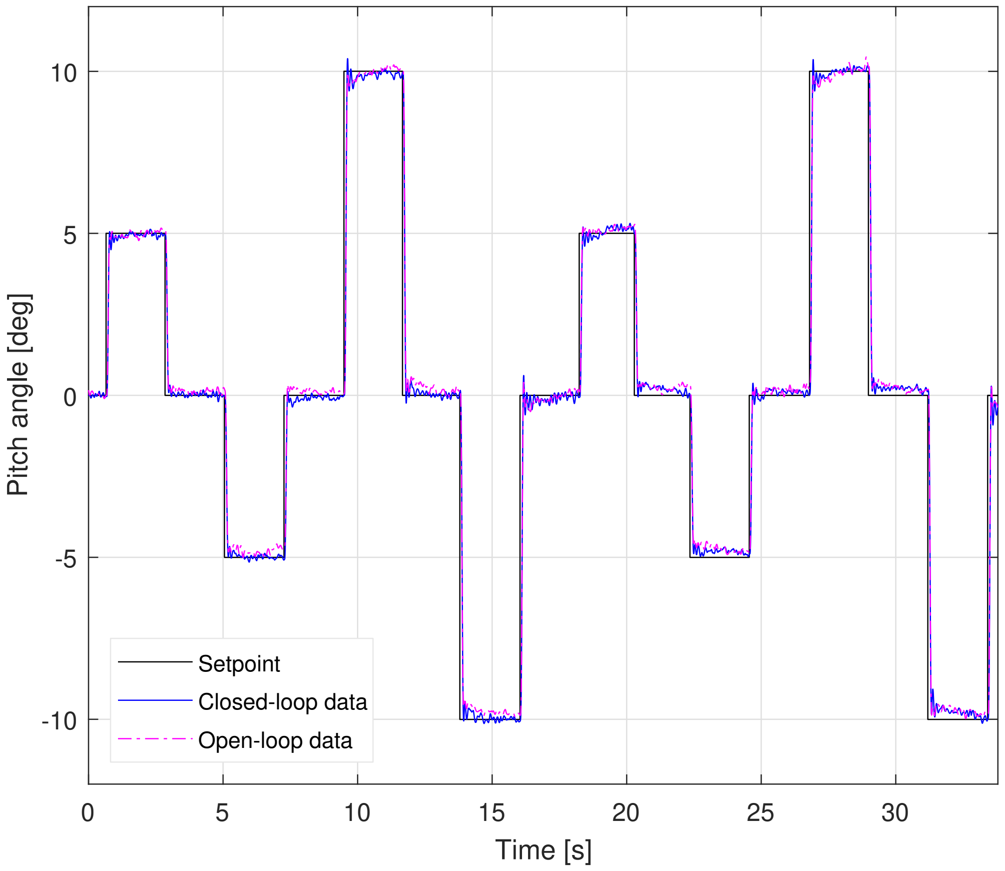


Figure 6.37: Setpoint tracking comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

and an enlargement of a single step response has been reported for both amplitudes, 5 deg and 10 deg in Figure 6.38 and in Figure 6.39 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 analogous to that of the VRFT tuning that exploits open-loop data.

When considering the disturbance rejection test (Figures 6.40 and 6.41) 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.

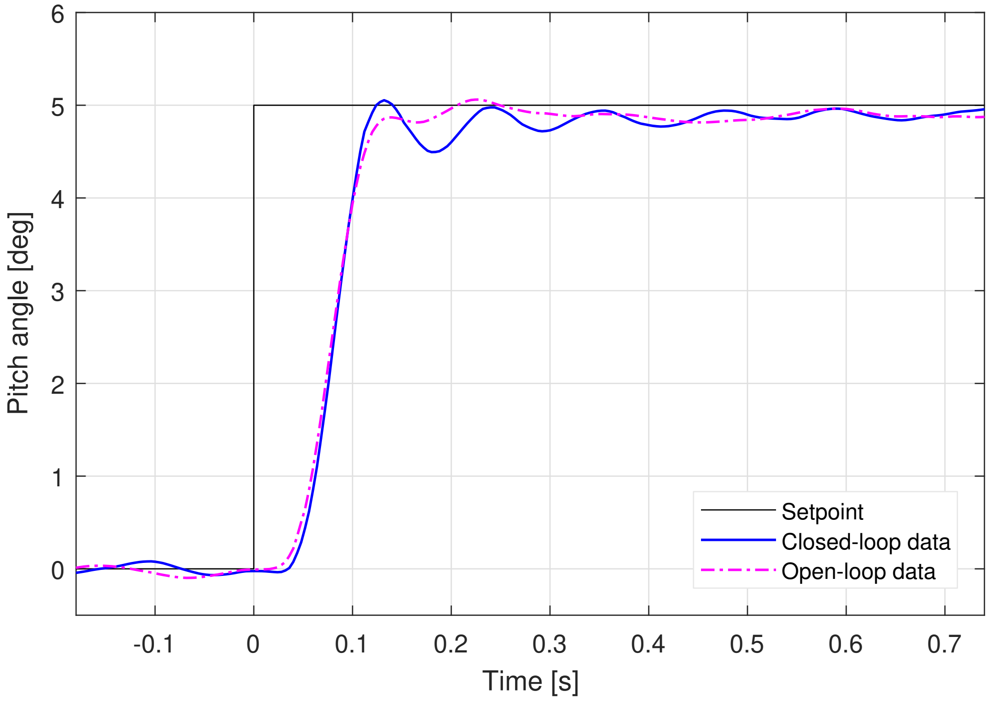


Figure 6.38: Setpoint tracking (5 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

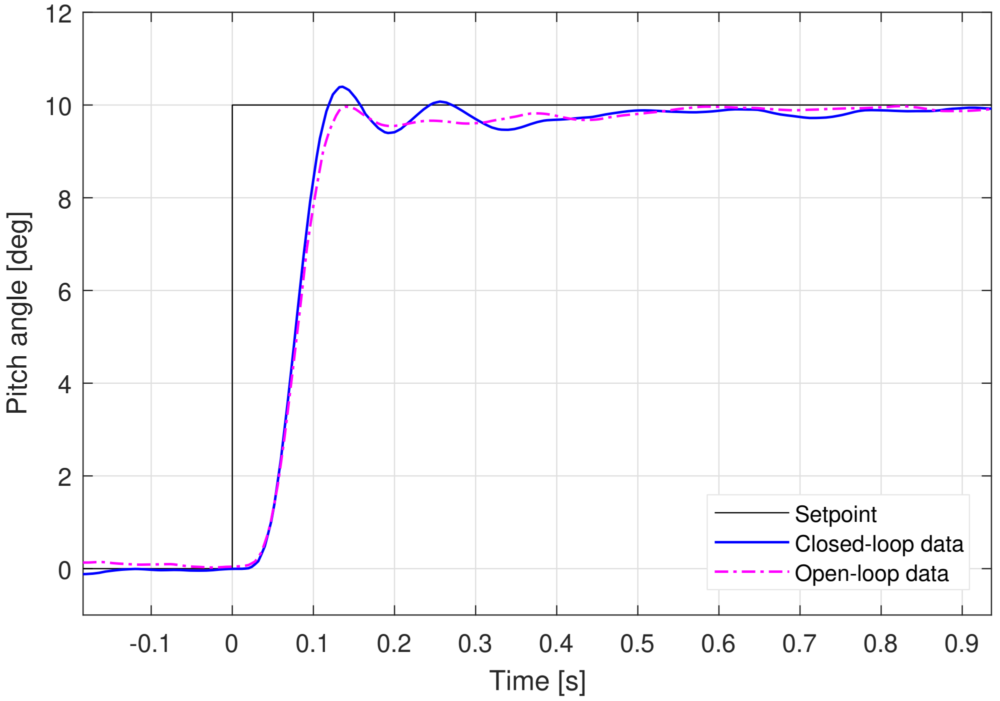


Figure 6.39: Setpoint tracking (10 deg step) comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

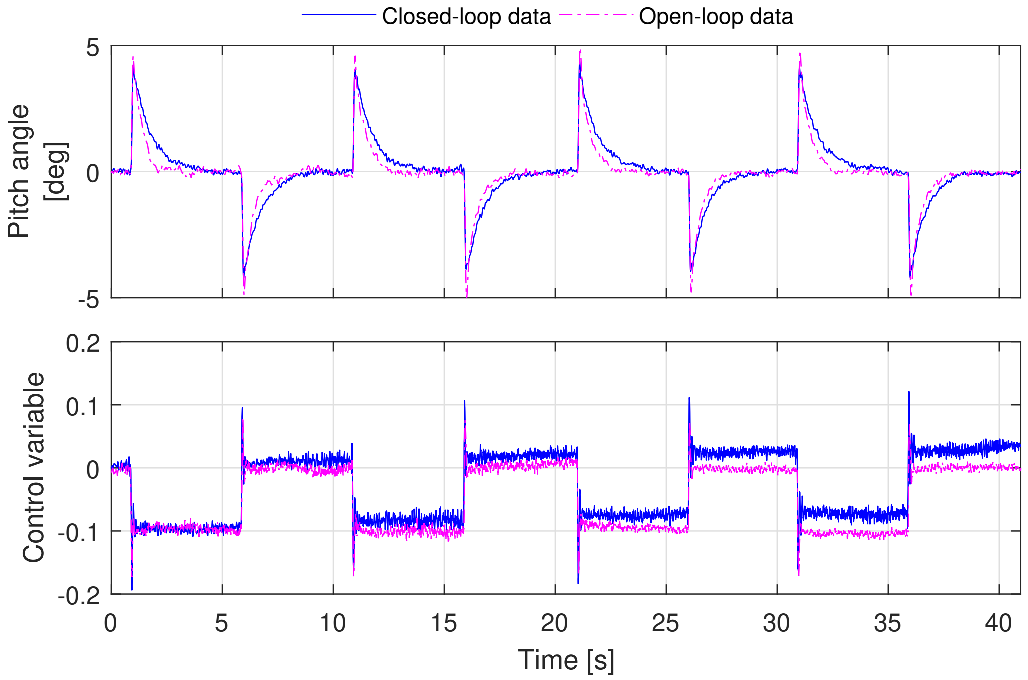


Figure 6.40: Load disturbance rejection comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

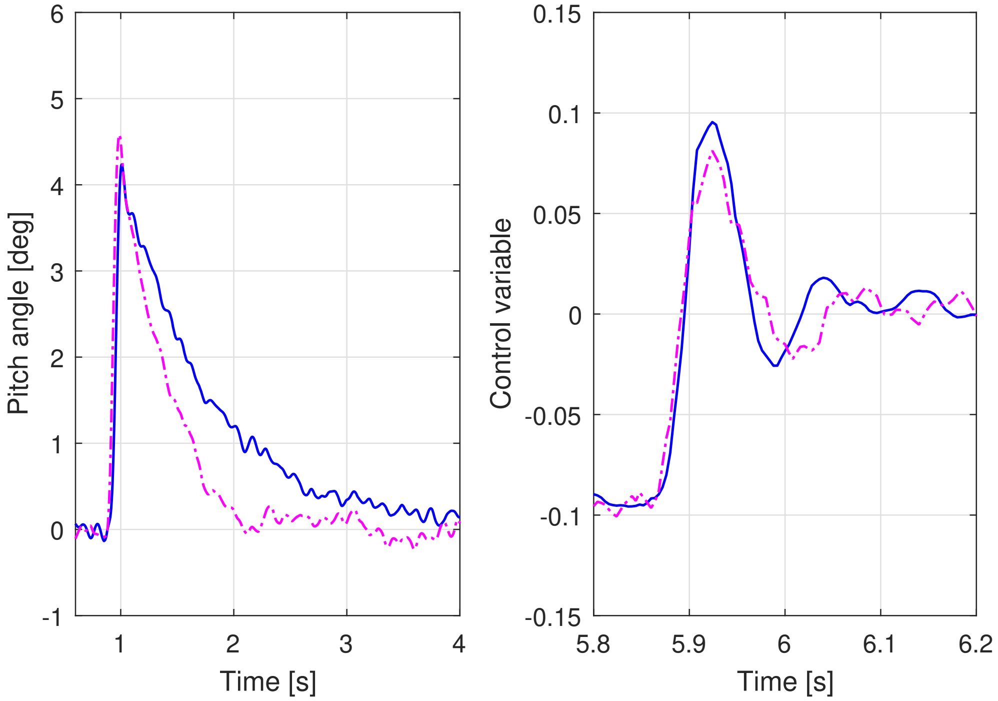


Figure 6.41: Zoomed-in view of the load disturbance rejection comparing VRFT with closed-loop data and VRFT with open-loop data (experiment).

6.3 Position controller

In this section, the results of the position controller tuning will be discussed in detail. In particular, an overview of the tuning experiment will be provided in Subsection 6.3.1. Then, after defining the reference models of inner and outer loops (Subsection 6.3.2), all the simulated MATLAB results will be shown in Subsection 6.3.3. Finally, the results of the validating experiments, which consist of in-flight tests, will be provided (Subsection 6.1.4).

For the sake of simplicity, only the VRFT method is considered and the original control architecture displayed in Figure 3.4 is exploited. Since this controller was never previously tuned, the closed-loop results will be compared with the closed-loop reference model.

6.3.1 Tuning experiment

The FCU of the ANT-1 quadrotor has been modified to perform the required excitation test. Furthermore, thanks to the small size of the UAV the test has been performed indoor in the flight 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 and aligned with the north reference in the centre of the arena. The thrust required in hovering conditions is kept constant throughout the test. In this situation, a PRBS pitch reference has been 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, while the position controllers are disabled (see Section 3.3). 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. Furthermore, as an additional safety measure, the test could be aborted at any time by using the remote controller.

The input and output time histories used for the data-driven controller design are shown in Figure 6.42. 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 5 seconds excitation tests have been 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

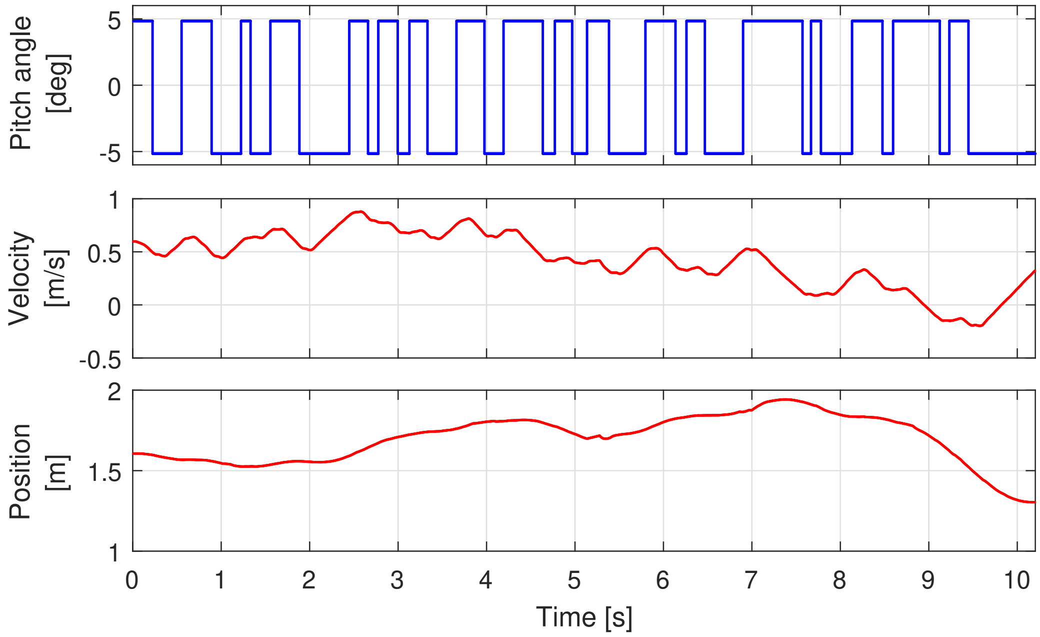


Figure 6.42: Open-loop experimental dataset used by data-driven method.

an array of infra-red sensitive cameras, provides the position of the UAV inside an indoor flight arena.

6.3.2 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 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 has been added and a check of the number of poles and zeros of the system has been carried out. By approximating the position dynamics model with a first-order transfer function of the form

$$P_i(s) = \frac{g}{s}, \tag{6.8}$$

where g is gravitational constant (see Subsection 6.3.3 for further details on plant model), and assuming the PID control architecture displayed in Figure 3.4, whose transfer functions are defined as

$$C_{PI}(s) = \frac{k_{P_i}s + k_{I_i}}{s} \quad \text{and} \quad C_D(s) = k_{D_i}s\,,$$

the closed-loop transfer function can be computed as follow:

$$\begin{aligned} F_i(s) &= \frac{C_{PI}(s)P_i(s)}{1 + C_D(s)P_i(s) + C_{PI}(s)P_i(s)} = \\ &= \frac{\dfrac{gk_{P_i}s + gk_{I_i}}{s^2}}{1 + gk_{D_i}s + \dfrac{gk_{P_i}s + gk_{I_i}}{s^2}} = \frac{gk_{P_i}s + gk_{I_i}}{(1 + gk_{D_i})s^2 + gk_{P_i}s + gk_{I_i}}\,. \end{aligned} \tag{6.9}$$

Looking at the Equation (6.9) it is clear that a second-order transfer function does not provide a good approximation of the actual closed-loop behaviour. More specifically, the reference model should include an additional zero, resulting in a continuous time transfer function of the form

$$M_i(s) = \frac{\omega_{ni}^2}{s^2 + 2\xi_i\omega_{ni}s + \omega_{ni}^2}\,\frac{s + \zeta_i}{\zeta_i}\,,$$

where ω_{ni} and ξ_i respectively are the natural frequency and the damping ratio of the inner reference model and ζ_i is the position of the additional zero. By choosing the reference model parameters stated in Table 6.9, the discrete time transfer function is

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

Note that the approximated position dynamics model in (6.8) does not include the attitude closed-loop system because this dynamic is much faster than the translational one.

Furthermore, 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 structure of the outer reference model can be defined. As in the attitude control loop, the outer controller transfer function is a simple proportional gain, $C_o(s) = k_{P_o}$, and the outer model is an integrator that takes the speed as input and returns the horizontal displacement as output. Therefore, the to be controlled system is given by

$$P_o(s) = F_i(s)\cdot\frac{1}{s} = \frac{gk_{P_i}s + gk_{I_i}}{(1 + gk_{D_i})s^3 + gk_{P_i}s^2 + gk_{I_i}s}\,,$$

thus, the following closed-loop transfer function can be derived:

$$F_o(s) = \frac{C_o(s)P_o(s)}{1 + C_o(s)P_o(s)} =$$

$$= \frac{gk_{P_i}k_{P_o}s + gk_{I_i}k_{P_o}}{(1 + gk_{D_i})s^3 + gk_{P_i}s^2 + g(k_{I_i} + k_{P_i}k_{P_o})s + k_{I_i}k_{P_o}} \, .$$

Unlike the inner loop, no additional poles or zeros are required. Therefore, the resulting continuous time reference model is of the form

$$M_o(s) = \frac{\omega_{no}^2}{s^2 + 2\xi_o\omega_{no}s + \omega_{no}^2} \, ,$$

where ω_{no} and ξ_o respectively are the natural frequency and the damping ratio of the outer reference model. The values of the parameters ω_{no} and ξ_o reported in Table 6.9 yield the following discrete time transfer function:

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

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.

All the characteristics of the inner and outer reference models are summarized in Table 6.9. The desired bandwidths reported in the table has been chosen so as maximise the matching problem between reference model and closed-loop behaviour of the real plant. As will be explained in Subsection 6.3.4, if simulated performance maximisation had been pursued, real-world stability would not be achieved.

	ω [rad/s]	ξ	Added poles/zeros	Weighting function (Low-pass filter)
Inner loop	**1.3**	**0.7**	Zero in **1.8** rad/s	-
Outer loop	**1.1**	**0.6**	-	$f_c = 80$ rad/s

Table 6.9: Inner and outer VRFT reference models.

6.3.3 Simulation results

The controller tunings obtained with the VRFT algorithm have been tested in a simulation environment, similarly to what was done in the attitude controller design. To do this, an outer plant model that transform pitch angle into linear speed has been defined. So, the performance of the controller has been tested before executing the in-flight tests, thus saving time. Nevertheless, due to more or less significant approximations of the plant model, good simulation results guarantee neither good performance of the real system nor closed-loop stability. For this reason, some experimental iterations could be required to define the reference models yielding controllers that provide good performance.

Plant model

As already stated, the knowledge of some model information is essential to correctly define the reference model. At the same time, the more accurate the model is, the faster the tuning process will be. Indeed, with a detailed model, more reliance can be placed on simulation results, thus minimising experimental testing. Nevertheless, the lack of a plant model does not jeopardise the possibility to tune the controller, since the VRFT is a model-free method by definition.

Unlike the attitude control loop, no previously identified position dynamics model was available. In particular, it should be defined a dynamic relationship between the pitch angle $\vartheta(t)$ and the linear speed $v_x(t)$. 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. More specifically, by linearising the equation $m\dot{v}_x(t) = -mgtg\vartheta(t)$ for $\vartheta = 0$, the horizontal speed and position can be defined as

$$v_x(t) = -\int g\,\vartheta(\tau)\,d\tau \qquad \text{and} \qquad x(t) = -\iint g\,\vartheta(\tau)\,d\tau\,.$$

Hence, the discrete time transfer function taking the pitch angle as input and returning the linear speed as output is the following:

$$P_{\vartheta\to v_x}(z) = -g\,\frac{T_s}{z-1}\,.$$

As schematically shown in Figure 6.43, to obtain the position a second integrator block is needed. Obviously, such a defined 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

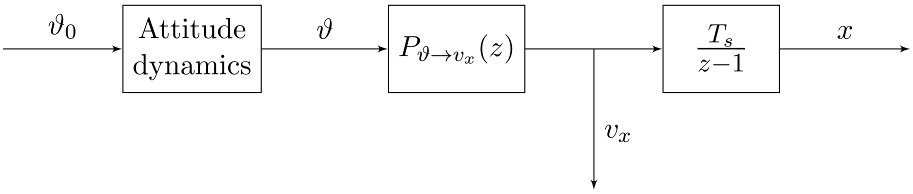


Figure 6.43: Longitudinal position dynamics model.

in the dynamic equation that becomes more and more important moving away from hovering conditions. These approximations result in a significantly less damped dynamic response of the simulated closed-loop system than that of the real one, as shown in Figure 6.44. Furthermore, the validity of the model is restricted to the vicinity of the hovering conditions also due to the linearisation of the translational dynamic equation.

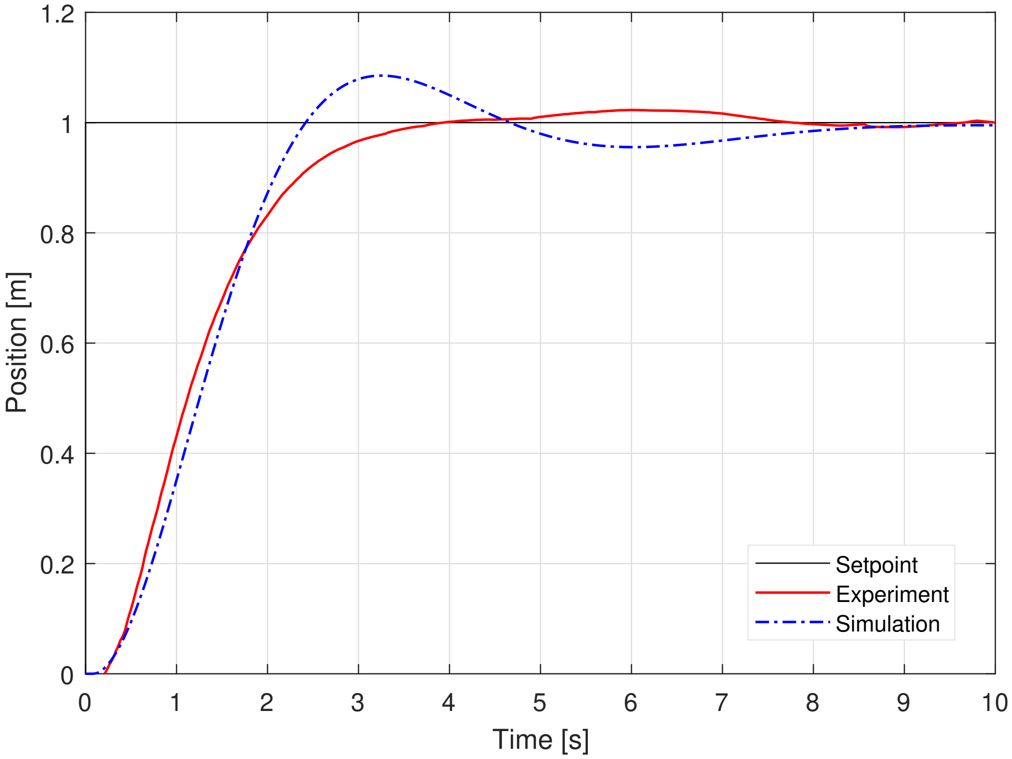


Figure 6.44: Comparison between simulated and real step response.

Inner and outer loop simulation results

The results of the speed control loop as well as the position control loop will be shown below both in frequency domain and time domain. All the results that will follow have been obtained by tuning the controller with the VRFT algorithm and imposing as reference models those described in Subsection 6.3.2.

The control architecture is identical to that implemented for the attitude control loop and it has been already described in Chapter 3. Just to 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.

Bode diagram and step response. Figure 6.45 and Figure 6.46 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 6.47 and Figure 6.48), the simulated system with the VRFT tuned controller almost perfectly match the desired closed-loop behaviour.

By looking at the Bode diagrams, it can be seen a change in the slope of the magnitude at approximately 40 rad/s, which results in a significant discrepancy between reference model and simulated system. This dynamics, not included in the reference model, is related to the attitude loop that is nested in the speed control loop. Although the VRFT algorithm has proved to be not so affected by such discrepancy, better performance has been achieved by considering a weighting function filtering over 80 rad/s.

Controller parameter values

To conclude, the controller parameters associated to the inner PID and the outer proportional gain are summarized in Table 6.10.

	k_{FF_i}	k_{P_i}	k_{I_i}	k_{D_i}	k_{P_o}
VRFT	**0**	**0.1806**	**0.05906**	**0.02172**	**0.77**

Table 6.10: Inner and outer controller parameters considering the VRFT algorithm.

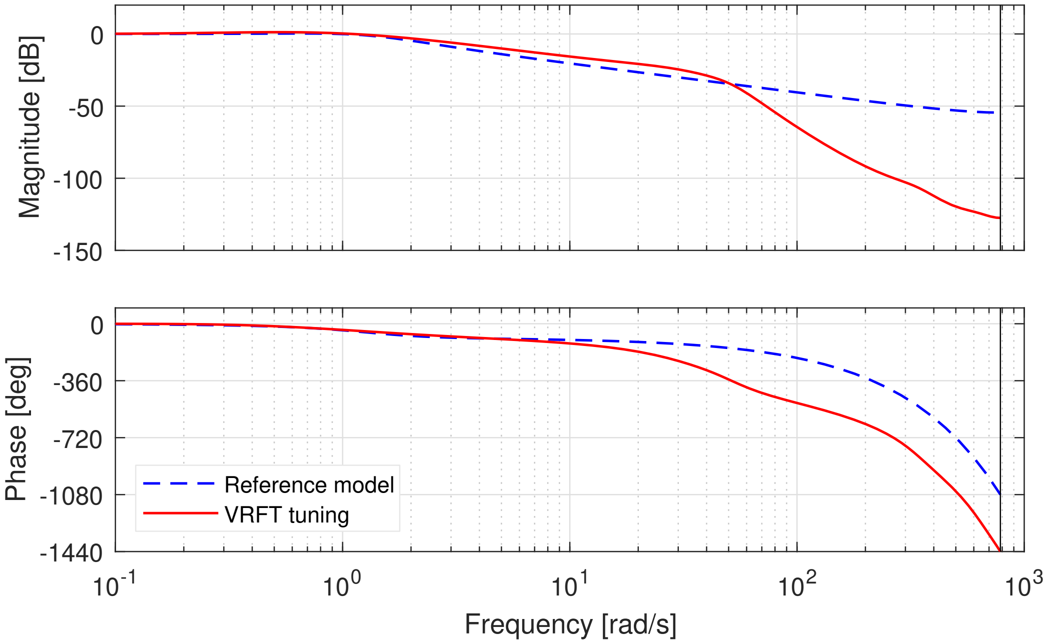


Figure 6.45: Comparison of the position inner loop Bode diagram considering the VRFT tuning (simulation).

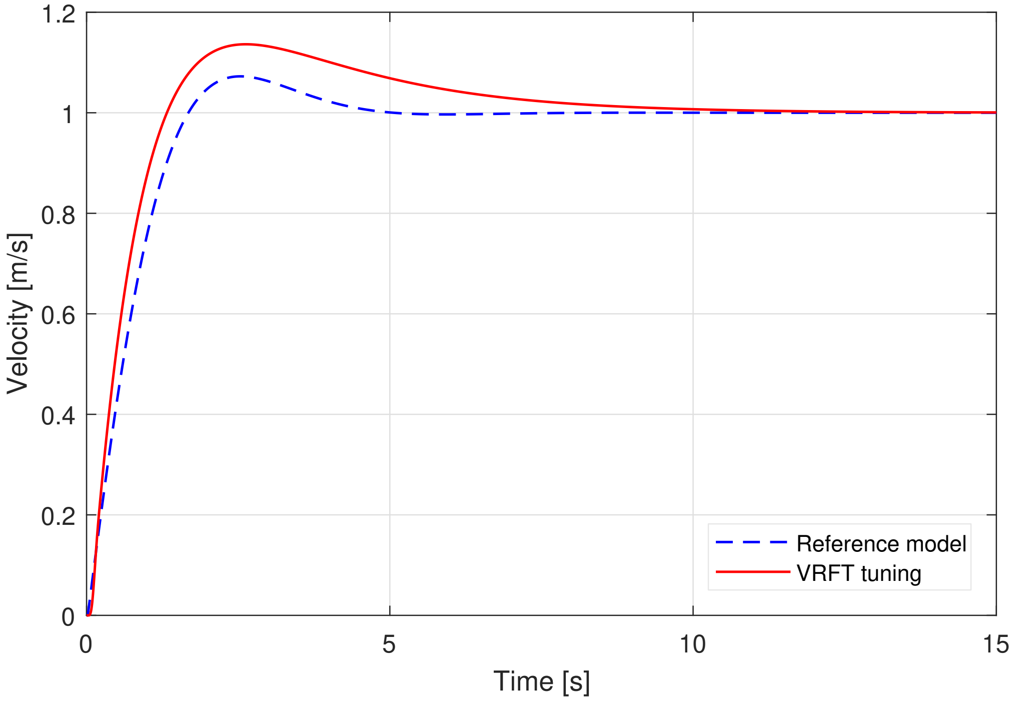


Figure 6.46: Comparison of the position inner loop step response considering the VRFT tuning (simulation).

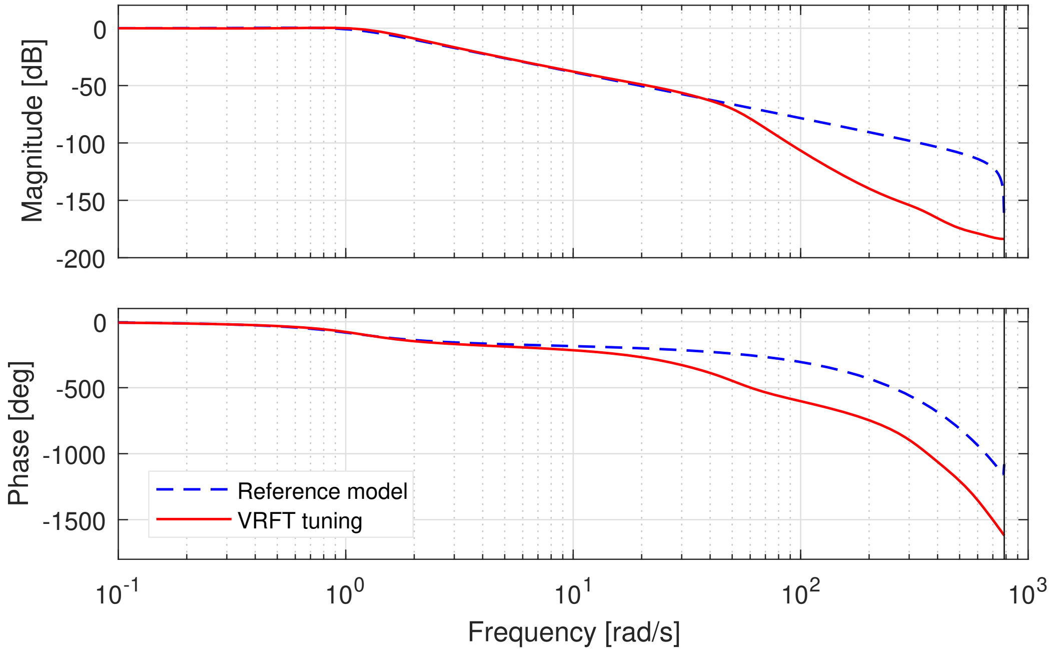


Figure 6.47: Comparison of the position outer loop Bode diagram considering the VRFT tuning (simulation).

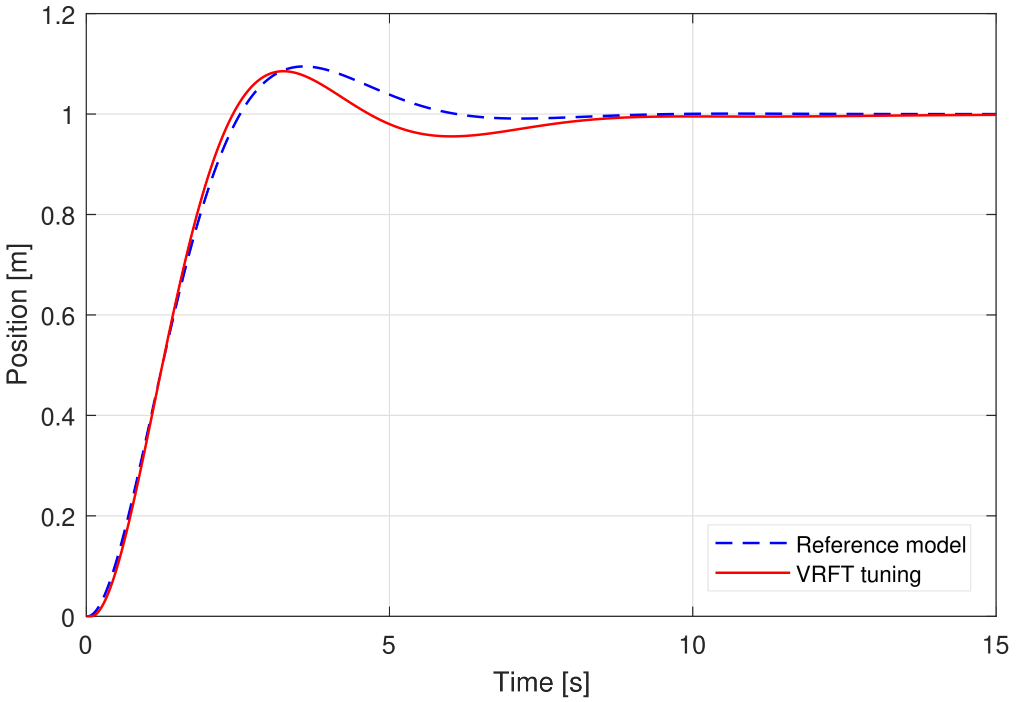


Figure 6.48: Comparison of the position outer loop step response considering the VRFT tuning (simulation).

6.3.4 Experimental results

As already stated in Subsection 6.3.3, the adoption of a very basic model of the to be controlled system 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. In this case, the simulated results have been used only as a starting point for the controller design, while the final tuning has been obtained by means of experimental tests.

Before performing the actual experimental test, closed-loop stability must be checked. To do this, the drone is manually controlled and kept close to the ground to prevent damages if instability occurs. Once stability has been assessed, the ground station can take control of the quadrotor, and the setpoint tracking test, which consists of a sequence of constant demanded positions, can be started.

Setpoint tracking test

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 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 6.49 shows the complete setpoint tracking test obtained by 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 6.50 shows a zoomed-in view of a single step.

As can be seen, the rise time as well as the settling time of the real system and of the reference model are quite similar. On the other hand, the real system results to be more damped than the desired closed-loop model and, as a consequence, more damped than the expected behaviour from the simulations. Indeed, as shown before, the simulated step response is very similar to that of the reference model. The observed discrepancy between real and simulated closed-loop behaviour is the results of neglecting aerodynamic effects in the simulation environment.

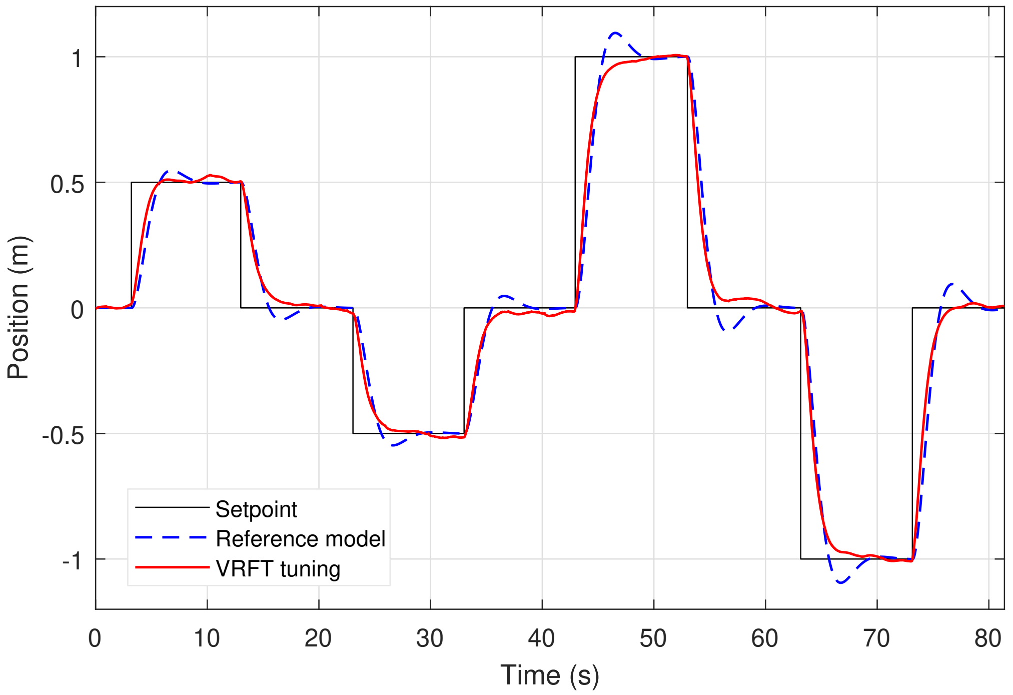


Figure 6.49: Position setpoint tracking with VRFT tuning and ideal reference model response (experiment).

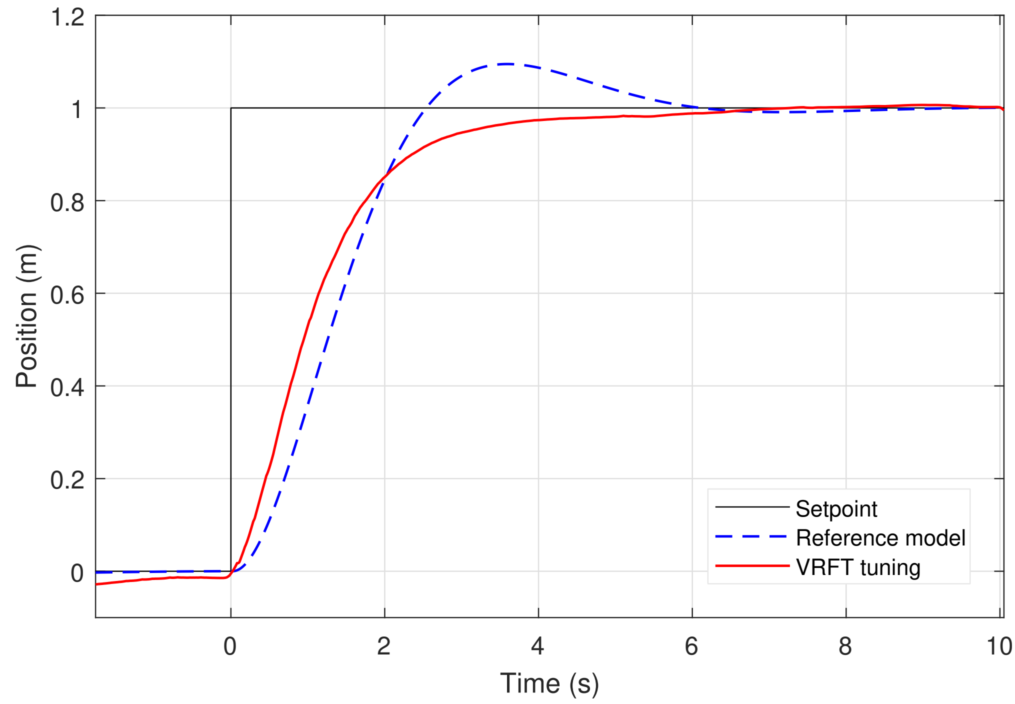


Figure 6.50: Position setpoint tracking (1 m step) with VRFT tuning and ideal reference model response (experiment).

Controller parameter values

Table 6.11 summarises all the controller tunings tested on the real plant. The first designed controller (third row of the table) has been obtained with the VRFT method by imposing as desired bandwidths 1.6 rad/s and 1.5 rad/s on the inner and outer reference models respectively. This tuning results in a stable closed-loop behaviour in the simulation environment, but turned out to be destabilising when applied to the real plant.

Therefore, it has been necessary to relax the reference models, thus, as a second attempt, it has been imposed desired bandwidths equal to 1.3 rad/s and 1 rad/s. These reference models yielded smaller controller parameter values (first row of the table) and thus a slower dynamic response, but ensuring the closed-loop stability.

Finally, an intermediate tuning was carried out in order to make the system faster without compromising the stability (second row of the table). Note that all the simulated and experimental results discussed so far have been obtained with these latter controller parameters. The reference models used to obtain this tuning are described in Subsection 6.3.2. Just to recall, the inner and outer models has a desired bandwidth equal to 1.3 rad/s and 1.1 rad/s respectively. The damping ratio as well as the additional zero and the weighting function are the same for all the three tunings presented here and are shown in Table 6.9.

As for the attitude VRFT controller tuning, an instrumental variable has been adopted to deal with noisy data. Specifically, it has been built through the identification of an ARX(5,5) model for both the inner and the outer loops.

k_{FF_i}	k_{P_i}	k_{I_i}	k_{D_i}	k_{P_o}	**Stability**	
					Sim.	Exp.
0	0.1213	0.08951	0	0.68	✓	✓
0	**0.1806**	**0.05906**	**0.02172**	**0.77**	✓	✓
0	0.2583	0.1231	0.03564	0.96	✓	✗

Table 6.11: Optimal speed and position controller parameters for three different tunings.

Chapter 7

Conclusions and future developments

Below is a brief summary of the purpose of this thesis, the observations made and the results obtained. Furthermore, the aspects that are considered worthy of further study and possible future developments are discussed.

Thesis overview and conclusions

The main purpose of this thesis was to exploit data-driven approaches to tune the PID controllers gains of a cascade attitude and position flight control system, as well as assess their applicability to Micro Aerial Vehicles.

After a detailed classification of these methods, the three most promising algorithms have been presented. From this preliminary analysis, they emerge as a valuable approach to tune the UAV controllers. Indeed, these methods require 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 also computationally efficient, they allow a fast re-tuning of the controllers in case of plant performance reduction or operating conditions change.

The three data-driven algorithms selected and exploited in this thesis are the VRFT, the CbT and the controller unfalsification methods. The VRFT has been an obvious choice, both because it was already extended in [14] to tune a cascade pitch controller, and because of the excellent performance showed in previous works, *e.g.*, [10, 11]. In order to overcome the limitations of the VRFT method in dealing with data heavily affected by noise, the CbT has been considered as a wise alternative in view of the possible noisier measurements of the MAV with respect to previously tested UAVs. The noise robustness of the CbT makes it essential in many practical applications and this added value has been deemed worthy to be studied in this thesis. On the other hand, the lack of an accurate plant model leads to the impossibility for both VRFT and CbT methods to guarantee the stability of the closed-loop system, at least before implementing and testing the controllers on the real plant. This drawback could be unacceptable in the case of larger and more expensive multirotor platforms, and, even more so, if manned aircraft are considered. For this reason, a recently developed data-driven algorithm that includes an effective stability constraint, i.e. the controller unfalsification method, has been 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 because of the complexity of performing open-loop experiments to collect the required data. To deal with this problem, some extensions to the standard data-driven algorithms are presented. In particular, these extensions allow the application of data-driven approaches to the classical control system architecture of UAVs and also the exploitation of closed-loop flight test data.

All the experimental activities have been performed with a small-scale fixed-pitch quadrotor developed as part of a previous master thesis [41], eventually proving the adaptability of data-driven approaches also to MAV applications. The high flexibility of the on-board firmware allowed the implementation of two control architectures in order to validate all the data-driven algorithms proposed in this work. Furthermore, the limited size of the quadrotor facilitated the execution of the flight tests in open-loop conditions, paving the way for the position controller tuning.

All the tested data-driven algorithms have been proven successful in tuning a cascade PID flight control system. The experimental tests, as well as the results from numerical simulations, showed that comparable performance can be achieved using data-driven and model-based control tunings. In particular, as expected, the CbT tuning provided slightly lower performance than the VRFT tuning due to the high SNR value of the gathered data. Furthermore, the controller unfalsification method demonstrated to be effective in preventing closed-loop instability of the real plant. Concerning the VRFT tuning obtained by means of closed-loop, in-flight experiments, the behaviour is only slightly more oscillating than that achieved with the open-loop testing performed on the test-bed. Lastly, the VRFT algorithm has been selected, and successfully deployed, for the position controller tuning (outermost loop of the flight control system).

In conclusion, the data-driven PID tunings herein proposed showed excellent tracking and disturbance rejection capabilities and, thus, can represent a worthwhile solution for the fast deployment of high performance attitude and position controllers for UAV applications.

Future developments

Leveraging the promising results of the VRFT tuning obtained with data collected in closed-loop conditions, it is possible to abandon the assumption of an attitude control system based on individual SISO loops (pitch and roll DoFs), and to deploy a MIMO control architecture whose gains can be tuned simultaneously through a suitable excitation of the vehicle dynamics. For this purpose, a flight test is mandatory to gather the data-set for the data-driven method, and the input signals must be designed to excite both pitch and roll dynamics. Obviously, a MIMO attitude controller is useless for practical purposes, because of the almost-perfect decoupling of the quadrotor pitch and roll DoFs, especially in near-hovering conditions. Nevertheless, the design of a MIMO controller for an UAV is an intermediate preparatory step, which should be envisaged before extending data-driven approaches to the design of the MIMO attitude control system of a helicopter. Indeed, the pitch-roll and the heave-yaw DoFs of a helicopter are strongly coupled with each other, making this kind of controller more valuable than individual SISO loops. In particular, two experimental tests could be sufficient to tune the entire attitude control system, including the coupled dynamics controllers: one should excite the pitch-roll dynamics, while the other the heave-yaw dynamics. In principle, it should be possible to derive two MIMO controllers for the inner loop. The first one takes as input variable the pitch and roll rates, while the second one the heave and yaw rates, respectively returning as outputs the longitudinal and lateral cyclic pitches, and the main and tail rotors collective pitches.

The two most important obstacles in the diffusion of data-driven methods in the aeronautical field, namely the execution of closed-loop tuning experiments and the *a-priori* guaranteed stability, have been addressed in this thesis. Therefore, unacceptable safety risks can effectively be avoided in the case of a manned helicopter.

In conclusion, it is advisable to proceed as follows to extend the application of data-driven tuning methods from multirotors to rotorcraft:

- Tune a MIMO controller for the pitch-roll dynamics of a quadrotor (note that the VRFT algorithm (see Algorithm 1) is currently not limited to SISO systems).

- Tune the control system of a small or a full-scale helicopter in hovering conditions by assuming decoupled dynamics. This means tuning the control loop of each DoF independently, by relying on a different input-output data-set for each loop.

- Tune the control system of a small or a full-scale helicopter in hovering conditions by modelling the control structure with two MIMO controllers for the inner loops (pitch-roll and heave-yaw) and one for the outer loop (longitudinal-lateral positions). By doing so, the coupled plant dynamics is taken into account and ideally better performance should be achieved than in the previous case. The hovering helicopter dynamics is still weakly coupled if compared to translational flight conditions.
- Extend the previous studies to the translational flight, repeating the tuning procedure for increasing airspeed values and implementing a gain scheduling approach to cope with the variations in the helicopter dynamics, as well as in the controls cross-couplings, with the airspeed.

Bibliography

[1] 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.

[2] 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.

[3] P. G. Hamel and J. Kaletka. Advances in rotorcraft system identification. *Progress in Aerospace Sciences*, 33(3):259 – 284, 1997.

[4] M. B. Tischler and R. K. Remple. Aircraft and rotorcraft system identification. *AIAA education series*, 2006.

[5] R. Jategaonkar. *Flight vehicle system identification: a time domain methodology*, volume 216. AIAA, Reston, VA, USA, 2006.

[6] 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.

[7] K. J. Åström and T. Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Isa Research Triangle Park, NC, 1995.

[8] 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.

[9] D. Invernizzi, P. Panizza, F. Riccardi, S. Formentin, and M. Lovera. Data-driven attitude control law of a variable-pitch quadrotor: a comparison study. *IFAC*, 44(17):236–241, 2016.

[10] P. Panizza, D. Invernizzi, F. Riccardi, S. Formentin, and M. Lovera. Data-driven attitude control law of a variable-pitch quadrotor. *American Control Conference*, 2016.

[11] T. Chupin. Data-driven attitude control design for multirotor UAVs. Master's thesis, Politecnico di Milano, 2017.

[12] 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.

[13] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin. Iterative feedback tuning: theory and applications. *IEEE Control Systems*, 18(4):26–41, Aug 1998.

[14] 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.

[15] 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.

[16] G. Battistelli, D. Mari, D. Selvi, and P. Tesi. Direct control design via controller unfalsification. *International Journal of Robust and Nonlinear Control*, 2017. rnc.3778.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010.

[23] 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.

[24] M. La Civita. *Integrated Modeling and Robust Control for Full-Envelope Flight of Robotic Helicopters.* PhD thesis, Carnegie Mellon University, 2003.

[25] 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.

[26] 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.

[27] E. Capello, A. Scola, G. Guglieri, and F. Quagliotti. Mini quadrotor UAV: design and experiment. *Journal of Aerospace Engineering*, 25(4):559–573, 2012.

[28] R. Lozano, P. Castillo, and A. Dzul. Modeling and control of mini-flying machines, 2005.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

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

[36] 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.

[37] A. Lecchini, M. Campi, and S. M. Savaresi. Virtual reference feedback tuning for two degree of freedom controllers. 16:355 – 371, June 2002.

[38] L. Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[39] Ente Nazionale Aviazione Civile (ENAC). *Remotely Piloted Aerial Vehicles Regulation - Issue no. 2, Revision 3.* [Updated March 2017; accessed: 27 Nov 2017]. Available from: https://www.enac.gov.it/la_normativa/normativa_enac/.

[40] Federal Aviation Authority (FAA). *Aircraft Registry - Aircraft Registration: Unmanned Aircraft* [Internet]. [Updated July 2017; accessed: 27 Nov 2017]. Available from: https://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/UA/.

[41] D. Chevallard. Design, identification and control of a micro aerial vehicle. Master's thesis, Politecnico di Milano, 2017.

[42] M. Giurato. Design, integration and control of a multirotor UAV platform. Master's thesis, Politecnico di Milano, 2015.