CONTRIBUTIONS TO QUANTITATIVE FEEDBACK THEORY DESIGN AND PRELIMINARY APPLICATION TO A VARIABLE-PITCH QUADCOPTER

Arnold Pretorius
August 4, 2023
PRTARN001

A thesis submitted in fulfilment for the
degree of Doctor of Philosophy
in
Electrical Engineering

Department of Electrical Engineering
UNIVERSITY OF CAPE TOWN

Supervisor: Professor Edward Boje

Abstract

This thesis details the mathematical and mechanical modelling and design, state estimation, and preliminary control of a novel variable-pitch quadcopter. The experimental framework is first developed, which includes the physical quadcopter platform, as well as a vision-based motion capture system. Modelling and system identification methods are applied to the quadcopter low-level subsystems, in order to understand and verify the fundamental dynamics of the system. Following this, a state estimation algorithm, based on the extended Kalman filter, is developed, which fuses camera information from the motion capture system to provide pose estimation of the quadcopter platform. A novel rotor thrust observer scheme is also presented, which enables on-board estimation of the quadcopter rotor thrusts during operation. Preliminary low-level control of the rotor speed and thrust is demonstrated, which is facilitated by the on-board rotor speed and thrust estimates. Finally, a simulated demonstration of the position control scheme is provided, which makes use of the previously modelled subsystems and designed control schemes. The simulation environment includes nonlinearities and noise effects that emulate that of the real-world experimentation, and the variable-pitch quadcopter is shown to perform as expected. This thesis investigates the state of the art of quantitative feedback theory, with a particular focus on reducing feedback controller design conservatism. Starting with the generalised single-input-single-output problem formulation, we introduce a novel means of synthesizing a per-plant closed-loop model specification that caters to the signal and phase limitations of every plant instance in the plant set. This information is then incorporated into a univariate constraint set on the feedback controller element, which is predicated on the existence of a valid, non-empty pre-filter solution space in the arithmethic-complex plane. Using a simple example case, we are able to show that the method is effective in balancing the tracking performance across the entire plant set, subject to the inherent signal limits. In a subsequent contribution, we introduce a new approach to the 2×2 model-error tracking problem that combines a plant-inverting design routine with a novel non plant-inverting method, with the aim of reducing the controller design conservatism. We show that geometric-based existence conditions can be exploited to arrive at a univariate design constraint set on the particular feedback controller element of interest, whilst reducing design conservatism at all pertinent frequencies of interest. This method is shown to substantially outperform traditional plant-inverting 2×2 methods, especially at the gain-phase crossover range. Serving as the main QFT contribution of this thesis, we develop a generalised multi-variable refinement approach to the tracking error problem that is intended to ease the feedback control design at all frequencies. Assuming a valid *a priori* feedback design exists, a feedforward filter is synthesized using optimisation, with the intention of loosening the strictures on a subsequent differential feedback design.

The resulting prototype control solution is then used to provide additional gain and phase information that aids in reducing the design conservatism when applying the triangle inequality. This process can be applied iteratively in order to refine the loop transfer behaviour and reduce the feedback controller gain. The method is shown to surpass current multivariable QFT design routines in specific benchmarking examples in terms of expanding the admissible feedback controller per-frequency solution space, especially in the gain-phase crossover region.

Acknowledgements

I would like to express immense gratitude to my supervisor, who has not only provided invaluable and expert professional guidance, but who has always allowed me the freedom to explore different research avenues throughout my postgraduate studies. Prof, thank you for always making yourself available to me, and for introducing me to the wonderful world of Quantitative Feedback Theory.

I thank the National Research Foundation (NRF) for subsidizing my postgraduate fees and allowing me the opportunity of continuing with this thesis past the nominal 3-year timeframe.

A special thank you goes to Yashren Reddi, who has been undeniably crucial to the advancement of my academic career right from the day I first met him. Yash, I look back fondly on us sharing a little office at the back of the Control Lab and having in-depth conversations that would range from deeply academic to downright asinine! You are the epitome of an all-round engineer and I strive to emulate your ability.

To *Ma en Pa*, thank you for prioritising my education right from the beginning, and providing the best possible foundation for me to build a life that I am truly happy in. Your unwavering support and faith in my ability is astounding and your model work ethic and drive has made me the engineer that I am today.

To my little monkey, Sam, you wont be able to read this for a while, but thank you for giving me fleeting moments to incrementally complete this PhD! More seriously though, your presence in my life is an absolute blessing — I would be an empty shell without you. I look forward to one day teaching you all about control theory.

And finally, to my life partner and best friend. Jamie, thank you for all your support and consideration over the last few years. Your patience and understanding during this (overly) extended postgraduate period has been more than generous and I can honestly say I would not be submitting this document now without you. I am very excited to start on all our many life journeys that were previously put on hold for the PhD.
To the moon and back.

Declaration of Authorship

The work presented in this thesis is my own, conducted under the supervision of Professor Edward Boje. This thesis contains research that has been published or submitted for publication, as shown below. In each case, I was the primary researcher, first author, and wrote all the sections.

Paper 1: Pretorius, A, Boje, E. Design and Modelling of a Quadrotor Helicopter with Variable Pitch Rotors for Aggressive Manoeuvres. *IFAC Proceedings Volumes*, Volume 47, Issue 3, 2014, Pages 12208-12213, ISSN 1474-6670, ISBN 9783902823625,
https://doi.org/10.3182/20140824-6-ZA-1003.01586.
This work been published in IFAC-PapersOnLine (formerly known as IFAC Proceedings Volumes).

Paper 2: Pretorius, A, Boje, E. Shutter delay estimation for a low-cost stand-alone visual tracking system, *IFAC-PapersOnLine*, Volume 50, Issue 1, 2017, Pages 11441-11446, ISSN 2405-8963,
https://doi.org/10.1016/j.ifacol.2017.08.1813.
This work been published in IFAC-PapersOnLine (formerly known as IFAC Proceedings Volumes).

Paper 3: Pretorius, A, Boje, E. Robust plant by plant control design using model-error tracking sets. *International Journal of Robust and Nonlinear Control*. 2019; 29: 3330–3340.
https://doi.org/10.1002/rnc.4553.
This work has been published in the International Journal of Robust and Nonlinear Control.

Paper 4: Pretorius, A, Boje, E. A complementary quantitative feedback theory solution to the 2×2 tracking error problem. *International Journal of Robust and Nonlinear Control*. 2020; 30: 6569–6584.
https://doi.org/10.1002/rnc.5120.
This work been published in the International Journal of Robust and Nonlinear Control.

Paper 5: Pretorius, A, Boje, E. A refinement approach to the multivariable tracking error problem. *International Journal of Robust and Nonlinear Control*. 2022; 32(12): 7016- 7038,
https://doi.org/10.1002/rnc.6183 .
This work has been published in the International Journal of Robust and Nonlinear Control.

Declaration on Inclusion of Publications

I confirm that I have been granted permission by the University of Cape Town's Doctoral Degrees Board to include the following publication(s) in my PhD thesis, and where co-authorships are involved, my co-authors have agreed that I may include the publication(s):

- Pretorius, A, Boje, E. Design and Modelling of a Quadrotor Helicopter with Variable Pitch Rotors for Aggressive Manoeuvres. *IFAC Proceedings Volumes*, Volume 47, Issue 3, 2014, Pages 12208-12213, ISSN 1474-6670, ISBN 9783902823625, `https://doi.org/10.3182/20140824-6-ZA-1003.01586`.
- Pretorius, A, Boje, E. Shutter delay estimation for a low-cost stand-alone visual tracking system, *IFAC-PapersOnLine*, Volume 50, Issue 1, 2017, Pages 11441-11446, ISSN 2405-8963, `https://doi.org/10.1016/j.ifacol.2017.08.1813`.
- Pretorius, A, Boje, E. Robust plant by plant control design using model-error tracking sets. *International Journal of Robust and Nonlinear Control*. 2019; 29: 3330–3340. `https://doi.org/10.1002/rnc.4553`.
- Pretorius, A, Boje, E. A complementary quantitative feedback theory solution to the 2×2 tracking error problem. *International Journal of Robust and Nonlinear Control*. 2020; 30: 6569–6584. `https://doi.org/10.1002/rnc.5120`.
- Pretorius, A, Boje, E. A refinement approach to the multivariable tracking error problem. *International Journal of Robust and Nonlinear Control*. 2022; 32(12): 7016- 7038, `https://doi.org/10.1002/rnc.6183` .

SIGNATURE:

DATE: August 4, 2023

STUDENT NAME: Arnold Pretorius

STUDENT NUMBER: PRTARN001

Plagiarism Declaration

I, **Arnold Pretorius***, know the meaning of plagiarism and declare that all of the work in the document, save for that which is properly acknowledged, is my own.*

SIGNATURE: **DATE:** August 4, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The contributions of this thesis to control engineering using QFT were motivated by the potential use of these tools for practical control design of a quadcopter with innovative, variable-pitch rotors. The thesis details the mathematical and mechanical modelling and design, state estimation, and low-level control of the variable-pitch quadcopter shown in Figure 1.1. This thesis excites a wide range of engineering disciplines, including control engineering, robotics, software design, aeronautics, and electromechanical engineering.

This thesis makes incremental contributes to the field of quantitative feedback theory (QFT) bound generation methods, with predominant focus on the multivariable reference tracking problem. The developments use engineering and mathematical insights to inform novel bound generation methods that will overcome some of the shortcomings and pitfalls that exist in the current QFT methodology.

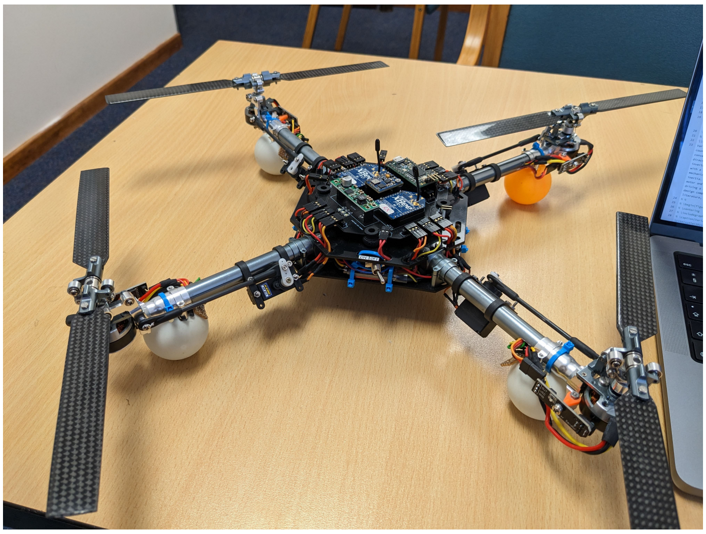


Figure 1.1: Fully assembled variable-pitch quadcopter.

1.1 Motivation

1.1.1 Variable-pitch quadcopter

Quadrotor helicopters, also referred to in short as quadcopters, are interesting and versatile robotic platforms that have captured significant interest from hobbyists, end-users and robotics research groups across the world. The widespread appeal of a four-rotor helicopter is derived from the relatively simple mechanical setup (and by extension, dynamic model), relative to that of the mechanically-complex, conventional helicopter system. The vertical take-off and landing (VTOL) capability of quadcopters is also appealing for conducting research in small or indoor environments.

One of the predominant control challenges for unmanned aerial vehicles (UAVs) centres on performing aggressive autonomous manoeuvres away from near-hover conditions. Several groups have already succeeded in achieving fairly acrobatic flight regimes on physical quadrotors, such as the work in [7, 8, 9]. Generic proportional-integral-derivative (PID) control schemes (and their associated subsets) are commonly designed using quasi-decoupled attitude plant models, based on near-hover conditions, but the resulting control designs are often only valid when the vehicle is actuated over a single degree of freedom, or using sufficiently small angular changes. This has the effect of limiting the flight envelope of the aerial vehicle, as large deviations from hover can result in degraded tracking performance, or even instability.

With the fairly recent introduction of variable-pitch quadcopters, these robots can now do impressive manoeuvres without being restricted by the bandwidth-limiting inertia of the motor-rotor pair [10]. The pitch actuation of the rotor blades addresses the issue of the relatively slow motor time constant for acrobatic flight, but there is potential for improved flight performance using a more refined and considered mechanical and control design methodology that captures and accounts for the quadcopter dynamics at all operating points of interest.

1.1.2 Quantitative feedback theory

Quantitative feedback theory is a powerful and pragmatic tool to address realistic control problems that possess non-negligible parameter uncertainty [11]. The design procedure consists of selecting a feedback controller and pre-filter (or feedforward filter) that satisfy frequency-domain specifications enforcing appropriate performance and stability properties of the closed-loop system [12]. The model matching (or model-error) problem formulation is a more recent tracking-based QFT approach that imposes gain and phase requirements on the closed-loop reference to output transfer behaviour. This approach has been shown to surpass the traditional above/below bounding methodology in terms of time-domain tracking precision [13] and design conservatism [14], and lays the framework for most modern QFT bound generation routines. Model-error specifications are also closer to H_∞ bounds even though the underlying plant uncertainty is not norm bounded.

Following the advent of bounding the tracking transfer behaviour in both gain and phase [15], research interest has circulated around addressing several key challenges in modern QFT bound generation routines, namely:

(i) consideration of plant signal limitations, such as plant input and input-rate saturation,

(ii) decoupling the pre-filter (or feedforward filter) and feedback controller design constraints and apportioning bandwidth appropriately between the two controllers,

(iii) reducing design conservatism of the feedback controller and pre-filter as a result of multivariable loop coupling.

Challenge (i) is commonly addressed using retroactive control measures [16], but is seldom included in a formal design [17]. More generally, the designer attempts to minimise gain at every design frequency and reduce resonant modes in the closed-loop frequency response magnitude, with the implicit intention of suppressing large control action. Recursive design is used to palliate any signal violations from the first-pass design.

Successfully facilitating (ii) is often accomplished via appropriate use of existence conditions. Specifically, existence conditions for the pre-filter can be enforced on the feedback control design, with the intention of removing a degree of freedom from the design constraints [4]. The coupled nature of the two-degree-of-freedom control scheme means that the selection of the feedback controller will constrain the resulting design region of the pre-filter in terms of location, shape, and size. Sub-optimal design of the feedback controller (in terms of overfulfilling the design inequalities) may result in an unnecessarily large pre-filter solution space, whereas aggressive design of the feedback controller may leave insufficient design flexibility in the pre-filter solution space. Simply put, the division of design flexibility between the two controllers is essential to finding and completing a pragmatic control design. Alternative approaches to (ii) exist, such as making use of *a priori* information in order to synthesize the pre-filter [18] prior to design of the feedback controller. While this does simplify the design problem considerably, selection of the pre-filter is generally non-trivial and a sub-optimal selection of the pre-filter will negatively affect the feedback controller solution space.

Similar to (ii), the challenge in (iii) is brought about from the multivariate design constraints that arise when extending the SISO magnitude bounding approach to multi-input-multi-output (MIMO) systems. Following the multivariable QFT framework, the triangle inequality is applied to a (typically) plant-inverting constraint set (see Figure 1.2), which replaces unknown terms with their worst-case gain and phase tolerances [19]. This facilitates a univariate constraint set that enables determinate boundary generation, but the conservatism brought on by the worst-case assumptions can result in large gain-phase feedback controller solutions, especially at the mid to high frequencies where the loop gain is not large [20]. Worse yet, if the system has element-wise non-minimum phase behaviour, the conservative constraint set may enforce design regions that are incompatible with the stability design regions. This type of problem also extends to the pre-filter or feedforward filter design, as the

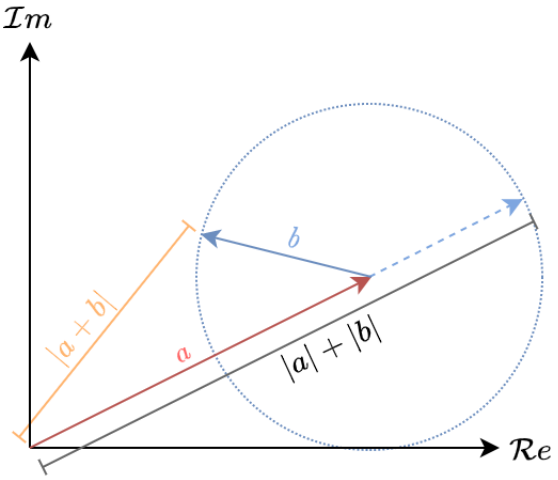


Figure 1.2: Visual representation of the triangle inequality in the arithmetic-complex plane. The corresponding inequality is $|a+b| \leq |a|+|b|$.The overbounding is most dramatic when vector a and b are opposite in direction.

stay-in design regions require strict gain-phase adherence to complete the feedback control design — something that is exacerbated by design conservatism.

The conventional QFT framework's use of bounding frequency-domain information with appropriate magnitude specifications (above/below or model-error bounding) is not innately well-suited to solving the aforementioned challenges of (i)-(iii). However, the resulting constraint-based design inequalities are structured in a way that can be directly applied to numerical optimisation methods. This, coupled with leveraging geometrically-motivated existence conditions of the controller elements, has the potential to alleviate said shortcomings and break new ground in the QFT research sphere.

This thesis exploits the mathematical and geometric structure of the tracking design constraints in order to offer updated solutions to the key challenges of (i)-(iii). This is assisted by constraint-based numerical optimisation, with the intention of circumventing previous pitfalls in the QFT bound generation methodology. As is the nature of multivariable QFT design methodologies, the proposed solutions, while able to reduce design conservatism relative to the current literature, cannot in general remove all design conservatism as part of the SISO equivalent loop-by-loop control design procedure.

1.1.3 Application of QFT to a variable-pitch quadcopter

The variable-pitch quadcopter is a versatile robotic platform that has the potential for real-world application, whilst also serving as a challenging multivariable test platform for the development of applied QFT-based control principles. This is evidenced by the nonlinear, coupled attitude dynamics of the quadcopter away from hover (see Section 3.4.2), as well as its direct effect on the translational dynamics (see Section 3.4.1). As with any real-world platform, plant uncertainty additionally contributes to the design problem, which is innately suited to the QFT paradigm. Design and implementation insights from the integration of the two disciplines can then be abstracted to other multivariable control engineering problems.

1.2 Aim and objectives

The aim of this project was to provide the rigorous preliminary foundation of a variable-pitch quadcopter and motion capture system, the combination of which can be extended as part of future research endeavours to include full flight tests.

The project objectives are summarised below.

(i) The first objective was to develop a novel experimental platform, which takes the form of a variable-pitch quadcopter. This included mechanically designing and constructing the robot, as well as implementing the various electromechanical elements on the platform. Control of the system also entails writing software that can communicate and interact with the quadcopter sub-systems.

(ii) The second objective was to devise and implement a system that is capable of estimating the quadcopter pose (location and orientation) with sufficient precision and robustness, as high-fidelity state information is necessary for robust feedback control.

(iii) The final objective was to investigate current quantitative feedback theory methods pertaining to SISO and MIMO systems, with the intention of exploring improved design methodologies. Insights from the developed design routines can then be applied to the variable-pitch quadcopter as part of future work.

1.3 Contributions to knowledge in the field

1.3.1 Variable-pitch quadcopter

The variable-pitch quadcopter contributes to knowledge in the field based on the mechanical and mathematical modelling and design, as well as the novel state estimation and sensor fusion algorithms employed. By extension, the documented electromechanical, software, and theoretical insights serve to advance certain aspects of the field of robotics. Further detail is provided below.

Mechanical modelling and design of a novel variable-pitch quadcopter (Chapter 2)

A novel variable-pitch quadcopter was designed and built, with the intention of surpassing the fundamental limitations of traditional speed-controlled quadcopters. The physical platform went through various design iterations [21], and the current iteration combines the benefits of traditional fixed-pitch quadcopters, with variable-pitch quadcopters. The integration of the mechanical design and control design choices related to the state operating conditions remove nonlinearities and cross-coupling effects from the generic UAV model, thereby resulting in a simplified attitude plant model, regardless of the flight operating region. Additionally, the electromechanical configuration of the quadcopter enables thrust estimation and control of each rotor, which facilitates high-bandwidth loop closure at the rotational acceleration level — where disturbances act on the quadcopter.

Mathematical modelling and system identification of a variable-pitch quadcopter (Chapter 3, Chapter 5)

A high-fidelity model of the variable-pitch quadcopter is constructed [21], which breaks the full nonlinear system into modular sub-systems, and extracts the fundamental behaviour of the system. The rotor thrust is modelled from first principles using a combination of blade element theory and momentum theory and the mathematical model is verified using experimental results.

The low-level rotor speed and thrust characteristics were captured using experimental methods and analysed using system identification techniques. Following this, the linearised, uncertain plant sets were developed, which captured the parameter variations for the flight regime of interest. This facilitated QFT-based feedback control design of the rotor speed and thrust. Information from the closed-loop rotor speed and thrust subsystems was then incorporated into the quadcopter attitude plant set, which enabled subsequent design of an attitude and position controller. A simulation of the quadcopter position control system was provided, and serves to demonstrate the holistic quadcopter model validity.

Development of a set of state estimators that are able to collectively estimate the variable-pitch quadcopter pose and thrust (Chapter 4)

A low-cost, visual-based motion capture system was designed, built and tested. The cameras operate in the infra-red spectrum, and is capable of estimating pose (translation and orientation) of a rigid-body vehicle [22]. This cost-effective solution makes use of a quasi-continuous extended Kalman filter and achieves sub-centimetre position accuracy. The generic hardware and software structure allows for any number of cameras to be added to the system.

Additionally, an on-board explicit complementary filter was implemented that fuses off-board attitude estimation from the camera system with that of on-board sensor information from an inertial measurement unit (IMU). In this way, the low-bandwidth information from the camera system is combined with high-bandwidth gyroscope information, which results in an improved quasi-realtime estimate of the quadcopter attitude and attitude rate.

Finally, on-board, dynamic rotor thrust estimation was achieved using real-time measurements of the motor currents and speeds. The thrust estimator was calibrated using a custom-built thrust test rig.

1.3.2 Quantitative feedback theory

The QFT thesis contribution can be summarised as offering novel design routines to three classes of problem, namely, single-input-single-output (SISO) systems, two-input-two-output (TITO) systems, and multi-input-multi-output (MIMO) systems. Each successive contribution takes a progressive next step of elevating the class of problem and makes use of knowledge and design insights gained from previous work. This is especially apparent in the latter contribution, where the feedforward bound generation methodology is a multivariable generalisation of the 2×2 approach used in the TITO design formulation. Further detail is provided below.

Development of a novel multi-model tracking error design routine for single-input-single-output systems (Section 6.4)

Per-plant reference models are synthesized based on the inherent plant signal limitations using constrained optimisation. This places realistic expectations on each closed-loop plant case, as opposed to arbitrarily bounding the frequency responses of every plant case based on a single model specification. The existence of a viable pre-filter is predicated upon satisfying a set of constraints on the feedback controller element and this design process implicitly maximises the design regions for the feedback controller, albeit at the cost of greater stricture on the design regions of the pre-filter. A level set of tracking tolerances is used to inform the designer of relevant design trade-offs between controller complexity and tracking precision. See Section 6.4 for more detail.

A tracking error bound generation routine for the class of two-input-two-output systems that overcomes the issues persistent in previous QFT-based design routines that rely on plant-inverting splittings (Section 6.5)

Supplementing conventional bound generation routines with a non plant-inverting approach has the potential to substantially reduce the level of design conservatism in the gain-phase crossover region (where the cost of feedback is often highest). This has the effect of easing the control design challenge and obtaining a significantly lower gain/phase feedback control solution. See Section 6.5 for more detail.

Development of a novel refinement approach for the generalised multivariable model tracking error problem (Section 6.6)

The proposed method relies on reducing the conservatism in the tracking error design constraint set by incorporating gain and phase information from a previous design iteration. Additionally, the incremental tracking allowance is scaled according to a division of tolerances, which enables the designer to update the feedback controller elements to improve the overall design through reduced conservatism. The recursive nature of this design routine is exploited to make iterative refinements to the feedback controller, whilst enabling the designer to balance gain and phase allowance across the various channels. See Section 6.6 for more detail.

A bound generation routine that applies to generalised nondiagonal, multivariable feedforward filters (Section 6.6)

The design region synthesis relies on using appropriate existence conditions and allows for element-by-element design of the feedforward filter without induced design conservatism. Successful design of a specific row of the controller gives rise to the solution space of the subsequent row, based on meeting the imposed existence condition. This approach only requires a valid feedback controller design to proceed, and substantially reduces the burden on the designer in terms of meeting the frequency response requirements of the closed-loop system. Note that the two-input-two-output precursor to this approach is first introduced in Section 6.5. See Section 6.6 for more detail.

1.4 Scope and limitations

The full development of the experimental platform used in this research project took place at the University of Cape Town, without assistance from external parties or other students/technicians. While not directly part of the project scope, the various hardware and equipment has been purchased with a very limited budget (hence often carrying the tag of "low-cost").

The control design techniques contained in Section 6.4-6.6 have a sole focus on QFT-based design and synthesis routines — other control design methods are not considered or included as comparison.

1.5 Thesis structure and outline

The thesis outline is as follows:

The experimental framework, which includes the electromechanical design of a variable-pitch quadcopter platform and vision-based motion capture system, is developed in **Chapter 2**. A summary of the ground station, simulation environment and thrust test rig is also provided.

Chapter 3 presents the stringent mathematical modelling applied to the experimental platform. The focus in this chapter is to isolate each fundamental sub-system of the full nonlinear system, in order to gain insight into the underlying dynamics. Low-level system identification of the quadcopter is also presented.

Derivations and justifications of the chosen state estimation schemes are provided in **Chapter 4** . This includes the centralised motion capture system, on-board attitude refinement, and rotor thrust estimation. The methodology and performance results of the state estimation sub-systems are also presented.

Chapter 5 provides preliminary low-level control design, implementation, and analysis of the variable-pitch quadcopter system. The control design includes the rotor speed and thrust control, whereas the system analysis provides insights into the control authority allocation and mapping approaches. Attitude plant sets are also determined, after considering the aforementioned low-level loop closure, which establishes the framework for future applied work on the quadcopter control design. A simulation of the closed-loop position control system is provided, using the developed simulation environment. This contains brief explanations of prototype attitude and position controller designs, along with a demonstration of a cross-axis quadcopter manoeuvre in the inertial-frame.

Chapter 6 considers generalised QFT reference tracking design methodologies. The chapter includes a brief summary of the development of frequency-domain methods and then presents integrative material that relates to the formative work in the field. Sections 6.4-6.6 include three self-contained journal publications that each comprise: an introduction and literature review; a problem statement and proposed method; worked example(s) demonstrating the particular method; and a summative conclusion. Minor adjustments have made been made to the publications for sake of consistent reading, but otherwise remain unchanged.

The conclusion of this thesis is drawn in **Chapter 7**, and recommendations on future research are offered.

1.6 Data availability

The functions and algorithms that form part of this thesis dissertation are provided in this GitHub repository. Content-specific links are also provided throughout the thesis document, where appropriate.

Chapter 2

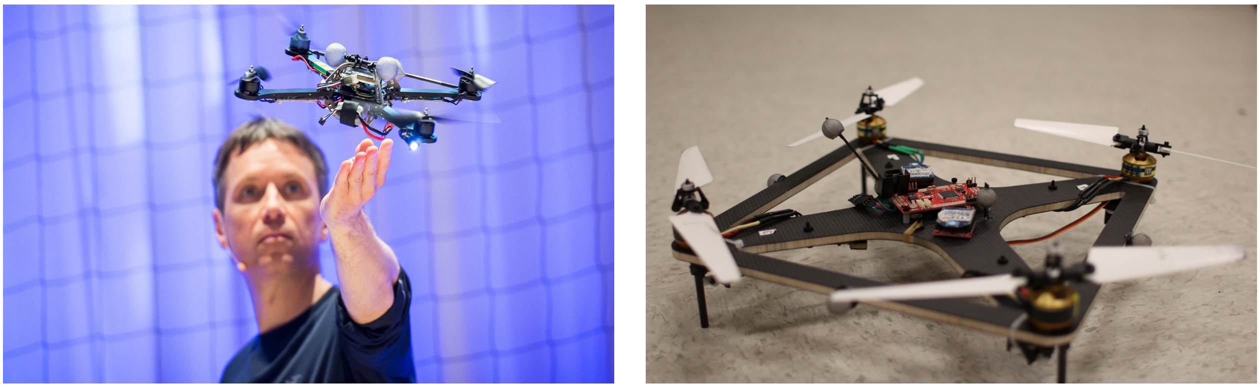
Experimental framework

This chapter is based, in part, on the following published paper:
Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres [21].

Section 2.1 provides a review of the literature relevant to speed-controlled and variable-pitch quadcopters. Section 2.2 outlines the mechanical and instrumentation elements that make up the designed and constructed variable-pitch quadcopter platform. Section 2.3 details the hardware and functional components of the designed and implemented vision-based motion capture system. Section 2.4 provides information about the ground station processes, functions, and interaction with the aforementioned motion capture system and quadcopter platform. A high-level description of the simulation environment is supplied in Section 2.5. Section 2.6 contains information pertaining to the thrust test rig, which is used for rotor system identification, calibration, and testing. Finally, Section 2.7 introduces the designed roll rig, which is used for rudimentary single-axis testing.

2.1 Literature review

The quadcopter platform is a popular means of testing control algorithms [23], as its vertical take-off and landing (VTOL), and steady-state hover capabilities allow for most experiments to be performed indoors, within conventional laboratory confines. The mechanical design of speed-controlled quadcopters is fairly consistent in the literature (for example, [24]), and commonly consists of four motor-propeller units, mounted symmetrically on the ends of a "+" shaped rigid platform (although interesting variations can also be found in the literature, such as in [25]). This structural setup is beneficial, as inertial symmetry about the body-frame x- and y-axes reduces plant coupling in the roll and pitch channels, whilst also allowing for control parity in the two channels [26]. The rotor speeds, and by extension, rotor thrusts, are adjusted by the supplied motor voltages, and actuation of the quadcopter is achieved by varying the combination of the four rotor speeds, resulting in differential moments about the quadcopter's centre of mass. This naturally induces additional dynamics related to the non-zero net angular momentum across the rotor discs.[27]



(a) Speed-controlled quadcopter (b) Variable-pitch quadcopter

Figure 2.1: A (a) speed-controlled quadcopter in flight during TED Talk from Raffaello D'Andrea [1], and (b) variable-pitch quadcopter platform from the work in [2] and [3].

Variable-pitch quadcopters assume the same base structural set-up as the speed-controlled counterparts, albeit with the added components required to enable blade pitch actuation. Unlike conventional speed-controlled quadcopters with uncambered propellers, variable-pitch quadcopters can produce thrust in both directions, which allows for larger control actions in the roll and pitch channels, as well as inverted flight. The most common method of power transmission is to directly drive each rotor with a corresponding motor [28]. This has the benefit of being mechanically simple, but is sub-optimal in terms of reducing the quadcopter's mass moment of inertia. Alternative methods, which aim to minimise the inertia, include using a single central motor and timing belts [29], as well as two central anti-parallel motors driving a set of bevel gears [21]. The trade-off for the reduced inertia is design complexity (more moving parts). Other slight design alterations can be found in the literature, such as the design in [30].

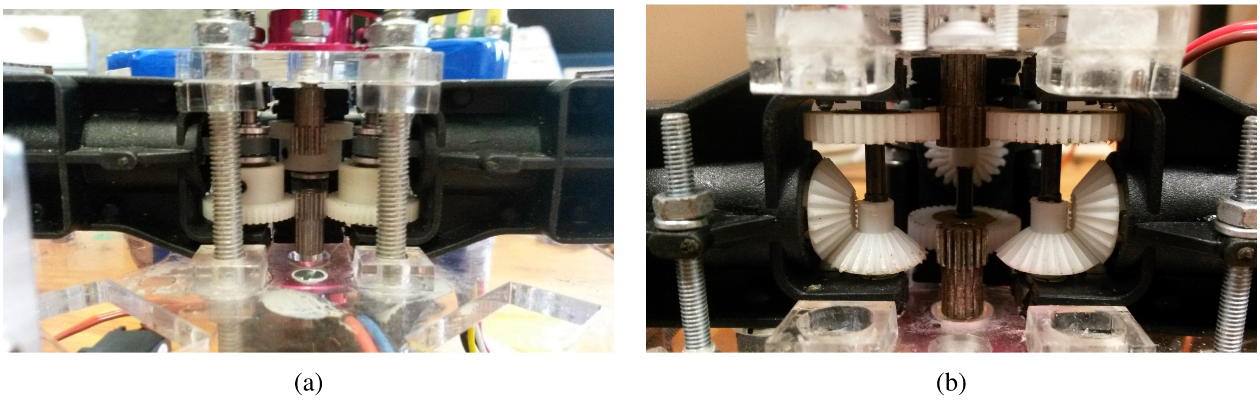
2.2 Variable-pitch quadcopter

A custom-made variable-pitch quadcopter was chosen as the primary experimental platform, as it is a sufficiently complex multivariable system, with innate mechanical and control challenges. Additionally, the mechatronics intricacies involved excite a broad range of engineering disciplines.

2.2.1 Mechanical design

Power transmission system

The mechanical aspects of the variable-pitch quadcopter went through various design iterations, whereby several design improvements were sought. Initially, the four rotors were driven by two co-axially located motors, mounted in the centre axis of the quadrotor, as shown in Figure 2.2a. Power transmission to the rotors was



(a) (b)

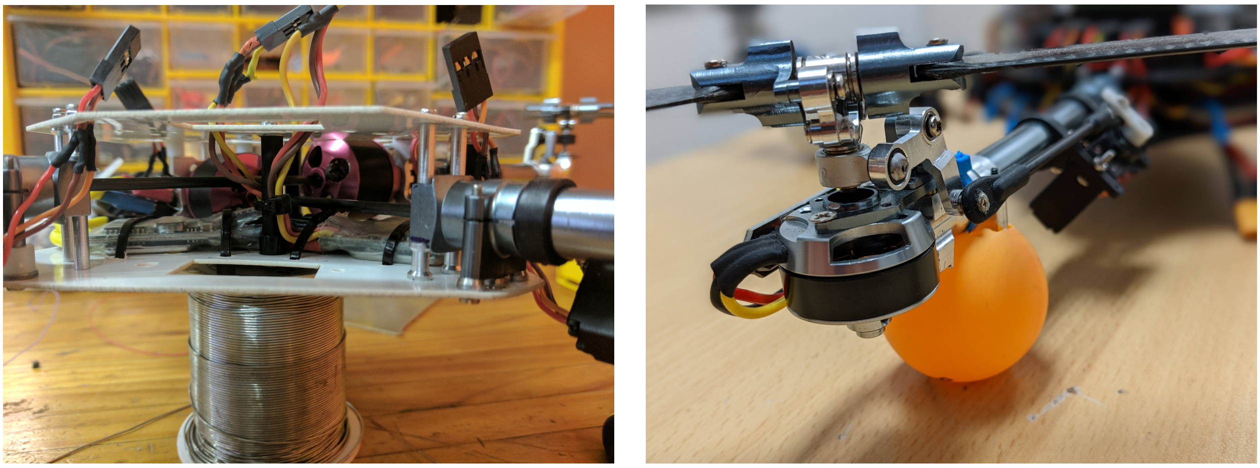
Figure 2.2: (a) First design iteration of variable-pitch quadcopter gearbox, showing co-axially mounted motors driving central gearbox. Power transmission from the gearbox to rotors is facilitated using timing belts. (b) Second design iteration of variable-pitch quadcopter gearbox, showing co-axially mounted motors driving central gearbox. Power transmission from the gearbox to rotors is facilitated using a shaft and bevel gear pair. Both images are shown from the perspective of a removed rotor arm.

achieved using a combination of spur gears and timing belts (one belt per rotor arm). The centralised motors resulted in a lower mass moment of inertia in the roll and pitch channels. Additionally, the anti-parallel configuration of the two motors assisted in balancing the net angular momentum of the robot, thereby removing gyroscopic coupling when the motor speeds were matched (see Section 3.4.2 for more information). However, the large relative torque requirements on the timing belts resulted in slip and wear, and induced large amounts of mechanical vibrational noise on the system, as well as uncertainty in the rotor speeds. Additionally, the constant slipping wore the teeth of the belt and quickly degraded the power transmission from the motors to the rotors.

The second iteration of the transmission system replaced each timing belt and pulley pair with a shaft and bevel gear pair, as seen in Figure 2.2b. This setup exhibited improved vibrational noise characteristics, but the gears had a similar problem in that the rotor torque and high angular velocities were too demanding on the small bevel gears, ultimately resulting in significant gear teeth warping and gear slip.

With reference to Figure 2.3a, the third design iteration located the two motors in line with the shafts running along the quadcopter arms. This had the benefit of removing the first stage of gearing (no longer requiring spur-pinion gearing). Additionally, the reduction in mechanical components made for a lighter quadcopter with fewer moving parts. The longevity of the transmission systems improved using this mechanical configuration, but the bevel gears located on the rotor shafts would eventually experience sufficient wear that resulted in gear slip.

The final power transmission design iteration, shown in Figure 2.3b, simplifies the mechanical setup by directly driving each rotor with a dedicated motor. Although this design results in a higher rotational inertia (reducing the plant gain in the attitude channels), the reduced mechanical complexity is appealing. Additionally, as will be explained in Section 4.4, this configuration allows one to independently estimate the thrust and torque



(a) (b)

Figure 2.3: (a) Third design iteration of variable-pitch quadcopter gearbox, showing motors directly driving rotor shafts. (b) Final design iteration of power transmission system, showing motor directly driving corresponding rotor.

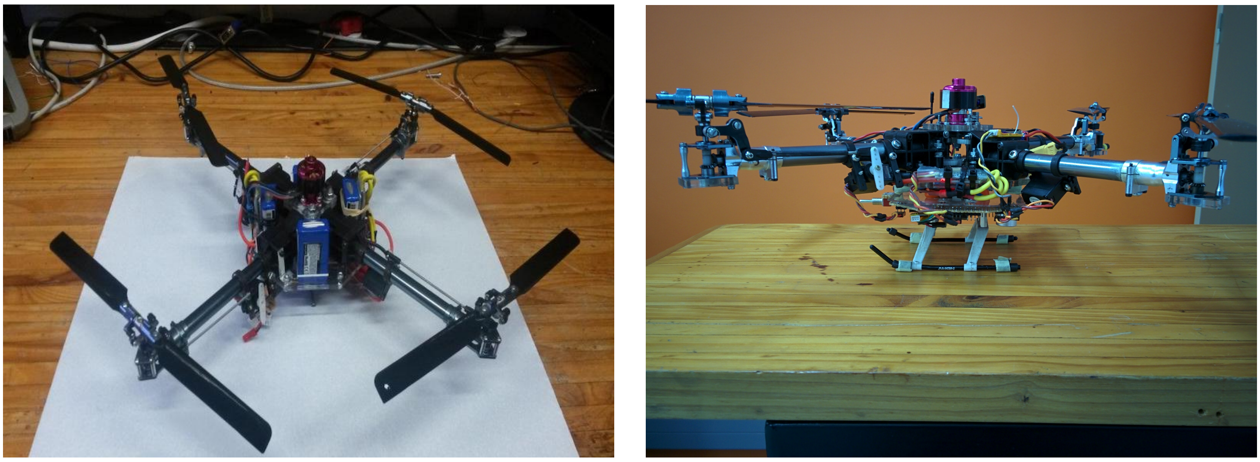
of each rotor. This approach is beneficial for high performance control design (see Section 5.2.3), as low-level inner loop closure can be used to substantially reduce plant uncertainties and nonlinearities. A comparison of the four mechanical design iterations are shown in Figure 2.4.

Servomotor-rotor system

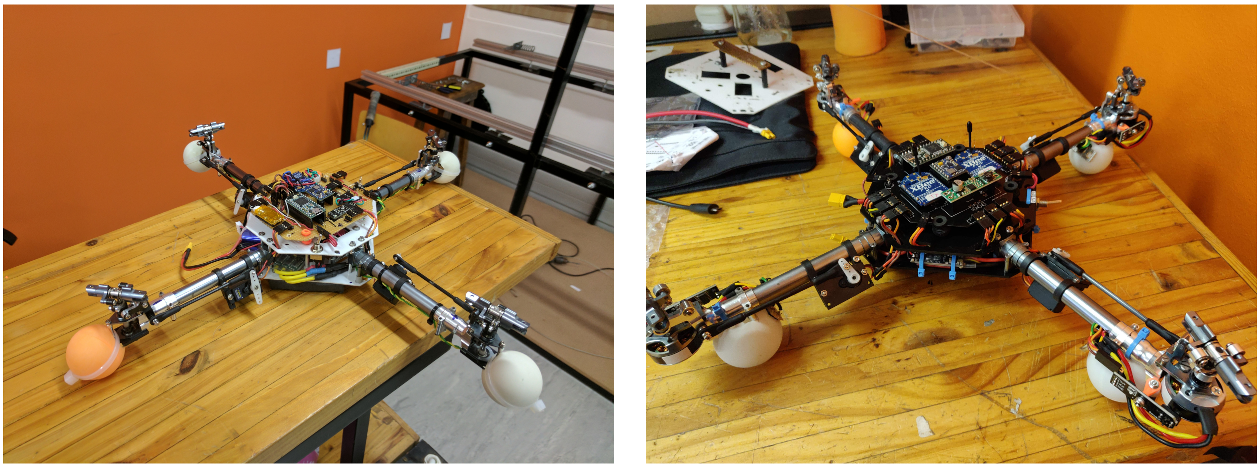
With reference to Figure 2.5, the variable-pitch quadcopter uses four heavily customised helicopter tail rotor units, which are sourced from the HK-450 remote control helicopter. Each tail rotor is located on the end of a hollow, cylindrical aluminium boom, which is connected to the central mounting platform. Each rotor is directly driven by a Turnigy Park 300 brushless DC motor, which is interfaced with by a Flycolor Raptor BLS electronic speed controller (ESC). The motor parameters are summarised in Table 2.1. The motor (and by implication, rotor) speed is measured using a TCRT5000 infra-red reflective switch, which uses an IR transceiver to detect when the motor completes a revolution. The bi-directional tail rotor pitch is achieved using a HK-933 metal gear servomotor, via a transmission system, which consists of multiple stages of servo horns and push rods. The servomotors are mounted on the cylindrical booms using plastic servo mounts and servomotor actuation results in blade pitch actuation, as seen in Figure 2.5-2.6.

Rated nominal voltage	Maximum current	Internal resistance	Motor constant	Dimensions
7.4–11.1 V	9 A	0.18 Ω	1600 $\frac{RPM}{V}$	16 × 28 mm

Table 2.1: Summary of pertinent parameters of the Turnigy Park 300 brushless DC motor as provided by the manufacturer.



(a) First design iteration (b) Second design iteration



(c) Third design iteration (d) Final design iteration

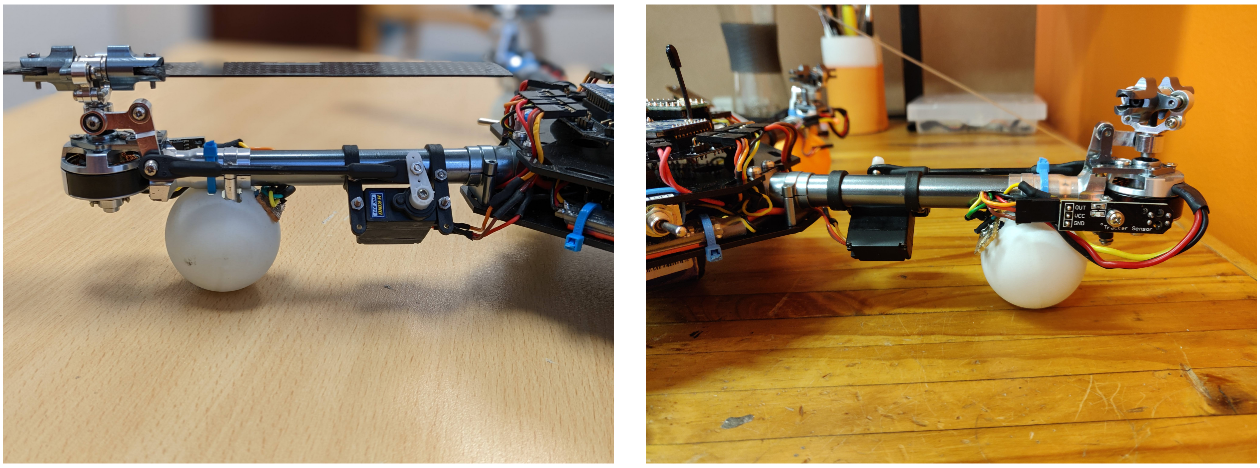
Figure 2.4: A comparison of all four mechanical design iterations of the variable-pitch quadcopter.

Rotor blades

Each tail rotor has a pair of uncambered (symmetrical) carbon fibre blades, which are mounted on the corresponding rotor hub using set screws. The rotor hub assembly is shown in Figure 2.7, and the relevant properties of the rotor blades are summarised in Table 2.2.

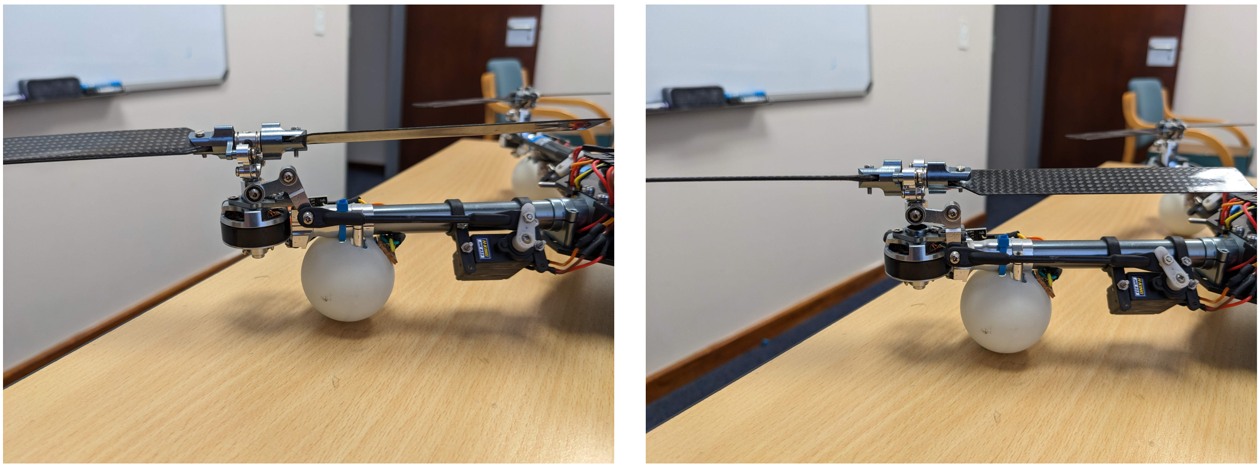
Radial length	Chord length	Thickness	Mass
0.103 m	20 mm	3 mm	4 g

Table 2.2: Mechanical properties of carbon fibre rotor blades.



(a) (b)

Figure 2.5: (a) Front and (b) back view of the variable-pitch quadcopter arm. A heavily modified helicopter tail rotor mechanism is used to achieve the rotor blade actuation.



(a) (b)

Figure 2.6: Two examples of how the blade pitch actuation results from servomotor actuation via the push rods and levers.

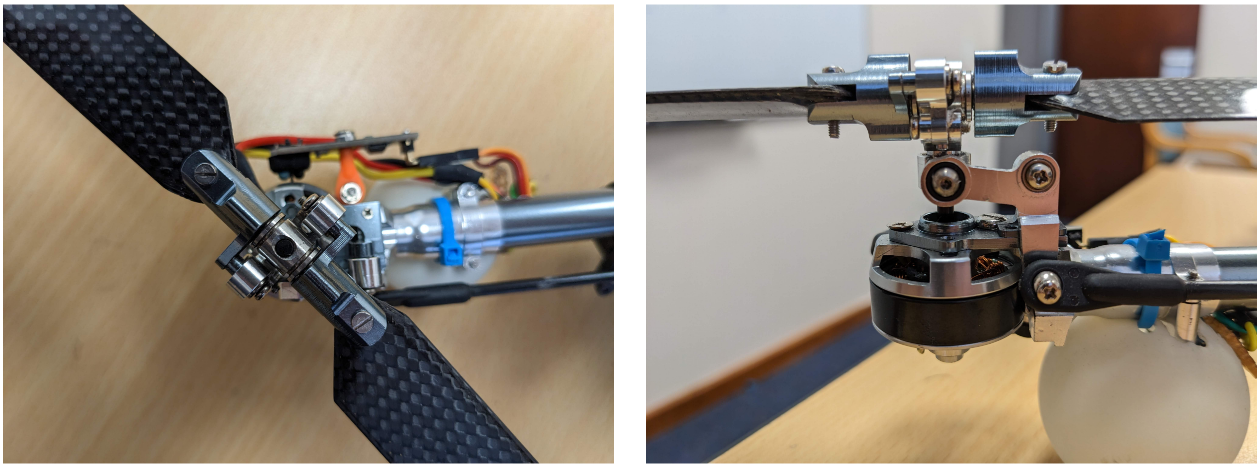

(a) (b)

Figure 2.7: (a) Top-down view and (b) frontal view of rotor hub.

Central platform

As shown in Figure 2.8, the central mounting platform is comprised of two 2mm thick octagonal laser cut ABS sheets, which secure and interface with the cylindrical boom ends. The four electronic speed controllers (ESCs) are mounted on the topside of the bottom mounting platform, whereas a removable battery is mounted on the underside of the bottom plate, and powers the quadcopter during flight. A printed circuit board (PCB), containing the majority of the electronics, sits on plastic vibration dampers, and is mounted on the topside of the top central platform. Cabling from the PCB is routed to the aforementioned peripherals through the cylindrical hollow booms.

Infra-red LED markers

With reference to Figure 2.5, each rotor arm has a table tennis ball mounted near the end of the cylindrical boom. This is primarily necessitated by the motion capture system (see Section 4.2), but also serves as convenient, lightweight landing gear. Each table tennis ball (excluding the orange ball) has an array of infra-red (IR) LEDs inside, and the table tennis balls act as spherical diffusers to distribute the light evenly. The orange table tennis ball is used to visually identify the "front" of the quadcopter during flight.

Power supply

A 3-cell lithium polymer (LiPo) battery is used as the power source on the quadcopter during flight. This battery directly powers the electronic speed controllers, and indirectly powers the other electronics (via a 5 volt or 3.3 volt regulator). The battery capacity and discharge capabilities are selected in order to balance sufficient flight time with low mass.

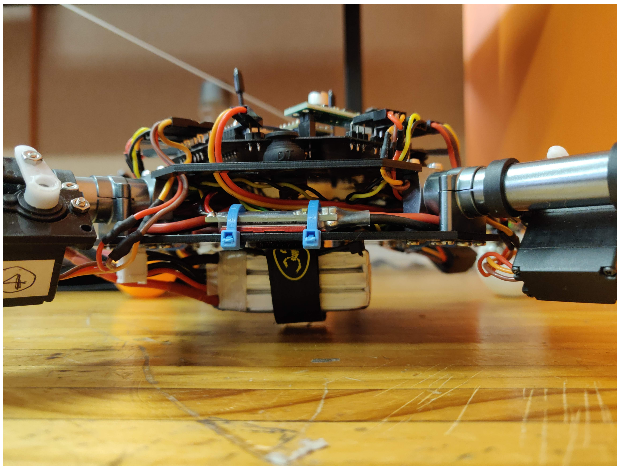


Figure 2.8: Central platform of quadcopter. The majority of the electronics are located within and on top of the central platforms, with the removal LiPo battery mounted below the platform.

Some of the pertinent planar dimensions of the variable-pitch quadcopter are provided in Table 2.3 and depicted in Figure 2.9.

l	r_Δ	w	c
200 mm	103 mm	129 mm	20 mm

Table 2.3: Summary of some of the pertinent dimensions on the quadcopter platform, including blade length, r_Δ, distance from rotor axis to centre of mass, l, central platform width, w, and blade chord length, c.

2.2.2 Instrumentation

As seen in Figure 2.10, the on-board electronics comprise several individual modules, the majority of which are located on a single PCB. The PCB was designed using the program KiCAD, with the thickness chosen as 0.6mm in order to reduce the overall mass of the quadrotor. The schematic for the designed PCB is shown in Figure 2.11, and the top and bottom layouts are shown in Figure 2.12. The four circular cutouts in Figure 2.12 are used as mounting points and interface with plastic vibration dampers (to reduce vibrational noise effects on the on-board sensors).

Micro-controller

The selected micro-controller is a Teensy 3.2, which has an over-clocked processor speed of 96MHz. This ARM-based micro-controller was primarily chosen based on its small form factor, low mass, and large number of timers and timer channels (required for the large number of peripherals that require input capture and output compare). Additionally, the microcontroller's on-board 3.3 volt regulator can source up to 1 ampere of current, removing the need for an external 3.3 volt regulator.

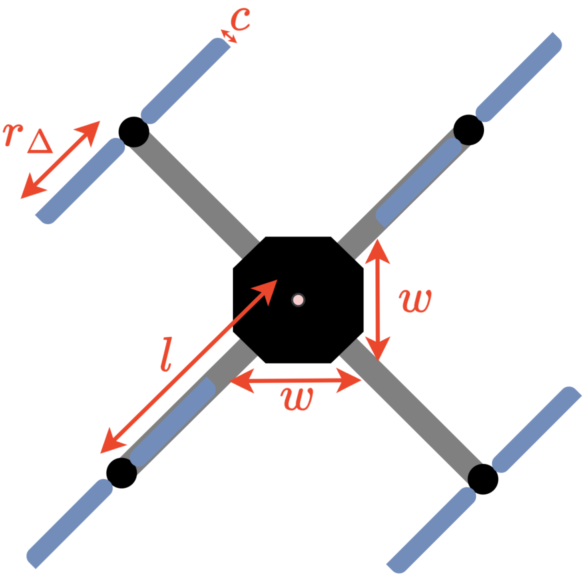


Figure 2.9: Diagram of quadcopter platform with physical dimensions included.

Servomotors

The HK-933 servomotors accept pulse-width modulated (PWM) signals from the micro-controller at a rate of up to 200Hz. This command signal is provided in the form of a 16-bit number, which specifies the active (or "on time") of the PWM signal. Each servomotor has a operating range of approximately $[-90°, 90°]$ and are powered by a 5 volt regulator.

Electronic speed controllers

The Flycolor Raptor BLS ESCs are rated for 30 Amperes of current and are controlled using PWM. As with the servomotors, a 16-bit command signal, sent at up to 200 Hz, specifies the active time of the PWM signal, which is then converted into an average voltage on the ESC. The ESCs interface directly with the motors and schedule the multi-phase operation of the electromagnets based on the commanded signal. The ESCs have a compact and lightweight design that is beneficial for the operation of the relatively small quadcopter. The ESCs are powered by a 5 volt regulator.

Speed encoders

With reference to Figure 2.5, the motor speeds are measured using a TCRT5000 infra-red reflective switch, which directs infra-red light on the motor outrunner as it rotates. A single white stripe (on the otherwise black outrunner of the motor) reflects the infra-red light every revolution, which is then received by an infra-red receiver. Input capture on the Teensy micro-controller is then used to convert the pulse train into a corresponding rotational velocity. Given the sensing scheme of the speed encoders, the effective update rate of each

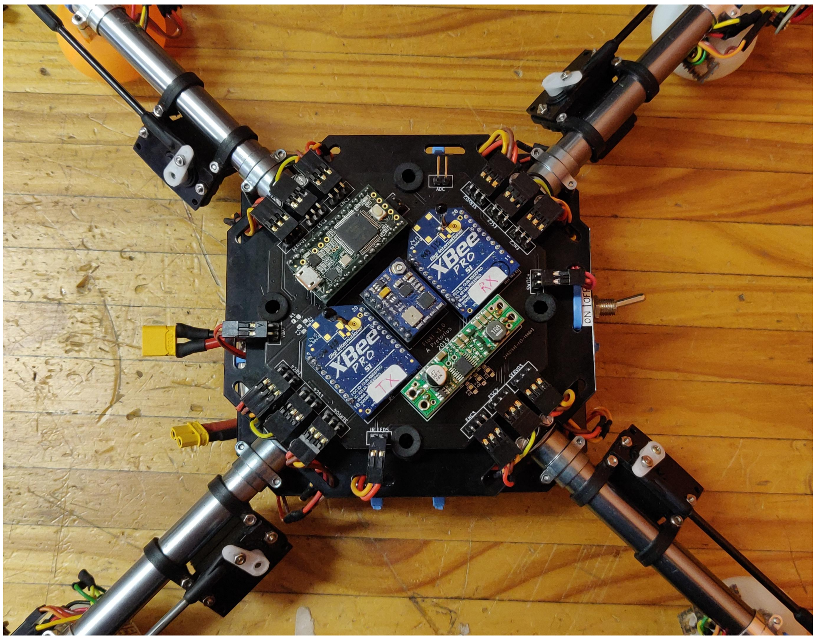


Figure 2.10: PCB of quadrotor electronics, mounted on top of the central platform using vibration dampers.

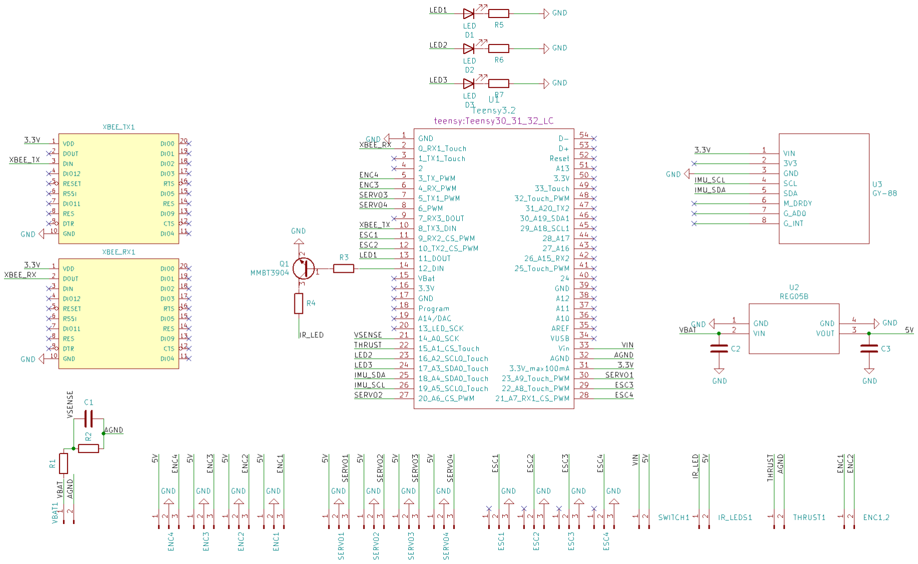
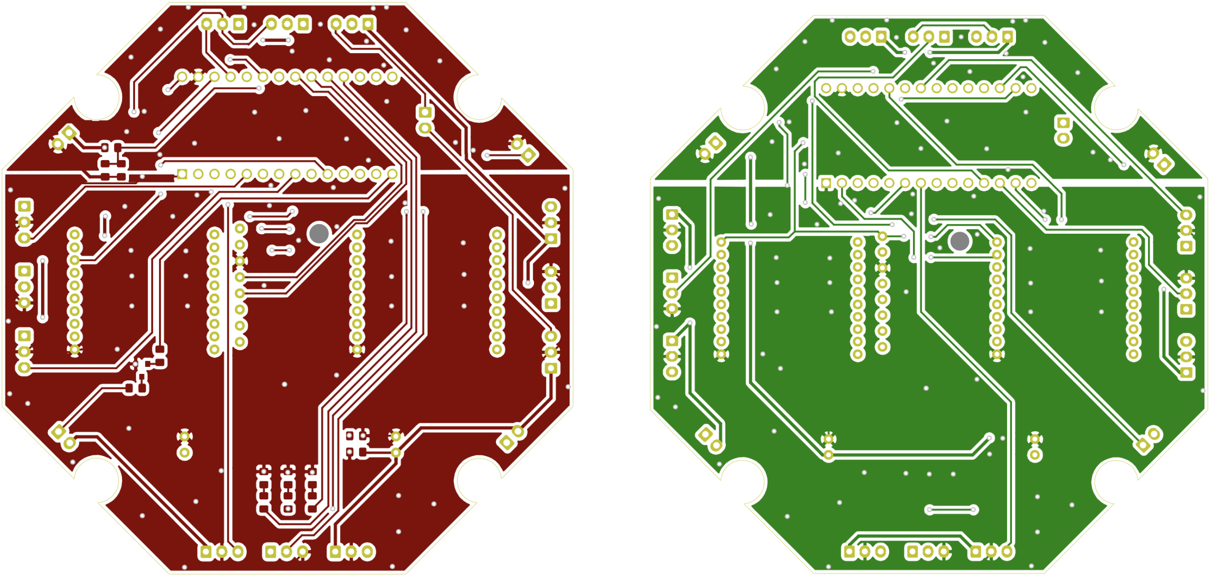


Figure 2.11: Schematic of designed PCB.



(a) Top layout (b) Bottom layout

Figure 2.12: (a) Top and (b) bottom layout of the designed two-layer PCB.

motor speed estimate is proportional to the corresponding motor speed. For more information, see Section 5.2.2.

The rotor speed encoder class is available for viewing in the floki GitHub repository.

Inertial measurement unit

An MPU-9150 inertial measurement unit (IMU) interfaces with the micro-controller, using I2C, and provides 16-bit sampled body-frame gyroscope and accelerometer information, with a nominal update rate of 1kHz. The IMU also has internal low-pass filtering capabilities that can be used to improve the signal-to-noise ratio at the cost of signal bandwidth. The IMU possesses a thermistor and is used to actively account for temperature-dependent drift. The IMU is powered by the micro-controller's 3.3 volt regulator.

The IMU class is available for viewing in the floki GitHub repository.

Communications module

Xbee Pro S1 radio modules are used as the communication link between the quadcopter and ground station, with an over-the-air speed of 250 kb/s. Xbee modules are unable to simultaneously transmit and receive data, so two pairs are used. One on-board Xbee will therefore exclusively transmit data to the ground station Xbee, while the other on-board unit will only receive data. With this scheme, high fidelity, low latency communication is assured, at a transmission rate of 200 Hz. The Xbees communicate with the micro-controller using UART, at a baudrate of 200 kbps. The on-board Xbees are powered by the micro-controller's 3.3 volt regulator.

The quadcopter-based Xbee class is available for viewing in the floki GitHub repository.

Infra-red LEDs

The micro-controller controls the IR LED arrays within the table tennis balls, using PWM, which can adjust the LED brightness as required, as well as turn the arrays off when required. The LED arrays are powered by a 5 volt regulator.

Status LEDs

Three status LEDs are present on the PCB. These are used to convey information about the state of the quadrotor when a fault has occurred (which cannot be broadcasted over radio). For example, if one of multiple critical failures are detected on the micro-controller (such as loss of communication with the base station), the quadrotor will enter an emergency landing procedure, before powering down the motors. The status LEDs will then indicate which criticial failure(s) occurred, which is represented as a binary number based on the three LED states.

Battery voltage and load cell readings

The battery voltage is measured using an analogue-to-digital converter (ADC), after appropriate scaling, filtering, and buffering. Similarly, a second ADC can be used to measure load cell readings from the thrust rig (see Section 2.6 for more information). Both ADCs have 16-bit resolution, and are nominally polled every 5 milliseconds.

Current sensors

The current demand of each motor is measured using an in-line hall effect linear current sensor. The readings from each motor current sensor is then interpreted by the micro-controller using four ADC channels. This allows for real-time electrical torque estimation, and is also used to estimate the rotor thrust (see Section 4.4 for more information).

Voltage regulators

A dedicated 5 volt switch-mode regulator is used to power the micro-controller, servomotors, speed encoders, and IR LED arrays. This regulator is capable of sourcing up to 3 Amperes of current. A 3.3 Volt switch-mode regulator exists on the Teensy micro-controller, and is able to supply up to 1 ampere of current. The 3.3 Volt regulator powers the Xbee communication modules, IMU, and status LEDs. A 12 Volt lithium polymer (LiPo) battery powers the 5 Volt regulator.

Table 2.4 shows the mass breakdown of each major hardware component.

Module	Quantity	Total mass (g)
Tail rotor and boom	4	100
Brushless DC motor	4	120
Electronic speed controller	4	50
Optical speed encoder	4	20
Digital servomotor	4	100
Carbon fibre rotor blade pair	8	32
2mm ABS plastic platform	2	40
PCB motherboard	1	12
IR LED array and diffuser	4	30
LiPo battery	1	80
Linear current sensor	4	20
Miscellaneous (bolts, nuts, etc)	N/A	30
		634

Table 2.4: Mass breakdown of each hardware module on the variable-pitch quadcopter platform.

2.2.3 Software

The Teensy 3.2 micro-controller is programmed in C++. Every 5 milliseconds (equivalent update rate of 200 Hz) the system loop is executed, which polls new data from the IMU and ADCs, updates the on-board estimators, calculates the control action for the various control loops, and sends commands to the PWM channels. The majority of the micro-controller routines are interrupt-based, and are structured such that the system operates in quasi-realtime. As the safety of the quadcopter is inherently sensitive to any peripheral failing, redundancies and watchdogs are also present. The abstracted loop structure of the quadrotor's micro-controller is shown in Table 2.5.

The developed code for the variable-pitch quadcopter can be viewed in the floki GitHub repository.

2.3 Motion capture system

A visual-based motion capture system was chosen as the basis of the final quadcopter state estimator. Other alternatives do exist, such as purely on-board solutions, but the fidelity, full state observability, and relative accuracy is a priority.

Figure 2.13 shows the underside of an early iteration of the variable-pitch quadcopter, which attempted to only use on-board sensors to estimate the quadcopter state vector. The sensor fusion scheme made use of an ultrasonic sensor, optical flow sensor, and IMU to determine the partial state of the quadcopter. The information was fused on the onboard microcontroller and allowed for semi-autonomous near-hover flight, but the lateral position states were unobservable due to the limited sensor information.

Function call	Description
Xbee→receiveData	Read, validate, and decode new data (if available). Clear Xbee watchdog buffer if data is valid.
IMU→getData	Read, temperature compensate, and low-pass filter data.
PoseEstimator→iterate	Fuse IMU and camera information (if new camera data is available). Otherwise rely solely on IMU attitude estimate (open-loop).
BatteryVoltage→getMeasurement	Poll ADC for battery voltage, followed by correct scaling and filtering.
MotorCurrent→getMeasurement	Poll ADC for motor current, followed by correct scaling and filtering.
LoadCellRig→getMeasurement	Poll ADC for load cell reading, scale, and filter (if performing load cell tests).
AttitudeControl→iterate	Update control action for body-frame roll, pitch, and yaw-rate. This makes use of the latest information from the Xbee, IMU, and PoseEstimator function calls.
Encoder→getMotorSpeed	Get motor speed estimates from timer channels.
MotorSpeedControl→iterate	Update control action for motor speeds. Reference speed is obtained in Xbee function call.
ThrustEstimator→getEstimate	Get thrust estimates, using battery voltage and motor current.
ThrustControl→iterate	Update control action for rotor thrusts based on thrust estimate.
PWM→write	Write command signals to servomotor and ESC PWMs.
LED→write	Write command signals to status and IR LEDs.
Xbee→sendData	Send pertinent data to ground station.
WatchDog→refresh	Increment watchdog buffers.

Table 2.5: Abstracted loop structure of quadcopter for single loop iteration.

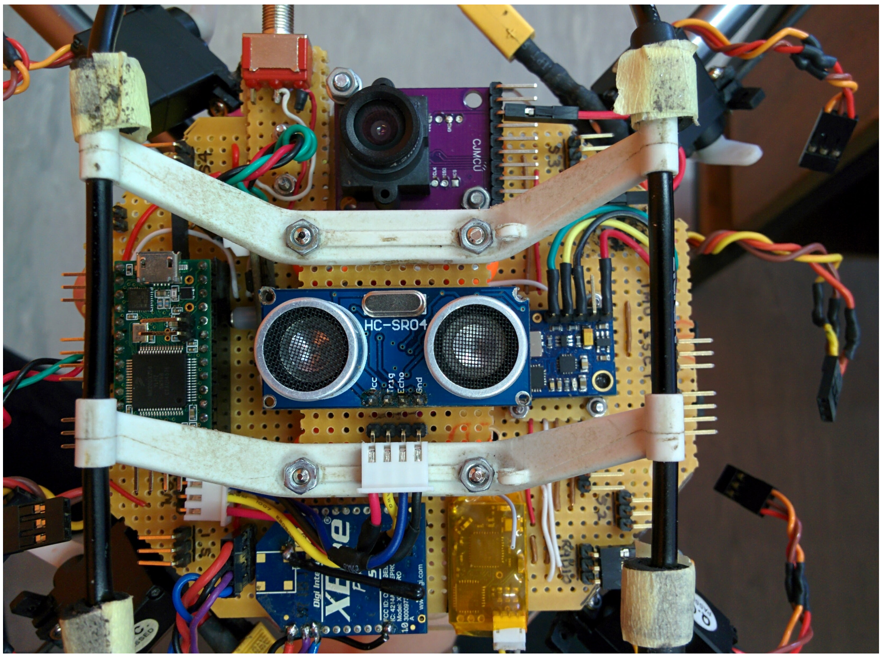


Figure 2.13: The underside of an earlier quadcopter state estimation scheme that made use of only on-board sensors. Information from the ultrasonic sensor, inertial measurement unit, and optical flow sensor were fused to provide partial state estimation.

Following this, the choice was made to instead design a custom visual tracking system that could meet the required performance specifications, whilst also being affordable. This motion capture system is currently being used for other robotics projects that require absolute motion tracking information. Many off-the-shelf solutions exist, which boast high accuracy and low latency, but the cost is exorbitant. Additionally, the tracking specifications, such as accuracy and frame rate, are overkill in terms of the requirements of stabilising and controlling the designed quadcopter.

The physical set-up of the low-cost motion capture system uses five statically mounted Pixy camera modules [31] (see Figure 2.15a) that are spaced and located appropriately to give a redundant capture volume of 5×5×3.5 m. The camera modules have on-board image processors and are able to output multiple features' raw pixel locations at a rate of 50 Hz, with a post-processing resolution of 320×200. Each camera communicates with the ground station PC using UART, at 460800 bps, and the camera information is collated using object-oriented class structures within the base station environment (see Section 2.4). As shown in Figure 2.14, the first iteration of the motion capture system was configured to detect three distinct hues (red, blue, green) that represented the three distinct features on the quadcopter. It was later decided to instead use IR LEDs, as the RGB feature detection was inconsistent and often prone to occlusion issues and false classifications under operation in normal ambient lighting conditions. In order to facilitate IR light detection, a visible light filter was placed between the camera lens and charge-coupled device (CCD) to predominantly pass IR light, whilst attenuating most of the visible light. With reference to Figure 2.15b, this enables the motion capture system

Figure 2.14: First iteration of active markers on quadcopter. The motion capture system was configured to identify dominant blue, red, and green features within a camera image.

to distinctly detect the three IR LED features present on the quadcopter platform, which appears with a white hue. The cameras are calibrated using open-source MATLAB software [32]. With this, the camera intrinsics and distortion coefficients can be found. A simple co-planar calibration algorithm is used to simultaneously find all the cameras' extrinsic parameters. The flight test volume is located in a room that is shielded from any ambient and external light that may affect the visual tracking, such as IR light from the Sun. This is necessary to avoid incorrect feature detection that may result from IR reflective light on other objects in the flight space.

An extended Kalman filter (EKF) is used to fuse information from the cameras, and is able to estimate the pose (location and orientation) of the quadcopter, as well as the corresponding state derivatives. To improve the effective output data rate of the motion capture system, the EKF is iterated every millisecond. As the camera information is only received and processed every 20 milliseconds, this means that the EKF will operate in open-loop outside of the measurement update window, thereby allowing for a higher data output to the quadcopter if desired. For more information, see Section 4.2. The low cost of the modular cameras, and generic class structure of the camera information means that any number of cameras can be added to the system, without any fundamental hardware or software change required. Table 2.6 summarises the main features of the motion capture system. The source and header files for the serial communication between the Pixy cameras and ground station are available for viewing in the flightControl GitHub repository.

2.4 Ground station

The ground station is responsible for, among other things, the processing of the camera and quadcopter information, state estimation, position control, trajectory generation, high-level command signals, and data logging. Additionally, the ground station facilitates the viewing of a graphical user interface (GUI) that provides

(a) Mounted pixy camera module.

(b) Raw image from one of the Pixy camera modules.

Figure 2.15: An image of (a) one of the mounted Pixy cameras, as well as (b) a raw image from the camera. The three white detected features in (b) are the active IR markers, which appear with a white hue and are rigidly located on the quadcopter arms.

Feature	Description
Average positional accuracy	0.6 mm
Average angular accuracy	5.7°
Camera feed latency	8 ms
Update rate	50 Hz
Data output rate	1 kHz
Average execution time of EKF	0.63 ms
Post-processing camera resolution	320×200
Electromagnetic spectrum	Infra-red

Table 2.6: Key features of the stand-alone motion capture system.

flight information to the user. Each primary routine runs on a different CPU thread, avoiding any potential bottle-necking of routines. The full system runs on a program called Qt, and is based in C++. The real-time motion capture EKF is implemented in Qt using a linear algebra library that is able to perform matrix math with sufficient speed [33]. Figure 2.16 shows the abstracted high-level flow of information between routines. The routines are summarised as follows:

2.4.1 Serial communication routine

The serial communication routine governs the flow of information to and from the quadcopter. Data is sent to and from the ground station Xbee modules at a rate of 200 Hz. If the ground station Xbee receiver has new data (received from the quadcopter Xbee transmitter), an interrupt is triggered, which fetches, validates, and decodes the incoming data. Incoming and outgoing data is also time-stamped with microsecond precision. The packets of data that are sent and received are kept as small as possible to reduce transmission delays. For example, only the vectorial part of the camera quaternion is sent to the quadrotor, as the full quaternion can still be recovered (see Section 4.2 for more information). Watchdogs are also set up to monitor invalid data and loss

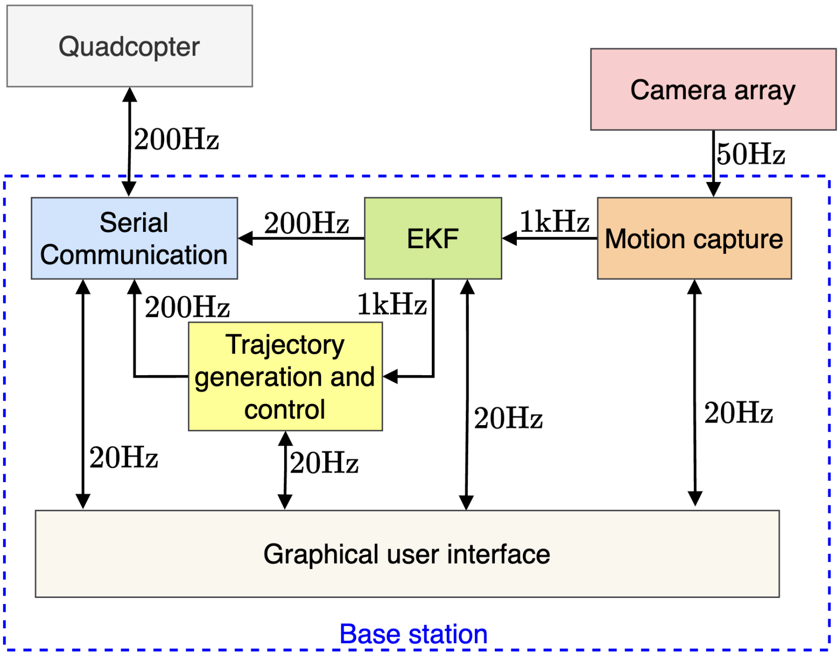


Figure 2.16: Abstracted flow of information between quadcopter, camera, and ground station routines.

of communication. Table 2.7 summarises the data that is sent to and received from the variable-pitch quadcopter.

Data	Type	Number of bytes	Destination
Gyroscope	Signed 16-bit	6	Base station
ECF quaternion vector	Signed 16-bit	6	Base station
Motor currents	Unsigned 8-bit	4	Base station
Servomotor angles	Unsigned 8-bit	4	Base station
Motor speeds	Unsigned 8-bit	4	Base station
Desired thrust vector	Signed 16-bit	6	Quadcopter
EKF quaternion vector	Signed 16-bit	6	Quadcopter
Motor speed reference	Unsigned 8-bit	1	Quadcopter
State machine vector	Unsigned 8-bit	1	Quadcopter

Table 2.7: Summary of data that is sent to and from quadcopter platform. The data type and total size is also included.

The ground station source and header files for the serial communication between the quadcopter and ground station are available for viewing in the flightControl GitHub repository.

2.4.2 Motion capture and EKF routine

This routine is responsible for collating and operating on the camera information. New data is asynchronously received from each camera, every 20 milliseconds. The raw pixel data needs to be validated, normalised, undis-

Function call	Description
Serial→receiveCameraData	Read, validate, and decode new data (if available). Clear watchdog buffer if data is valid.
CameraCalibration→iterate	Normalise, undistort, and apply intrinsic matrix to raw camera data.
CorrespondenceMatching→iterate	Back-project current *a priori* position estimate of features (from EKF) and compare with calibrated camera data. Match, and order features (if valid).
EKF→iterate	Iterate EKF. If data passes correspondence matching test, include data in innovation stage of EKF. Otherwise, EKF is iterated in open-loop (particular measurement is not trusted). EKF watchdog is present to detect potential filter instability (for example if data keeps failing correspondence matching).
DataLogging→iterate	Log EKF and motion capture data to text file (if enabled).
WatchDog→refresh	Increment watchdog buffers.

Table 2.8: Abstracted flow of motion capture routine

torted, and mapped, using the intrinsic properties of the cameras, before being usable. Because the detectable markers used on the quadrotor are three non-unique IR features (from the perspective of each camera), the motion capture routine is also required to perform correspondence matching, in order to keep track of the feature locations with respect to each camera frame. This is done by first projecting the current inertial-frame estimates of the features (obtained from the EKF) into each camera frame. New camera data can then be compared to the back-projected estimates, using a Euclidean distance metric, and the matching can proceed if the data is within a certain distance threshold. This stage is also important for removing false features, as well as accounting for feature occlusions, and feature mergers (two features appearing as one). If the new data is valid, it is used in the measurement stage of the EKF. More detail on the correspondence matching and EKF update procedure is given in Section 4.2. Table 2.8 shows the abstracted flow of the motion capture routine, executed every millisecond.

The following relevant files can be viewed in the flightControl GitHub repository:

- Motion capture source file
- Motion capture header file
- Camera EKF source file
- Camera EKF header file

2.4.3 Radio controller

A 5-channel radio controller is used as an ergonomic means to facilitate calibration, testing, and state transitions. For example, the controller joystick can be used to slew through the servomotors angles when performing the thrust testing (see Section 2.6). Two toggle switches are present on the controller. One is used to toggle between open-loop and closed-loop mode, whereas the other is used to arm/dis-arm the motors. This adds an extra level of fail-safe redundancy. The controller joystick is also used to generate attitude references during attitude-only tests. The radio controller transmits to the base station every 20 milliseconds and makes use of a watchdog to detect potential loss of communication.

The source and header files for the radio controller is available for viewing in the flightControl GitHub repository.

2.4.4 Position control routine

This routine makes use of the position estimate from the motion capture system, whereby simplistic commands are set using the GUI. The control loop operates at 1kHz, and the inertial-frame corrective action required is converted into a required body-frame thrust vector for use by the attitude and thrust controllers. More detail is provided in Chapter 5.

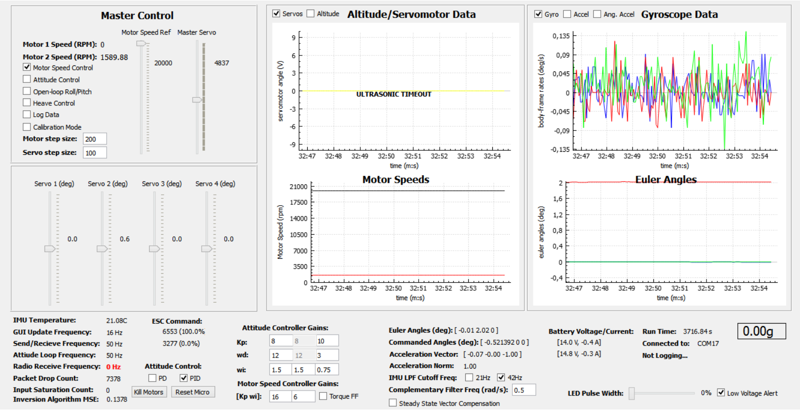
2.4.5 Trajectory generation routine

The trajectory generation routine, if enabled, can supply cubic spline trajectories, or trapezoidal velocity profiles. The user is required to first specify and save the necessary trajectory parameters, which then prompts the routine to determine the time-dependent functions. For more information, see Appendix B.

2.4.6 Graphical user interface

With reference to Figure 2.17, the graphical user interface (GUI) displays pertinent information and has the ability to plot any available data in quasi real-time. The GUI facilitates live interaction with the previously mentioned routines, and allows for different modes of operation, such as calibration, hardware-in-the-loop testing, and flight testing. More importantly, it is able to show warnings and issues, which the hardware/software may be experiencing, such as a partial loss of communication with the quadcopter. Data logging is also enabled using this interface, and is performed every millisecond. The GUI routine has the lowest priority of all the threads, and is updated every 50 milliseconds.

The source and header files for the graphical interface is available for viewing in the flightControl GitHub repository.



Master Control

Motor Speed Ref 20000
Master Servo 4837

Motor 1 Speed (RPM): 0
Motor 2 Speed (RPM): 1589.88
☑ Motor Speed Control
☐ Attitude Control
☐ Open-loop Roll/Pitch
☐ Heave Control
☐ Log Data
☐ Calibration Mode
Motor step size: 200
Servo step size: 100

Servo 1 (deg) 0.0
Servo 2 (deg) 0.6
Servo 3 (deg) 0.0
Servo 4 (deg) 0.0

IMU Temperature: 21.08C
GUI Update Frequency: 16 Hz
Send/Recieve Frequency: 50 Hz
Attiude Loop Frequency: 50 Hz
Radio Receive Frequency: 0 Hz
Packet Drop Count: 7378
Input Saturation Count: 0
Inversion Algorithm MSE: 0.1378

ESC Command:
6553 (100.0%)
3277 (0.0%)

Attitude Control:
☐ PD ☑ PID

Attitude Controller Gains:

Kp:	8	8	10
wd:	12	12	3
wi:	1.5	1.5	0.75

Motor Speed Controller Gains:
[Kp wi]: 16 6 ☐ Torque FF

Euler Angles (deg): [-0.01 2.02 0]
Commanded Angles (deg): [-0.521392 0 0]
Acceleration Vector: [-0.07 -0.00 -1.00]
Acceleration Norm: 1.00
IMU LPF Cutoff Freq: ☐ 21Hz ☑ 42Hz
Complementary Filter Freq (rad/s): 0.5
☐ Steady State Vector Compensation

Battery Voltage/Current:
[14.0 V, -0.4 A]
[14.8 V, -0.3 A]

Run Time: 3716.84 s
Connected to: COM17
Not Logging...

0.00g

LED Pulse Width: 0% ☑ Low Voltage Alert

Figure 2.17: User interface of base station showing various adjustable settings, as well as real-time data visualisation.

2.5 Simulation environment

The simulation environment was developed in MATLAB/Simulink. As shown in Figure 2.18, each subsystem of the quadcopter dynamics is modelled using functional blocks or embedded MATLAB functions. Additionally, real-world effects such as nonlinearities and measurement noise, are emulated to provide a more rigorous representation of the quadcopter's true behaviour. This allows for the various control loops to be stringently tested, before implementing them on the physical platform. The subsystems are summarised below.

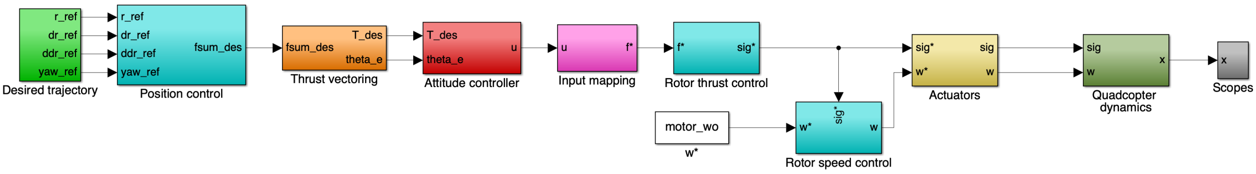
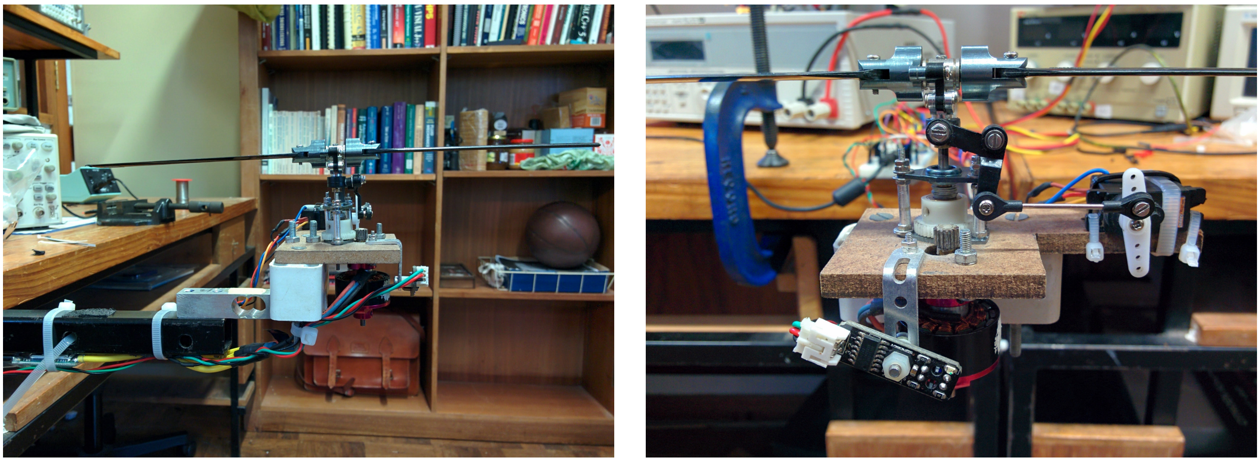


Figure 2.18: Top-level representation of simulation environment using Simulink. The simulation model captures the six-degree-of freedom motion of the quadcopter system, and contains real-world effects, such as measurement noise, digitisation, nonlinearities, and time delay effects.

- **Desired trajectory:** Generates the desired translation position, velocity, and acceleration, as well as the desired yaw angle (if required).
- **Position control:** The position control loop control makes use of the desired and measured position and yaw information to define a desired thrust vector using a PI-like control scheme, based on Section 5.9.2.
- **Thrust vectoring:** This subsystem converts the desired inertial-frame thrust vector into the desired body-frame thrust scalar, and desired orientation, based on Section 5.6.
- **Attitude control:** The attitude loop control uses the desired and measured roll/pitch orientation to reorient the quadcopter's thrust vector in the correct direction. The yaw channel is exclusively used to regulate



(a) Side view. (b) Front view.

Figure 2.19: First version of thrust test rig used to profile static and dynamic thrust and torque profiles.

the angular rate about the body-frame z-axis. See Section 5.8.1 for more detail.

- **Input mapping:** This subsystem converts the desired body-frame heave and torques into the corresponding four desired rotor thrusts based on Section 5.3.
- **Rotor thrust control:** The rotor thrust controllers are responsible for regulating the rotor thrust onboard the quadcopter. The thrust estimate is derived using the motor current and facilitates high-bandwidth uncertainty reduction. See Section 5.2.3 for more detail.
- **Rotor speed control:** The rotor speed controllers regulate the rotor speeds based on the defined nominal operating condition. The expected load torques from changing the blade angles are fed forward to the speed controller to proactively counteract disturbances effects. See Section 5.2.2 for more detail.
- **Actuators:** This subsystem represents the servomotor-rotor and motor speed dynamics, which is based on the system identification experimentation contained in Section 3.6-3.7.
- **Quadcopter dynamics:** This subsystem relates the rotor speeds and blade angles to the rotor thrusts and torques based on Section 3.3 and 3.8. The thrusts and torques are then applied to the rigid body, and the translational and rotational states are derived, as detailed in Section 3.4.

The Simulink model of the variable-pitch quadcopter is available for viewing in the simulationEnvironment GitHub repository.

2.6 Thrust test rig

With reference to Figure 2.19, a simple thrust test rig was constructed, with the intention of performing various types of sub-systems testing, calibration, and system identification. Additionally, this test rig serves as a convenient way to secure the quadcopter when performing hardware-in-the-loop testing and calibration.

A straight bar load cell, with a Wheatstone bridge configuration, is fixed to a rigid steel beam, which is offset from a workbench. The Wheatstone configuration removes rotor torque effects from the load cell readings, thereby only capturing vertical deflections of the bar. An interfacing plate is attached to the load cell bar, which can facilitate different mountings, such as a replication of a quadcopter rotor (as shown in Figure 2.19) or the entire quadcopter platform (useful for hardware-in-the-loop testing). The interfacing plate is sufficiently offset from the ground, such that ground effect (the apparent increase of generated thrust) is negligible.

The strain gauge sensor outputs a differential voltage (proportional to the strain force experienced), which is amplified using an INA101 instrumentation amplifier, followed by an anti-aliasing filter with a 100 Hz cut-off frequency, DC offsetting, and finally signal buffering. The conditioned signal is then sent to a micro-controller, using a 16-bit ADC, and sampled at a rate of 200 Hz. The digital measurement is then further filtered using a second-order digital low-pass filter, with the cut-off frequency and damping factor selected depending on the type of test performed (for example, static vs dynamic thrust tests). Finally, the signal is sent to the base station and logged. In the case of replicating a quadcopter rotor, the motor speed, servomotor angle, and current is also measured and logged. The thrust test rig is calibrated using a set of known masses (within the operating region of the test).

2.7 Roll rig

A simplistic roll rig was designed to perform decoupled, single-axis analysis and control on the roll/pitch channels. As shown in Figure 2.20, the variable-pitch quadcopter is able to move through a continuous 360° rotation. The quadcopter arms are mounted to interfacing plates, and the plates rotate relative to the roll rig pillars using press-fit ball bearings.

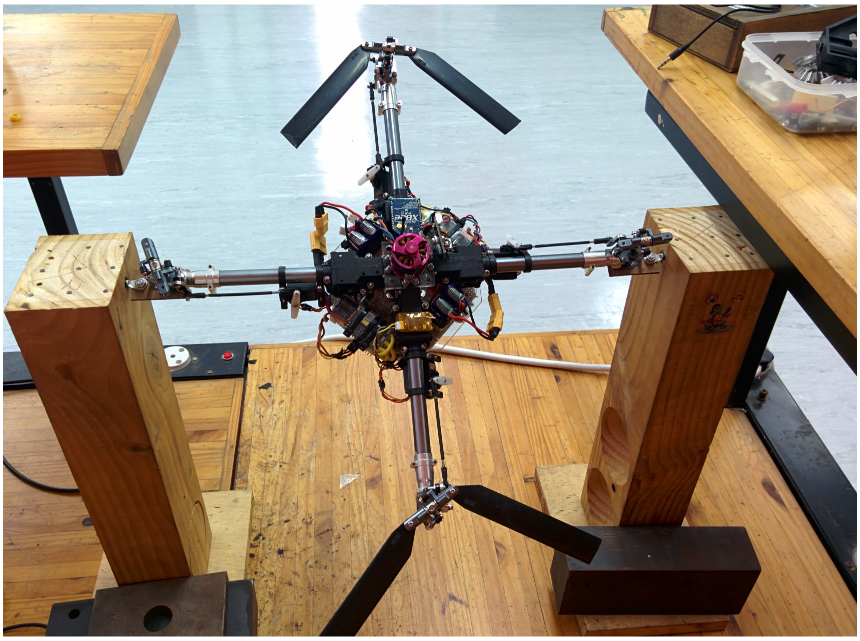


Figure 2.20: Variable-pitch quadcopter attached to roll rig.

Chapter 3

Modelling and system identification

This chapter is based, in part, on the following first-authored and published paper:
Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres [21].

Section 3.1 reviews the literature relevant to modelling and system identification of a quadcopter. Section 3.2 defines the reference frame convention that is used throughout the dissertation, as well as the state variables that describe the quadcopter pose. Section 3.3 develops the rotor thrust and torque equations from first principles using a combination of blade element theory and momentum theory. The generalised rigid-body translational and rotational dynamics of the variable-pitch quadcopter are shown in Section 3.4. Section 3.5 provides information on the chosen nominal operating conditions of the variable-pitch quadcopter system. The servomotor-rotor model, relating the commanded servomotor angle to the corresponding rotor blade angle, is provided in Section 3.6. Section 3.7 applies system identification techniques to the rotor speed system, in order to develop a linearised, open-loop relationship between the commanded servomotor angle, motor voltage, and rotor speed. Section 3.8 makes use of the thrust test rig to confirm the mathematical thrust and torque derivations from Section 3.3. Additionally, this section uses system identification methods to model the open-loop rotor thrust and torque transfer functions in the operating region of interest.

3.1 Literature review

The linearised near-hover dynamics of the conventional speed-controlled quadrotor are widely known [34] [35], and are generally used to design control solutions capable of achieving conditional stability in the vicinity of the quadcopter's steady-state [36], [37], [38]. The controller performance, however, suffers during large and/or fast multi-plane commands, when the simplified linear model fails to approximate the true dynamics [39]. The author in [40] performs closed-loop system identification, while flying, to find an accurate near-hover first order model of the quadcopter. Research conducted in [41] addresses the propeller aerodynamics in more detail, accounting for effects such as induced velocity and blade flapping. The induced velocity, and the various flight states associated with it, are extensively investigated in [27]. Work in [42] exhaustively captures the nonlinear drag effects present on a quadcopter when translating at non-negligible velocities, such as body drag, rotor drag, and the effect of motion on the induced velocity. The authors in [43] use a first-order approximation of the aerodynamic drag on the quadcopter body, in order to improve the closed-loop performance of the system when translating. Similarly, researchers in [44] incorporate a linear drag model to improve on-board attitude estimation. In most work revolving around control design for a quadcopter, the concept of plant uncertainty is neglected. The work in [45] attempts to capture the uncertainty in a plant set, for use in designing a QFT-based controller, but does not address the multi-input-multi-output (MIMO) coupling the quadcopter would experience away from hover conditions. The authors of [46] and [47] describe the model uncertainty using a set of equivalent disturbances, which acts on the defined nominal linear model. The equivalent disturbance set is then used to improve on the nominal control design, by means of a robust compensator on the outer loop of the system.

3.2 Definition of axes and states

With reference to Figure 3.1, the quadcopter is modelled as a rigid-body vehicle and makes use of the east-north-up (ENU) axis convention, which obeys the right-hand rule. An inertial-frame (or world-frame) axis system

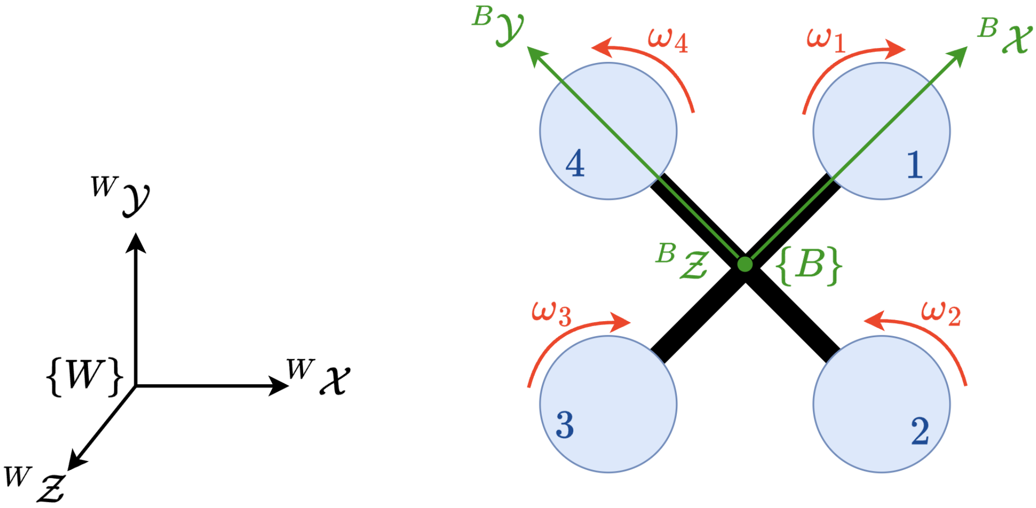


Figure 3.1: A visualisation of the quadcopter's chosen body-frame and inertial-frame reference system. The rotor enumerations are shown, as well as the corresponding rotor directions. Note that the reference frames obey right-hand rule.

is attached to the static test environment and is denoted as frame $\{W\}$. The body-frame axis system, denoted as frame $\{B\}$, is located on the quadcopter body, whereby the frame $\{B\}$ origin coincides with the perceived quadcopter centre of mass. The quadcopter position in the inertial-frame, $^W\mathbf{p}$, corresponds with the location of the frame $\{B\}$ origin with respect to frame $\{W\}$, and the quadcopter orientation is encoded using the quaternion

$$ {}^W\mathbf{q}_B = \begin{bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{bmatrix}_B , \tag{3.1} $$

which captures the orientation of frame $\{B\}$ with respect to frame $\{W\}$. The quaternion, while over-detetermined, does not experience gimbal lock, as is the case with the more common approach of using Euler angles. This is especially important if the operating region of the quadcopter orientation spans the entire 3DOF orientation space. For more information on quaternions and rotation matrices, see Appendix A or [48].

The inertial-frame velocity, ${}^W\mathbf{v}$, and body-frame velocity, ${}^B\mathbf{v}$, are solely related by the rotation matrix ${}^W\mathbf{R}_B = \mathbf{R}({}^W\mathbf{q}_B)$, namely ${}^W\mathbf{v} = {}^W\mathbf{R}_B{}^B\mathbf{v}$, where ${}^W\mathbf{v} = {}^W\dot{\mathbf{p}}$. The body-frame z-axis, ${}^B\mathcal{Z}$, originating from the quadrotor centre of mass, lies parallel to the motor axes, with ${}^B\mathcal{X}$ and ${}^B\mathcal{Y}$ coinciding with rotor arm 1 and 4, respectively. The vector ${}^B\mathbf{\Omega} = {}^B\left[\Omega_x\ \Omega_y\ \Omega_z\right]^T$ describes the body-frame rotational rate about the defined centre of mass.

The quadcopter has a constant mass, m, and rotational inertia, $\mathbf{J}$. The motor (and by implication, rotor) angular velocity vector is defined as $\boldsymbol{\omega} = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$, whereby the motors are configured such that the rotation vectors, ω_1 and ω_3, are anti-parallel to that of ω_2 and ω_4, with $\text{sgn}(\omega_1) = \text{sgn}(\omega_3) = -1$, and $\text{sgn}(\omega_2) = \text{sgn}(\omega_4) = 1$, based on right-hand rule.

Each rotor arm has a servomotor that can adjust the blade angle. The regulated servomotor angle 4-vector is defined as $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4]^T$. The blade angle vector is directly related to the servomotor vector, and is expressed as $\boldsymbol{\sigma} = [\sigma_1\ \sigma_2\ \sigma_3\ \sigma_4]^T$, with the positive direction of each blade angle corresponding to a positive body-frame change in the associated rotor thrust. Note that $\boldsymbol{\sigma}$ is not measured directly on the platform. The two vectors, $\boldsymbol{\omega}$ and $\boldsymbol{\alpha}$, make up the 8 control degrees of freedom (DoF) of the quadcopter. The quadcopter's full (measurable) state vector follows as

$$ \mathbf{x} = \begin{bmatrix} {}^W\mathbf{p} \\ {}^W\mathbf{v} \\ {}^W\mathbf{q}_B \\ {}^B\mathbf{\Omega} \\ \boldsymbol{\omega} \\ \boldsymbol{\alpha} \end{bmatrix} . \tag{3.2} $$

The states are described using the metric system, with the various units of measurement shown in Table 3.1.

State	Description	Unit of measurement
${}^{W}\mathbf{p}$	Inertial-frame position	metres (m)
${}^{W}\mathbf{v}$	Inertial-frame velocity	metres per second (m/s)
${}^{W}\mathbf{q}_{B}$	Inertial-frame quaternion	unitless
${}^{B}\boldsymbol{\Omega}$	Body-frame rotational velocity	radians per second (rad/s)
$\boldsymbol{\omega}$	Rotor speed vector	radians per second (rad/s)
$\boldsymbol{\alpha}$	Servomotor angle vector	radians (rad)

Table 3.1: Description of quadcopter states and corresponding units of measurement.

3.3 Rotor thrust and torque equations

The quadcopter possesses four rotors, each of which produce individual thrust and torque. Understanding the thrust and torque behaviour is important, as the collective thrust and torque fundamentally dictate the quadcopter's manoeuvrability and controllability whilst in flight. Rotor k will produce a thrust, f_k, and torque, τ_k, with the directional convention as shown in Figure 3.2. With reference to Figure 3.1-3.2, f_k and τ_k are parallel to the body-frame z-axis, ${}^{B}\mathcal{Z}$. Note that τ_k is anti-parallel to the corresponding rotor velocity, ω_k ($\mathrm{sgn}(\tau_k) = -\mathrm{sgn}(\omega_k)$), and can be thought of as a reaction torque on the body as a result of the aerodynamic effects on the propellers.

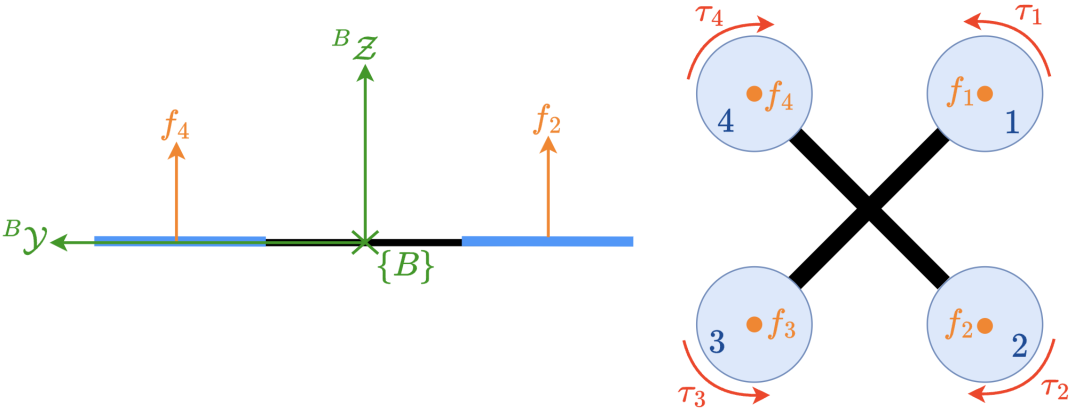


Figure 3.2: Diagrams indicating the direction of the rotor thrust and torque. Thrust and torque follows right-hand rule convention.

The rotor thrust and torque is modelled using a collection of momentum theory and blade element theory. The derivations are provided in the sections that follow.

3.3.1 Actuator disc theory

Actuator disc theory [49], also known as Momentum theory, describes the mathematical behaviour of an ideal actuator disc such as a propeller or helicopter rotor. The propeller is seen as an infinitely thin disc, which creates a constant velocity along the axis of the rotor. Under the correct conditions, a mathematical relationship can be made between the power, propeller length, motor torque, and induced velocity. The effect of losses is omitted.

In the context of this work, momentum theory is used to model the induced blade velocity, and its effect on the resulting rotor thrust and torque. Notably, the modelling approach in this section assumes that the actuator disc is in a vertical state, and does not experience translational velocity. In reality, the rotor discs of a quadcopter will experience lateral velocities during transient position changes. However, the simplifying assumptions presented in this section are still able to capture the predominant wake interaction behaviour that is expected to act on the quadcopter rotors.

Consider Figure 3.3 where the rotor axis, shown in the figure as an ideal actuator disc, is in a vertically orientated state. In the body-fixed frame, f is the rotor thrust, ϵ_a is the air velocity at various positions in the air stream, ϵ_i is the induced air flow from the rotating blades, and ϵ_c is the climb velocity. The airflow is assumed to be steady, uniform and incompressible. The mass flow rate of the air through the rotor, $\dot{m}$, can be described as

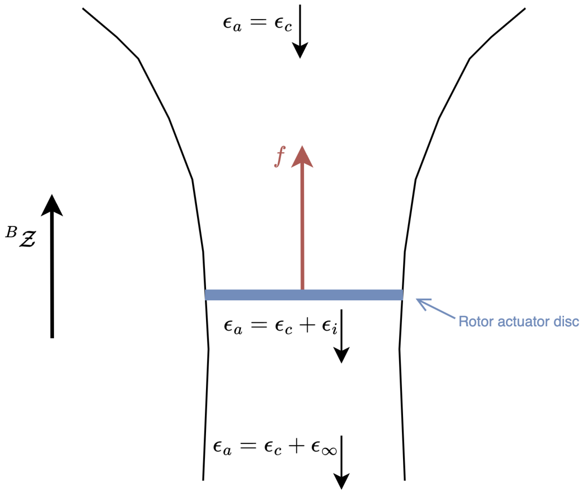


Figure 3.3: Airflow experienced by a rotor during a generalised vertical climb.

$$\dot{m} = \rho A(\epsilon_c + \epsilon_i) \tag{3.3}$$

where A is the rotor disc area. The rate of change of momentum between the undisturbed upstream conditions and the far wake can be equated to the generated thrust by

$$f = \dot{m}(\epsilon_c + \epsilon_\infty) - \dot{m}\epsilon_c = \dot{m}\epsilon_\infty \tag{3.4}$$

where ϵ_∞ is the induced air flow in the fully developed wake. The power in the air can be equated to the total rate of change of kinetic energy of the flow,

$$P_i = f(\epsilon_c + \epsilon_i) = \frac{1}{2}\dot{m}(\epsilon_c + \epsilon_\infty)^2 - \frac{1}{2}\dot{m}\epsilon_c^2 = \frac{1}{2}\dot{m}(2\epsilon_c\epsilon_\infty + \epsilon_\infty^2). \tag{3.5}$$

Using (3.4) and (3.5), the induced velocity in the far wake is deduced to be twice the rotor induced velocity,

$$\epsilon_\infty = 2\epsilon_i. \tag{3.6}$$

Using this result in conjunction with (3.3) and (3.4), the rotor thrust can be written directly in terms of rotor disc conditions as

$$f = 2\rho A(\epsilon_c + \epsilon_i)\epsilon_i. \tag{3.7}$$

Using the quadratic formula, the induced velocity is solved as

$$\epsilon_i = \frac{\sqrt{\epsilon_c^2 + \frac{2f}{\rho A}} - \epsilon_c}{2} \tag{3.8}$$

where the spurious solution of ϵ_i and ϵ_c in opposite directions is ignored. Note that in the case of the hover state ($\epsilon_c = 0$), the induced velocity reduces to

$$\epsilon_h = \sqrt{\frac{f}{2\rho A}}. \tag{3.9}$$

3.3.2 Lifting-line theory

According to lifting-line theory [50], the lift coefficient of a symmetrical, uncambered blade, with an elliptical cross-section, can be approximated as

$$c_L(\eta) = 2\pi\left(\frac{\kappa}{\kappa+2}\right)\eta = c_{L\eta}\eta, \tag{3.10}$$

where κ is the aspect ratio of the rotor blade, and η is the effective angle of attack of the rotor blade. By extension, the drag coefficient can be approximated as

$$c_D(\eta) = c_{Do} + \frac{c_L(\eta)^2}{\pi\kappa} = c_{Do} + \frac{4\pi\kappa}{(\kappa+2)^2}\eta^2 = c_{Do} + c_{D\eta}\eta^2, \tag{3.11}$$

where c_{Do} is the parasitic drag coefficient, which accounts for drag effects on the uncambered blade when the effective angle of attack is zero.

The aspect ratio of an aerofoil is the ratio between the effective blade length, r_Δ, and chord length, c,

$$\kappa = \frac{r_\Delta}{c}. \tag{3.12}$$

With reference to Table 2.3, the expected aspect ratio of the quadcopter blades is $\kappa = 5.15$, with an expected lift and drag coefficient constant of $c_{L\eta} = 4.53$ and $c_{D\eta} = 1.27$, respectively. c_{Do} can be determined by experimentation.

3.3.3 Blade element theory

Blade element theory [51] divides the propeller blade into infinitely many independent sections along the blade length. At each section, a force balance is applied, which involves the component force lift and drag and the corresponding thrust and torque produced. The total forces are then calculated by integrating over the propeller radius. A differential blade element of constant chord c and width dr at distance of r from the rotor axis is shown in Figure 3.4. As shown in Figure 3.5 the same blade element, shown now as a 2D cross-section, experiences a climb velocity, ϵ_c, linear velocity, $\omega\vec{r}$, and the induced velocity, ϵ_i. The climb velocity and induced velocity are anti-parallel to the rotor axis. β is the angle between the resultant air flow and the rotor plane of rotation, known as the inflow angle, η is the effective angle of attack, and $\sigma = \beta + \eta$ is the blade pitch angle. The vector sum of these velocities is

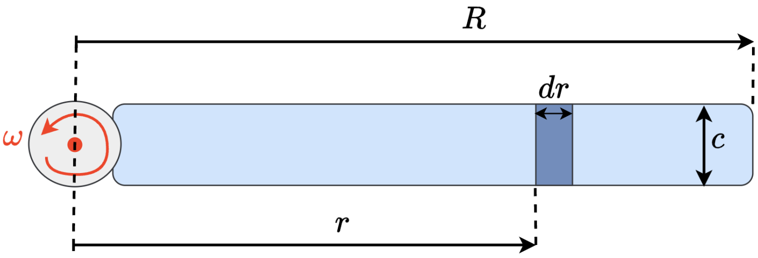


Figure 3.4: Top down view of rotating propeller blade with respective propeller dimensions.

$$\vec{v}_R = \omega\vec{r} + \vec{\epsilon}_c + \vec{\epsilon}_i. \tag{3.13}$$

The element lift, dL, acts perpendicular to the resultant airspeed vector, $\vec{v}_R$, whilst the element drag, dD, acts in parallel. The blade element lift and drag is described as

$$dL = \frac{1}{2} c_L(\eta)\rho|\vec{v}_R|^2 c dr, \tag{3.14}$$

$$dD = \frac{1}{2} c_D(\eta)\rho|\vec{v}_R|^2 c dr, \tag{3.15}$$

where ρ is the air density, $c_L(\eta)$ is the lift coefficient as a function of the angle of attack, $c_D(\eta)$ is the drag coefficient as a function of the angle of attack, and the magnitude of the resultant airspeed relative to the blade

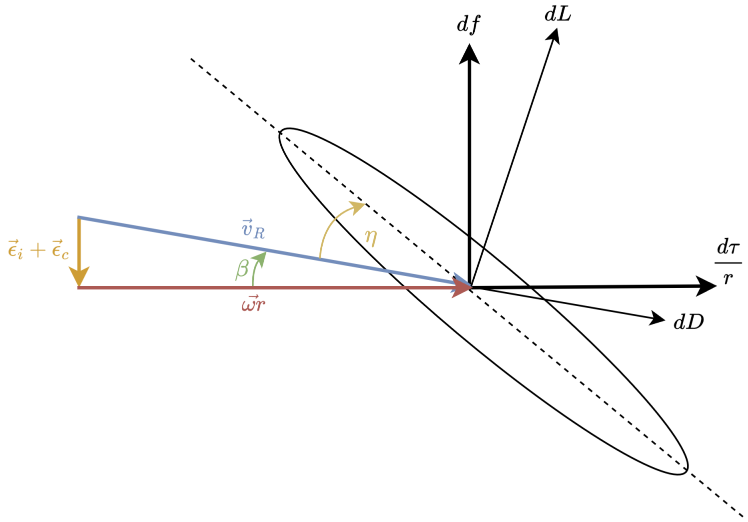


Figure 3.5: Propeller blade element velocity and force diagram.

element is

$$|\vec{v_R}| = \sqrt{w^2r^2 + (\epsilon_c + \epsilon_i)^2}. \tag{3.16}$$

The blade element thrust, df, and torque per radius, $\frac{d\tau}{r}$, are made up of the blade lift and drag, dL and dD respectively as

$$df = dL\cos(\beta) - dD\sin(\beta), \tag{3.17}$$

$$\frac{d\tau}{r} = dD\cos(\beta) + dL\sin(\beta). \tag{3.18}$$

If the number of propeller blades is n then,

$$df = \frac{1}{2}\rho cn[c_L(\eta)\cos(\beta) - c_D(\eta)\sin(\beta)][\omega^2r^2 + (\epsilon_c + \epsilon_i)^2]dr \tag{3.19}$$

$$d\tau = \frac{1}{2}\rho cn[c_L(\eta)\sin(\beta) + c_D(\eta)\cos(\beta)][\omega^2r^2 + (\epsilon_c + \epsilon_i)^2]rdr. \tag{3.20}$$

The inflow angle is calculated as

$$\beta = \tan^{-1}\frac{|\vec{\epsilon}_c + \vec{\epsilon}_i|}{|\omega\vec{r}|}, \tag{3.21}$$

and the small angle approximation of (3.21) follows as

$$\beta = \frac{|\vec{\epsilon}_c + \vec{\epsilon}_i|}{|\vec{\omega r}|}. \tag{3.22}$$

The inflow angle is generally assumed to be small. Consulting (3.22), this seems questionable near the rotor axis where $|\vec{\omega r}|$ is small, but there the blade loads are also small. The following suitable approximations can therefore be made:

$$|\vec{v}_R| \approx |\vec{\omega} r| \tag{3.23}$$

$$df \approx dL \tag{3.24}$$

$$d\tau \approx (\beta dL + dD)r. \tag{3.25}$$

Continuing from these approximations and making use of the small angle approximation, the thrust and torque element equations becomes

$$df = \frac{1}{2}\rho c n c_L(\eta)\omega^2 r^2 dr, \tag{3.26}$$

$$d\tau = \frac{1}{2}\rho c n \omega^2 [\frac{\epsilon_i + \epsilon_c}{\omega r} c_L(\eta) + c_D(\eta)] r^3 dr. \tag{3.27}$$

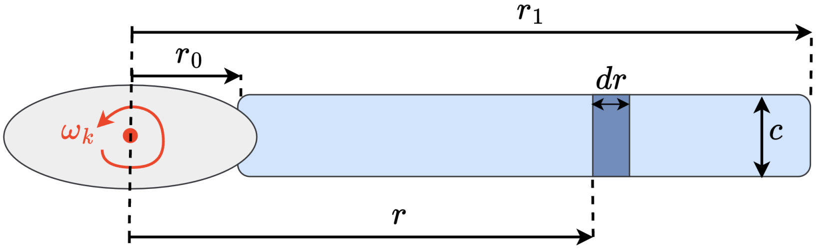


Figure 3.6: Simplified abstraction of top-down view of rotating propeller blade, with respective propeller dimensions. The rotor blade (blue) is attached to a rotor hub (grey) and is offset from the rotor axis by an amount of r_0 metres.

3.3.4 Determining rotor thrust and torque

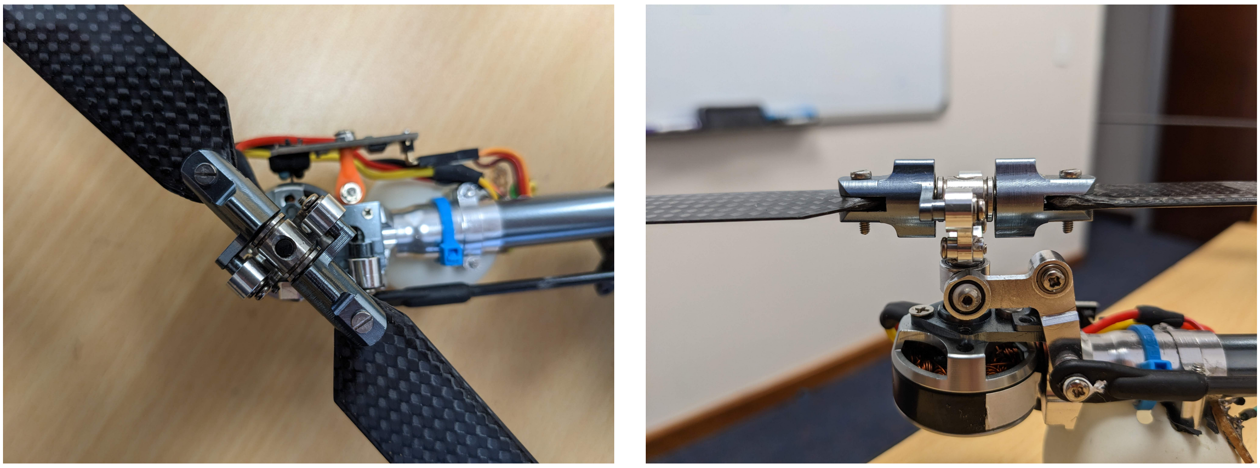
With reference to Figure 3.6, (3.26) and (3.27) are combined with the lift and drag coefficients in Section 3.3.2 to describe the blade element thrust and torque for rotor k as

$$df_k = \frac{1}{2}\rho c n c_{L_\eta} \eta_k \omega_k^2 r^2 dr, \tag{3.28}$$

$$d\tau_k = \frac{1}{2}\rho c n \left[\omega_k \epsilon_k c_{L_\eta} \eta_k r^2 + \omega_k^2 (c_{D_o} + c_{D_\eta} \eta_k^2) r^3\right] dr, \tag{3.29}$$

where $\eta_k = \sigma_k - \frac{\epsilon_{i_k} + \epsilon_{c_k}}{\omega_k r}$, is the effective element angle of attack introduced in Section 3.3.3, and $\epsilon_k = \epsilon_{i_k} + \epsilon_{c_k}$ is the airflow through the rotor disc, perpendicular to the rotor plane.

The total thrust and torque per rotor are found by integrating along the blade length. With reference to Figure 3.7, the blades are connected to a rotor hub, making the effective blade length range from r_0 to r_1 (the rotor hub does not contribute to differential lift and drag). This effectively changes the rotor disc (over a full rotor sweep) into an annulus shape. The rotor thrust and torque derivations follow as



(a) Top-down view

(b) Frontal view

Figure 3.7: (a) top-down view and (b) frontal view of rotor blades connected to rotor hub.

$$f_k = \int_{r_0}^{r_1} \frac{1}{2}\rho c n c_{L_\eta}\eta_k\omega_k^2 r^2 dr, \tag{3.30}$$

$$\tau_k = \int_{r_0}^{r_1} \frac{1}{2}\rho c n \left(\omega_k\epsilon_k c_{L_\eta}\eta_k r^2 + \omega_k^2(c_{D_o} + c_{D_\eta}\eta_k^2)r^3\right) dr. \tag{3.31}$$

After some simplification, the rotor thrust and torque equations become

$$f_k = \frac{1}{2}\rho c n c_{L_\eta}\omega_k r_\Delta^2 \left(\sigma\omega_k\frac{r_\Delta}{3} - \frac{1}{2}\epsilon_k\right), \tag{3.32}$$

$$\tau_k = \frac{1}{2}\rho c n r_\Delta^2 \left(\frac{1}{2}\epsilon_k^2(c_{D_\eta} - c_{L_\eta}) + \omega_k\epsilon_k\frac{r_\Delta}{3}\sigma_k(c_{L_\eta} - 2c_{D_\eta}) + \omega_k^2 c_{D_\eta}\sigma_k^2\frac{r_\Delta^2}{4} + \omega_k^2 c_{D_o}\frac{r_\Delta^2}{2}\right), \tag{3.33}$$

where $r_\Delta = r_1 - r_0$ is the blade length from Table 2.3. Following from this, the thrust equation for rotor k can be written as

$$f_k = a_\sigma\sigma_k - a_\epsilon\epsilon_k, \tag{3.34}$$

where

$$\begin{aligned} a_\sigma &= \frac{1}{6}\rho c n c_{L_\eta}\omega_k^2 r_\Delta^3, \\ a_\epsilon &= \frac{1}{4}\rho c n c_{L_\eta}\omega_k r_\Delta^2. \end{aligned} \tag{3.35}$$

Similarly, the torque for rotor k is expressed as

$$\tau_k = b_\epsilon\epsilon_k^2 + b_{\epsilon,\sigma}\epsilon_k\sigma_k + b_\sigma\sigma_k^2 + b_\omega \tag{3.36}$$

where

$$
\begin{aligned}
b_{\epsilon} &= \frac{1}{4}\rho c n r_{\Delta}^{2}(c_{D_{\eta}} - c_{L_{\eta}}),\\
b_{\epsilon,\sigma} &= \frac{1}{6}\rho c n r_{\Delta}^{3}\omega_{k}(c_{L_{\eta}} - 2c_{D_{\eta}}),\\
b_{\sigma} &= \frac{1}{8}\rho c n r_{\Delta}^{4}\omega_{k}^{2}c_{D_{\eta}},\\
b_{\omega} &= \frac{1}{4}\rho c n r_{\Delta}^{4}\omega_{k}^{2}c_{D_{o}}.
\end{aligned}
\tag{3.37}
$$

Equation (3.34) and (3.36) are in agreement with the work in [41]. The measurable climb velocity of each rotor is a function of the body-frame velocity parallel to ${}^{B}\mathcal{Z}$, and the rotational velocity ${}^{B}\mathbf{\Omega}$,

$$
\epsilon_{c} = \begin{bmatrix} \epsilon_{c_1} \\ \epsilon_{c_2} \\ \epsilon_{c_3} \\ \epsilon_{c_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} {}^{B}\mathbf{v} + \begin{bmatrix} 0 & -l & 0 \\ l & 0 & 0 \\ 0 & l & 0 \\ -l & 0 & 0 \end{bmatrix} {}^{B}\mathbf{\Omega},
\tag{3.38}
$$

where l is the rotor arm length, measured from the rotor centre to the quadrotor centre of mass, and ${}^{B}\mathbf{v}$ is the body-frame translational velocity vector. Incorporating the expression for induced velocity from (3.8), (3.34) can be expanded as

$$
f_{k} = a_{\sigma}\sigma_{k} - a_{\epsilon}\frac{\sqrt{\epsilon_{c_k}^{2} + k_{f}f_{k}} + \epsilon_{c_k}}{2},
\tag{3.39}
$$

where $k_{f} = 2/(\rho\pi r_{\Delta}^{2})$. Finally, solving for f_{k} yields

$$
f_{k} = a_{1}\sigma_{k} + a_{2} \pm a_{3}\sqrt{a_{4}\sigma_{k} + a_{5}},
\tag{3.40}
$$

where

$$
\begin{aligned}
a_{1} &= a_{\sigma} = \frac{1}{6}\rho c n c_{L_{\eta}}\omega_{k}^{2}r_{\Delta}^{3},\\
a_{2} &= \frac{1}{2}a_{\epsilon}\epsilon_{c_k} + \frac{1}{8}a_{\epsilon}^{2}k_{f} = \frac{1}{8}\rho c n c_{L_{\eta}}\omega_{k}r_{\Delta}^{2}\epsilon_{c_k} + \frac{1}{64\pi}\rho c^{2}n^{2}c_{L_{\eta}}^{2}\omega_{k}^{2}r_{\Delta}^{2},\\
a_{3} &= \frac{1}{8}a_{\epsilon} = \frac{1}{32}\rho c n c_{L_{\eta}}\omega_{k}r_{\Delta}^{2},\\
a_{4} &= 16a_{\sigma}k_{f} = \frac{16}{3\pi}c n c_{L_{\eta}}\omega_{k}^{2}r_{\Delta},\\
a_{5} &= 16\epsilon_{c_k}^{2} + 8a_{\epsilon}k_{f}\epsilon_{c_k} + a_{\epsilon}^{2}k_{f}^{2} = 16\epsilon_{c_k}^{2} + \frac{8}{\rho\pi^{2}r_{\Delta}^{2}}c n c_{L_{\eta}}\omega_{k}\epsilon_{c_k} + \frac{1}{4\pi^{2}}c^{2}n^{2}c_{L_{\eta}}^{2}\omega_{k}^{2}.
\end{aligned}
\tag{3.41}
$$

With reference to Section 3.3.1, the two roots in (3.40) stem from the induced velocity's arbitrary direction (relative to the produced thrust). As we are only interested in the case when the induced velocity is anti-parallel to the produced rotor thrust, the force equation in (3.40) can be reposed as

$$f_k = \begin{cases} a_1\sigma_k + a_2 - a_3\sqrt{a_4|\sigma_k| + a_5}, & \sigma_k \geq 0 \\ a_1\sigma_k - a_2 + a_3\sqrt{a_4|\sigma_k| + a_5}, & \sigma_k < 0 \end{cases}. \tag{3.42}$$

Making use of (3.42) and (3.34), the torque equation in (3.36) can be similarly resolved, but is cumbersome, and will be omitted for the reader's sake. The thrust equation in (3.42) is used both to determine appropriate components (e.g. rotor blade dimensions and motors), as well as operating conditions (e.g. rotor speeds and blade angles). Additionally, the nonlinear thrust and torque behaviour can be incorporated into the simulation environment from 2.5.

3.4 Generalised rigid-body dynamics

Consulting Section 3.3, the compact force and torque equations for rotor k are

$$f_k = f(\sigma_k, \omega_k, \epsilon_{c_k}), \tag{3.43}$$

$$\tau_k = \tau(\sigma_k, \omega_k, \epsilon_{c_k}), \tag{3.44}$$

where f_k and τ_k act parallel to ${}^B\mathcal{Z}$. It is assumed that there is no dissymmetry of lift and misalignment between the rotor axes.

3.4.1 Translational dynamics

The quadcopter's collective thrust vector acts parallel to ${}^B\mathcal{Z}$, and is described by

$${}^B\mathbf{f}_\Sigma = \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^{4} f_k \end{bmatrix}. \tag{3.45}$$

The inertial-frame gravitational acceleration vector,

$${}^W\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \tag{3.46}$$

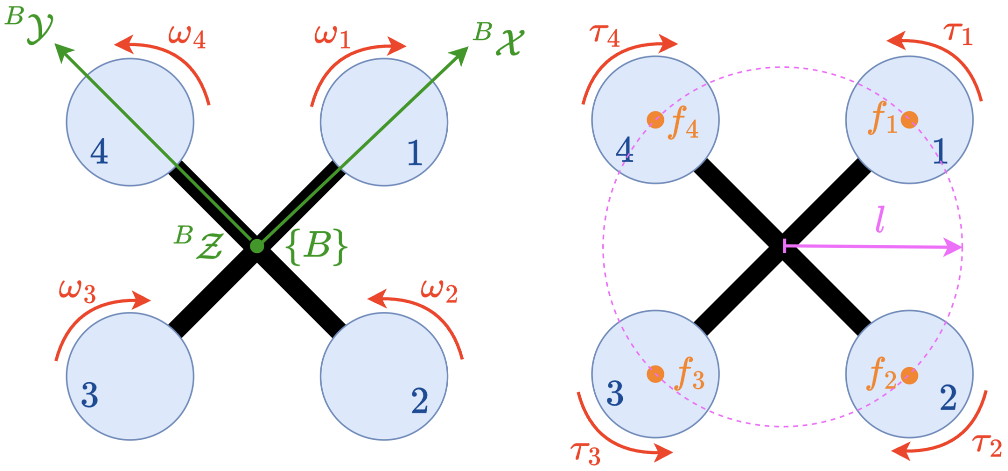


Figure 3.8: Diagram indicating the rotor velocities, thrusts, and torques, as described in the body-frame. The rotor thrust lever arm, l, is also shown.

is aligned with the inertial-frame z-axis, ${}^{W}\mathcal{Z}$, and is modelled as constant within the relatively small flying area. The net force of the quadcopter, described in the inertial-frame, is made up of the body-frame thrust (mapped to the inertial-frame), the gravitational force vector, and some arbitrary inertial-frame disturbance vector,

$$m{}^{W}\ddot{\mathbf{p}} = {}^{W}\mathbf{R}_{B}{}^{B}\mathbf{f}_{\Sigma} + m{}^{W}\mathbf{g} + {}^{W}\mathbf{d}, \tag{3.47}$$

where ${}^{W}\mathbf{d}$ contains unmodelled dynamic effects in the inertial-frame, such as aerodynamic drag on the quadcopter frame and rotor misalignment, and ${}^{W}\ddot{\mathbf{p}}$ is the translational acceleration of the quadcopter centre of mass, described in the inertial-frame. Controllability of the translational dynamics is therefore facilitated by a combination of the direction and magnitude of the thrust vector in the inertial-frame, which is expressed as

$${}^{W}\mathbf{f}_{\Sigma} = {}^{W}\mathbf{R}_{B}{}^{B}\mathbf{f}_{\Sigma}. \tag{3.48}$$

Note that (3.48) is a function of the orientation and body-frame thrust quantity. These two quantities therefore serve as practical control actions for the position control system. For more information, see Section 5.6.

3.4.2 Rotational dynamics

The quadcopter is designed to have a symmetric structure about the ${}^{B}\mathcal{X}$ and ${}^{B}\mathcal{Y}$ axes, with the four identical rotor arms located within the ${}^{B}\mathcal{X}$-${}^{B}\mathcal{Y}$ plane. Consulting Figure 3.8, the commanded yaw torque vector, lying parallel to ${}^{B}\mathcal{Z}$, is the sum of each individual rotor torque

$$m_z = \tau_1 - \tau_2 + \tau_3 - \tau_4 = \sum_{k=1}^{4} (-1)^{k+1}\tau_k, \tag{3.49}$$

where rotor torque vector, τ_k, is anti-parallel to the corresponding rotor velocity vector, ω_k. The commanded roll and pitch torques are proportional to the differential rotor thrusts about the quadcopter centre of mass,

$$\begin{bmatrix} m_x \\ m_y \end{bmatrix} = l \begin{bmatrix} f_4 - f_2 \\ f_3 - f_1 \end{bmatrix}, \tag{3.50}$$

where l is the distance from the rotor axis centre to the quadrotor centre, as shown in Figure 3.8. The full commanded torque vector follows as

$${}^{B}\mathbf{m} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} l(f_4 - f_2) \\ l(f_3 - f_1) \\ \sum_{k=1}^{4} (-1)^{k+1} \tau_k \end{bmatrix}. \tag{3.51}$$

The gyroscopic moment acting on the quadcopter is the cross product of the rigid-body rotational rate, ${}^{B}\mathbf{\Omega}$, with the net angular momentum of the motor-rotor subsystem (acting solely about the ${}^{B}\mathcal{Z}$ axis),

$${}^{B}\boldsymbol{\tau}_g = {}^{B}\mathbf{\Omega} \times (J_r + J_m) \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^{4} (-1)^{k} \omega_k \end{bmatrix} = \begin{bmatrix} (J_r + J_m)\Omega_y \sum_{k=1}^{4} (-1)^{k} \omega_k \\ (J_r + J_m)\Omega_x \sum_{k=1}^{4} (-1)^{k+1} \omega_k \\ 0 \end{bmatrix}, \tag{3.52}$$

where J_m is the combined inertia from the motor outrunner and rotor hub, and J_r is the two-bladed rotor inertia. The centripetal moment acting on the quadcopter is the cross product of the rigid-body rotational rate with the rigid-body angular momentum, which is described compactly as ${}^{B}\mathbf{\Omega} \times \mathbf{J}{}^{B}\mathbf{\Omega}$. Rotational acceleration of the rotors also causes a reactive torque on the quadcopter and can be described by

$${}^{B}\boldsymbol{\tau}_r = (J_r + J_m) \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^{4} (-1)^{k} \dot{\omega}_k \end{bmatrix}. \tag{3.53}$$

The net torque of the robot is proportional to the generated torque minus the centripetal, gyroscopic, and reactive moments

$$\mathbf{J}{}^{B}\dot{\mathbf{\Omega}} = {}^{B}\mathbf{m} - {}^{B}\mathbf{\Omega} \times \mathbf{J}{}^{B}\mathbf{\Omega} - {}^{B}\boldsymbol{\tau}_g - {}^{B}\boldsymbol{\tau}_r - {}^{B}\boldsymbol{\tau}_d, \tag{3.54}$$

where ${}^B\boldsymbol{\tau}_d$ is an unmeasurable disturbance torque (such as a wind gust), accounting for unmodelled dynamics, such as rotor axis misalignment. The mass moment of inertia matrix, $\mathbf{J}$, is estimated using CAD software to be

$$\mathbf{J} = \begin{bmatrix} 3.02 & 0.076 & 0.06 \\ 0.071 & 2.95 & 0.072 \\ 0.099 & 0.098 & 5.290 \end{bmatrix} \times 10^{-3}\ \text{kgm}^2, \tag{3.55}$$

with an estimated uncertainty of ±15%.

3.5 Nominal operating conditions

3.5.1 Constant motor speed operation

With reference to Section 3.3, the thrust and torque behaviour of each rotor is a function of the corresponding rotor speed (among other variables). Additionally, when the four rotor speeds are not equivalent in magnitude, cross coupling occurs in the roll-pitch channels, as seen in (3.52). While this coupling is inconsequential at near-hover conditions (when ${}^B\boldsymbol{\Omega} \approx 0$), highly dynamic multi-channel manoeuvres will result in dominant coupling, which is likely to degrade tracking performance. Given that the generalised variable-pitch quadcopter platform introduced in Section 2.2 is over-actuated (it contains redundant control actuation), the design choice was made to regulate each motor at a constant speed, namely $|\omega_1| = |\omega_2| = |\omega_3| = |\omega_4| = |\omega_0|$. Details of the rotor speed regulation design and implementation can be found in Section 5.2.2. Assuming one can regulate $|\omega_k| = |\omega_0|,\ \forall k \in \{1,2,3,4\}$ with sufficient precision under variable load torques (as a result of blade angle changes), the attitude dynamics in (3.54) simplify to

$$\mathbf{J}{}^B\dot{\boldsymbol{\Omega}} = {}^B\mathbf{m} - {}^B\boldsymbol{\Omega} \times \mathbf{J}{}^B\boldsymbol{\Omega} - {}^B\boldsymbol{\tau}_d. \tag{3.56}$$

In other words, ${}^B\boldsymbol{\tau}_g = {}^B\boldsymbol{\tau}_r = 0$. Note that this model simplification is not possible for speed-controlled quadcopters when operating outside of near-hover conditions.

3.5.2 Hover conditions

Given the choice of regulating all rotors at a constant speed, an appropriate rotor speed needs to be selected. With reference to (3.40), a higher rotor speed allows for smaller blade angle changes, thereby increasing the effective gain, and should be chosen such that there is sufficient blade actuation available (in terms of the reference trajectories). Notably, the maximum rotor speed and blade angle combination is limited by the motor current rating (see Table 3.5). Additionally, the frequency of oscillation of the rotor blades should not coincide with resonant modes of the quadcopter platform, as the on-board state estimator and controllers require high fidelity, low-noise information from the strap-down IMU module. Given the aforementioned requirements, a nominal rotor speed, ω_0, was selected, based on iterative testing, which gives a sufficiently high thrust-to-weight

ω_0	f_0	ϵ_{c0}
600 rad/s	1.55 N	0 m/s

Table 3.2: Nominal hover conditions for each rotor.

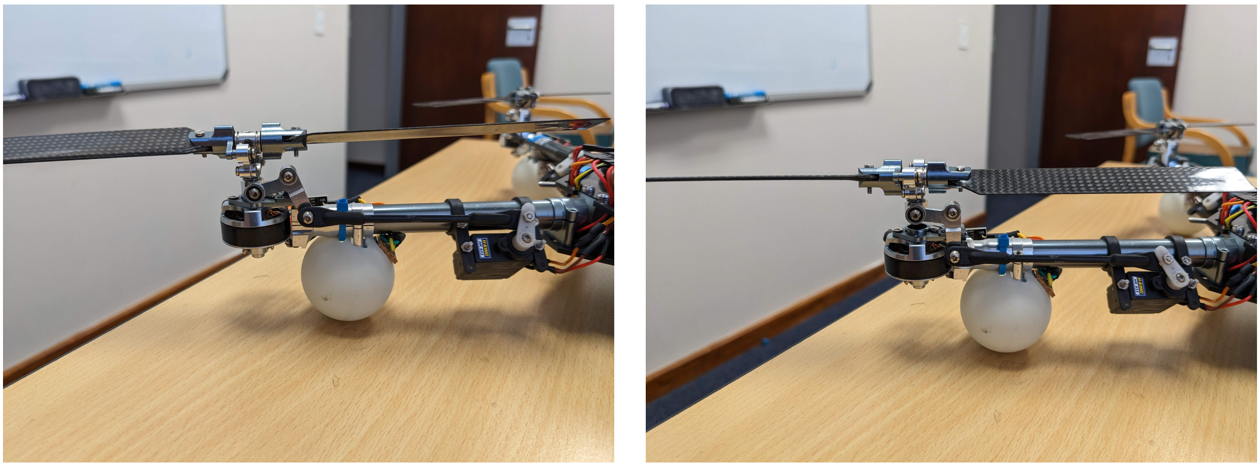

(a) (b)

Figure 3.9: Two examples of how the blade pitch actuation results from servomotor actuation via the push rods and levers.

ratio (> 2), whilst not exciting the natural frequency of the quadcopter body. This choice also ties in with the motor and rotor blade selection from Section 2.2.1. The nominal thrust required by rotor k to maintain hover (outside the ground effect region) is $f_0 = mg/4$. During hover conditions, the climb velocity of each rotor follows as $\epsilon_{c_k} = \epsilon_{c_0} = 0,\ \forall k \in \{1, 2, 3, 4\}$. Table 3.2 summarises the nominal hover conditions of the variable-pitch quadcopter.

3.6 Servomotor-rotor model

As shown in Figure 3.9, each rotor arm makes use of a servomotor to actuate the rotor blade angles. Transmission from the servomotor to propellers is achieved by using a combination of a servomotor horn and push rod, which interfaces with the modified tail rotor unit. This Section analyses the static and dynamic nature of the servomotor-rotor system.

3.6.1 PWM communication

The digital controllers within the servomotors expect a PWM "on time" between 1 ms and 2 ms, whereby this temporal range maps to the corresponding servomotor angular range of approximately $[-90, 90]$ degrees. The PWM channels have a 16-bit resolution (0 – 65535), with a sampling frequency of 200 Hz (equivalent to a sampling period of 5 ms). By extension a 1 ms and 2 ms PWM "on time" corresponds with the 16-bit values of 13105 and 26214, respectively. This mapping is used to set the desired servomotor angles in code, as well as to

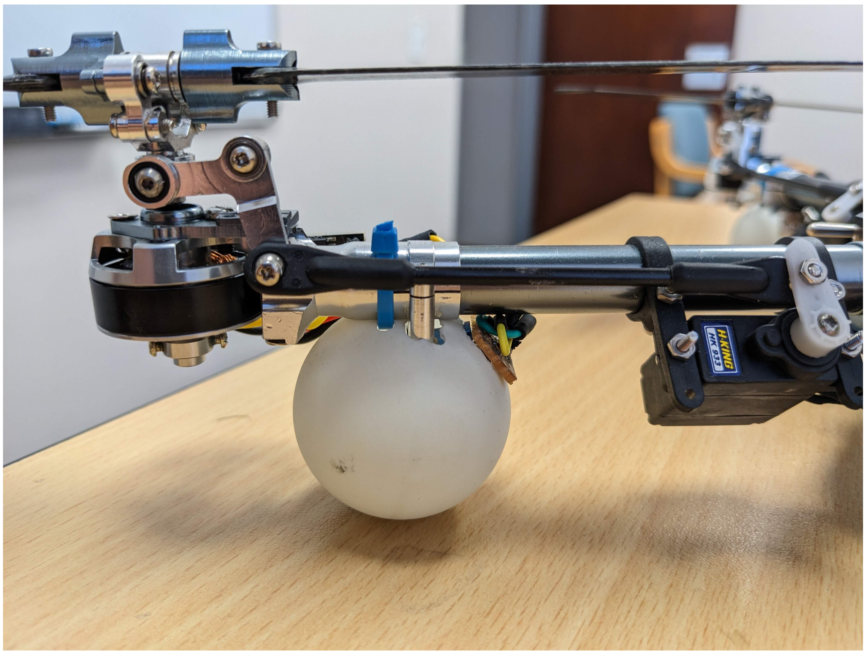


Figure 3.10: Close up view of servomotor-rotor blade actuation system.

protect against commanding an infeasible servomotor angle — one that is outside the range of $[13105, 26214]$.

3.6.2 Static mapping

The static relationship between the servomotor angle and blade angle provides some indication of the DC gain in the servomotor-rotor system. Specifically, the servomotor angle, by virtue of its internal feedback mechanism, is able to achieve any angle within its operating region (up to some finite precision and deadband), and acts as one of the available control actions on the quadcopter. The transmission from servomotor angle to blade angle is fundamentally mechanical, and is prone to real-world effects, such as gear backlash, out-of-play motion, and other types of relative "play" in the system. However, the analysis of the static mapping begins with a theoretical mathematical abstraction of the servomotor-rotor subsystem. Referring to Figure 3.10, one can algebraically relate the servomotor angle, α_k, and blade angle, θ_k, of rotor arm k for sufficiently small angle changes. Small angle operation is a plausible assumption as most aerofoils possess critical (stall) angles of attack that do not exceed 15-18° [51]. Note that while the servomotor angle is internally regulated, and measurable (if the servomotor potentiometer wiper voltage were to be broken out), the rotor blade angles are not. The small-angle mathematical relationship between servomotor angle α_k and blade angle σ_k is depicted in Figure 3.11, with the

r_1	r_2	r_3	r_4	k_σ
9 mm	17 mm	16 mm	9 mm	0.941

Table 3.3: Radial distances used in static servomotor to blade angle mapping

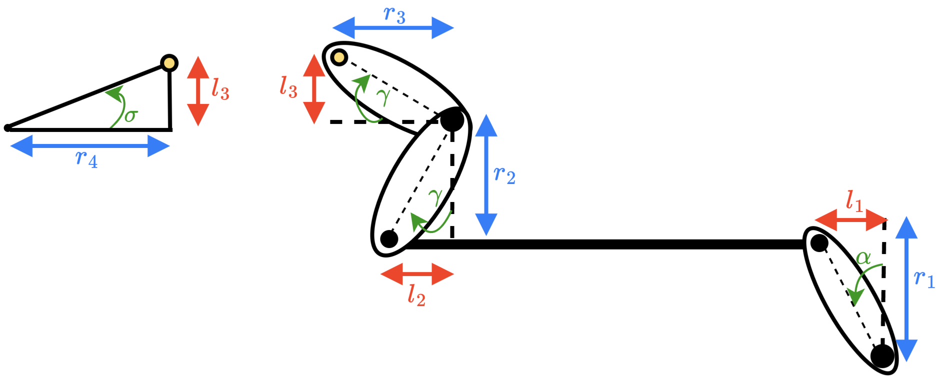


Figure 3.11: Diagram describing static mathematical relationship between servomotor angle and blade angle. Nonlinear effects, such as backlash, are omitted.

accompanying trigonometric relationships

$$\begin{aligned} \sigma &= \tan^{-1}\frac{l_3}{r_4}, \\ l_3 &= r_3 \tan\gamma, \\ \gamma &= \tan^{-1}\frac{l_2}{r_2}, \\ l_2 &= l_1, \\ l_1 &= r_1 \tan\alpha. \end{aligned} \tag{3.57}$$

Note that the common hinge point results in a common rotation angle, γ, when determining l_2 and l_3. Combining all the equations in (3.57) results in the following input-output mapping between the servomotor angle and blade angle,

$$\begin{aligned} \sigma_k &= \tan^{-1}\left(\frac{r_1 r_3}{r_2 r_4}\tan(\alpha_k)\right), \\ &= \tan^{-1}\left(k_\sigma \tan(\alpha_k)\right), \end{aligned} \tag{3.58}$$

with the radial parameters identified in Table 3.3, and $k_\sigma = \frac{r_1 r_3}{r_2 r_4}$. The mapping in (3.58) appears to be nonlinear at first glance. However, visualisation of the relationship in Figure 3.12 infers that the mapping behaves approximately linearly for the operating region of interest ($|\sigma_k| < 18°$) when using the parameters in Table 3.3. Following from this, the aforementioned relationship is suitably approximated as

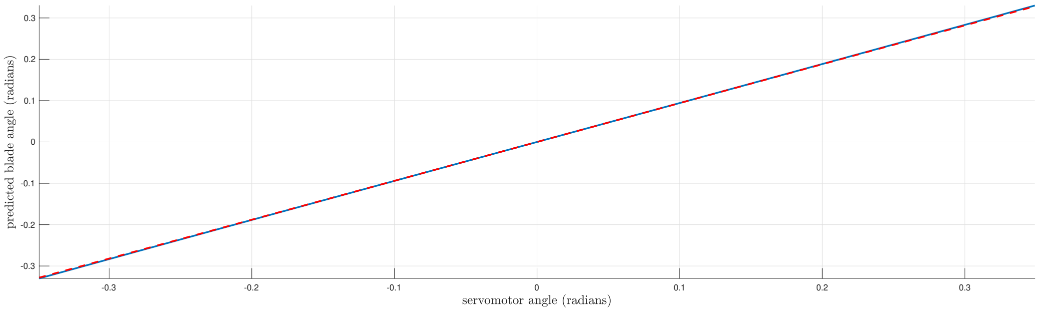


Figure 3.12: Predicted static servomotor to blade angle mapping from (3.59) is shown in blue, and the linearised mapping from (3.59) is overlaid in red.

$$\sigma_k = \tan^{-1}\left(k_\sigma \tan(\alpha_k)\right) \approx k_\sigma \alpha_k. \tag{3.59}$$

It is important to emphasize here that the quasi-direct behaviour of the physical actuation method means that this relationship may not be consistent (repeatable), and therefore cannot be relied upon in general. In addition, because of various forms of play and backlash (such as from the servomotor gears, servomotor horns, rotor bearings), this mapping is uncertain, and may possess other nonlinear characteristics.

3.6.3 Transfer function model

The digital servomotors are internally regulated and have manufacturer-specified performance characteristics as shown in Table 3.4. Frictional effects and aerodynamic hinge torque loading on the rotor blades is expected to change some of the off-the-shelf specifications, such as the slew rate. As mentioned in Section 2.2.1, each servomotor receives commands in the form of a 200 Hz PWM signal. This means that the worst-case delay of the servomotor is $t_\sigma = 5$ ms. Additionally, the rate-limited system will only behave linearly for small angle changes. Given the linearised relationship from (3.59), the blade angle dynamics is provisionally modelled as a second-order underdamped system [52],

Operating current	Slew rate	Stall torque	Deadband	Mass
150 mA	10.47 rad/s	2.0 kg.cm	$\leq 4\ \mu$s	12 g

Table 3.4: Servomotor characteristics as provided by manufacturer.

$$P_\sigma(s) = \frac{k_\sigma \omega_\sigma^2}{s^2 + 2\zeta_\sigma \omega_\sigma s + \omega_\sigma^2} e^{-s t_\sigma}, \tag{3.60}$$

where ω_σ is the natural frequency, ζ_σ is the damping ratio, and $t_\sigma \in [0, 0.005]$ seconds is the dead-time. The lower bound of $\underline{t}_\sigma = 0$ is not feasible in general, as this would require infinitely fast sampling time. Additionally, the worst-case dead-time of $\bar{t}_\sigma = 0.005$ corresponds to the servomotor's ADC taking the full period of the PWM wave to determine the duty cycle of the signal. While both bounds are extreme in the sense that they are not likely to match the true behaviour of the servomotor angle response, they serve as initial bounding values that can be used in the system identification stage (see Section 3.8 for more detail). Note that the plant input to (3.60) is the desired servomotor angle, nominally described in radians. Equation (3.60) is useful in terms of understanding the transfer behaviour from a servomotor command signal and blade angle. However, because the blade angle is not easily measured in a noninvasive means , it is difficult to confirm this behaviour. As the primary function of the servomotor actuation is to adjust the rotor thrust, one can instead use the thrust test rig, introduced in Section 2.6, in order to find the transfer behaviour from the servomotor command to the measured thrust. The main benefit with this approach is that additional aerodynamic effects related to the thrust production will be inherently captured in the system characterisation process, such as the time constant related to the induced velocity airflow. Details of the thrust and torque system identification can be found in Section 3.8.

3.7 Rotor speed model

3.7.1 PWM communication

Similar to digital servomotors, the motor ESCS expect a PWM "on time" between 1 ms and 2 ms, whereby this temporal range (1 – 2 ms) maps to the equivalent voltage applied to the motor. The PWM channels have a 16-bit resolution (0 – 65535), with a sampling frequency of 200 Hz (equivalent to a sampling period of 5 ms). By extension a 1 ms and 2 ms PWM "on time" corresponds with the 16-bit values of 13105 and 26214, respectively. This mapping is used to set the desired motor voltage in code, as well as to protect against commanding an infeasible motor voltage — one that is outside the range of $[13105, 26214]$.

3.7.2 Nonlinear model

The motor-rotor speed dynamics of rotor k can be suitably adapted from standard, field-controlled DC motor models [53] as

$$\begin{aligned}(J_m + J_r)\,\dot{\omega}_k =& k_\phi \left[\frac{v_k - k_\phi \omega_k}{r_\phi} - i_{nl_k}\right] - \tau_k, \\ =& k_\phi \left[i_k - i_{nl_k}\right] - \tau_k,\end{aligned} \tag{3.61}$$

where J_m is the combined rotational inertia of the motor outrunner and rotor hub, J_r is the two-bladed rotor inertia, k_ϕ is the per-phase motor torque constant, r_ϕ is the per-phase resistance, v_k is the equivalent DC voltage command to ESC k,

$$i_k = \frac{v_k - k_\phi \omega_k}{r_\phi} \tag{3.62}$$

is the current of motor k, and i_{nl_k} is the no-load motor current. Additionally, frictional torque effects have been omitted, as their contribution is negligible relative to the aerodynamic drag. The motor parameters provided by the manufacturer are summarised in Table 3.5. Note that the only measurable variables in (3.61) are ω_k, i_k, and v_k. Equation (3.62) suggests that the motor current can be estimated without being directly measured. However,

v_{max}	i_{max}	i_{nl}	k_ϕ	r_ϕ
12 V	9 A	0.5 A	0.006 Nm/A	0.18 Ω

Table 3.5: Summary of motor parameters as provided by manufacturer.

r_ϕ is expected to vary during flight as a result of temperature changes. Additionally, the no-load motor current, i_{nl_k}, depends on the rotor speed.

3.7.3 Linearised model

With reference to (3.36), the load torque experienced by rotor k, τ_k, is a function of the blade angle, motor speed, and climb velocity. This infers that the expanded rotor speed dynamics are inherently nonlinear, and coupled to the servomotor-rotor system in Section 3.6. As such, the nonlinear dynamics of the motor-rotor system in (3.61) are difficult to assess using traditional system identification methods. However, given the design choice that the rotor speeds are to be regulated at a constant speed, it is appropriate to linearise the dynamics in (3.61) at the nominal hover conditions, which results in

$$\begin{aligned}(J_r + J_m)\Delta\dot{\omega}_k &= -(b_{\Delta\omega} + \frac{k_\phi^2}{r_\phi})\Delta\omega_k + \frac{k_\phi}{r_\phi}\Delta v_k - b_{\Delta\sigma}\Delta\sigma_k,\\ &= -(b_{\Delta\omega} + \frac{k_\phi^2}{r_\phi})\Delta\omega_k + \frac{k_\phi}{r_\phi}\Delta v_k - b_{\Delta\sigma}k_\sigma\Delta\alpha_k,\end{aligned} \tag{3.63}$$

where $b_{\Delta\omega} = \left.\frac{\partial\tau_k}{\partial\omega_k}\right|_{(\omega_0,\sigma_0,\epsilon_{c_0})}$, $b_{\Delta\sigma} = \left.\frac{\partial\tau_k}{\partial\sigma_k}\right|_{(\omega_0,\sigma_0,\epsilon_{c_0})}$, and k_σ is from (3.59). Note that $\epsilon_{c_0} = 0$ at nominal hover. Notably, determination of the nominal blade angle at hover, σ_0, is challenging, as the blade angle cannot be measured while the rotor is in operation. Additionally, this value may change between tests or flights, as a result of mechanical wear and backlash in the system.

The input signals to linearised motor-rotor system k are the differential motor voltage, Δv_k, and the differential servomotor angle, $\Delta\alpha_k$. Based on (3.63), the speed response of (linearised) motor k will have the same time constant for both input signals, assuming Δv_k and $\Delta\alpha_k$ are instantaneously achievable. While Δv_k appears to have negligible dynamics (relative to the mechanical motor dynamics), $\Delta\alpha_k$ is subject to non-negligible dynamics, as detailed in Section 3.6.3. Furthermore, the commands to the ESCs and servomotors rely on PWM signals, at rate of 200Hz. As such, Δv_k and $\Delta\alpha_k$ will be subject to transport delays, based on the PWM period. Given the aforementioned uncertainty in the state space representation of the motor behaviour in (3.63), the rotor speed system is instead modelled as a grey box multi-input-single-output (MISO) system,

$$\Delta\omega_k(s) = P_{\omega_{\Delta v}}(s)\Delta v_k^*(s) + P_{\omega_{\Delta\alpha}}(s)\Delta\alpha_k^*(s), \tag{3.64}$$

where $\Delta v_k^*(s)$ and $\Delta\alpha_k^*(s)$ are the differential commands to ESC k and servomotor k, respectively, and $P_{\omega_{\Delta v}}(s)$ and $P_{\omega_{\Delta\alpha}}(s)$ are the corresponding open-loop plants that map the respective commands to the rotor speed output. The block diagram representation of this mapping is shown in Figure 3.13. Given that the micro-controller communicates with the ESCs and servomotors with a 200Hz PWM signal, both responses are expected to have the same worst-case delay of 5 milliseconds.

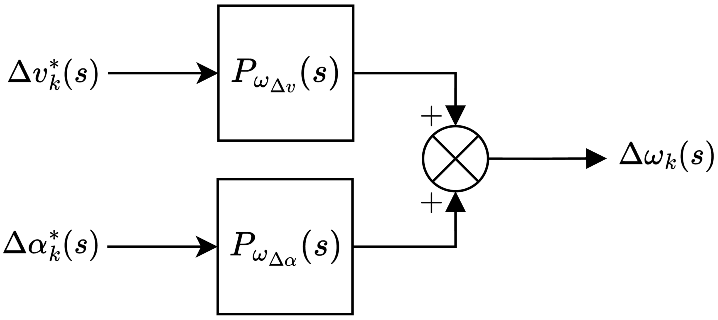


Figure 3.13: Block diagram of linearised open-loop rotor model.

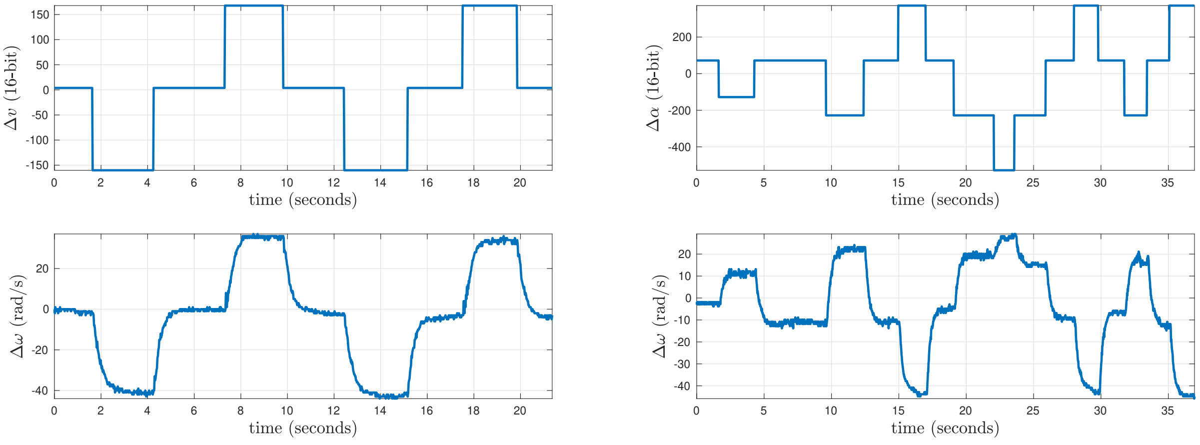
3.7.4 Transfer function model

System identification of the rotor speed parameters is facilitated using the thrust rig introduced in Section 2.6, whereby the entire quadcopter is mounted on the thrust rig. $P_{\omega_{\Delta v}}(s)$, representing the transfer behaviour from $\Delta v_k^*(s)$ to $\Delta\omega_k$, and $P_{\omega_{\Delta\alpha}}(s)$, representing the transfer behaviour from $\Delta\alpha_k^*(s)$ to $\Delta\omega_k$, are identified by first operating rotor k at nominal hover conditions, namely $\omega_k = \omega_0$ and $f_k = f_0 = mg/4$. The rotor speed is measured onboard (see Section 2.2.2), whereas the rotor thrust is measured using the load cell setup on the thrust rig. The rotor angle and open-loop ESC command are adjusted manually using a radio controller (see Section 2.4.3) until the nominal conditions are simultaneously met.

Keeping the blade angle constant ($\Delta\alpha_k^*(s) = 0$), a sequence of steps is then applied to the ESC command in order to determine $P_{\omega_{\Delta v}}(s)$, as shown in Figure 3.14a. The resulting motor speed is measured and logged, along with the applied differential voltage commands. Note that the voltage and ESC command, Δv_k^* and $\Delta\alpha_k^*$, respectively, are measured in their "raw" 16-bit form (as opposed to the corresponding Volt and radian equivalent). Using the MATLAB system identification toolbox [54], a first-pass $P_{\omega_{\Delta v}}(s)$ is estimated as

$$P_{\omega_{\Delta v}}(s) = \frac{0.232}{1+0.3s}e^{-0.004s}, \tag{3.65}$$

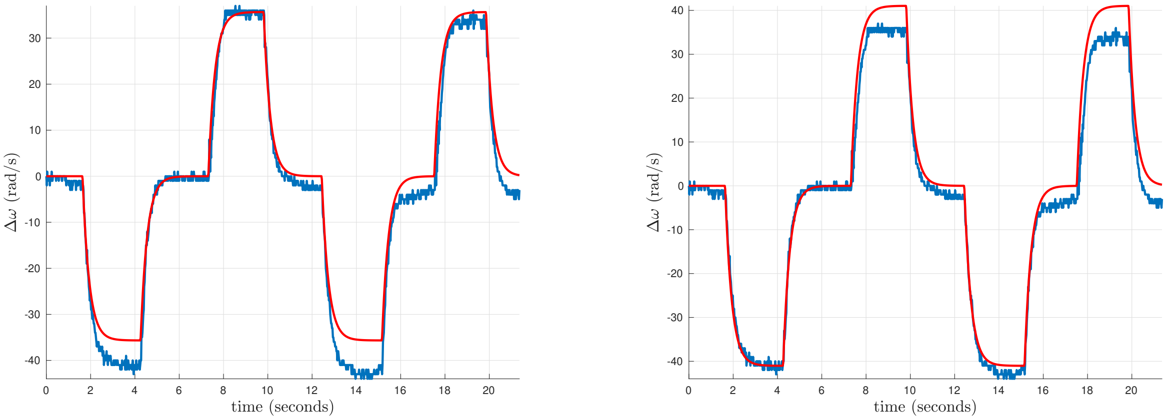
with a 92% goodness of fit, and a mean squared error of 4%. Note that a second-order transfer function model was also considered, and showed comparable goodness of fit. The first-pass parameters are then varied and compared against the measured response to develop a suitable plant set that represents the system uncertainty. The output response of two plant instances (a high and low gain instance) are compared against the measured



(a) $P_{\omega_{\Delta v}}(s)$. (b) $P_{\omega_{\Delta\alpha}}(s)$.

Figure 3.14: Input-output data for differential rotor speed as a result of (a) applying step inputs to the ESC command voltage, and (b) applying step inputs to the servomotor angle.

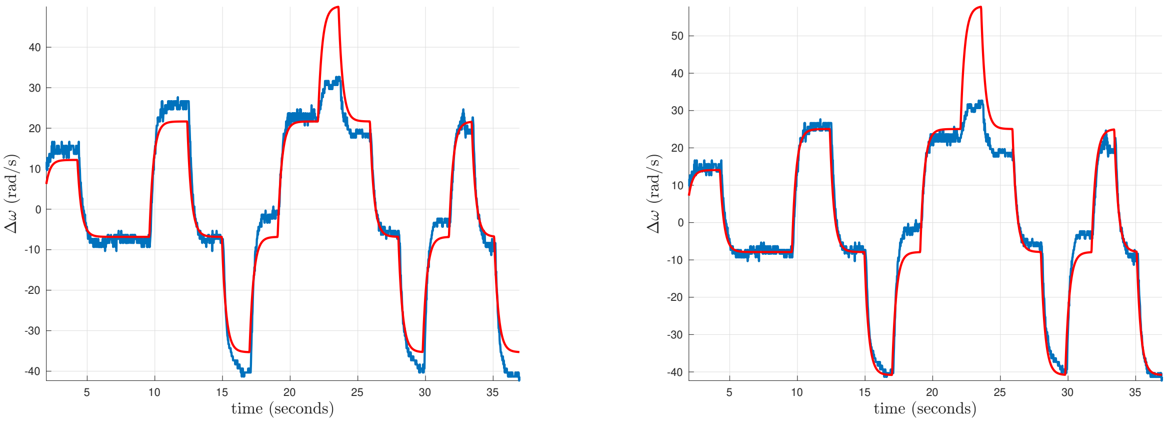
output response in Figure 3.15. The output response exhibits minor nonlinearities, as evidenced by the time-varying differential rotor speed when $\Delta v = 0$. This is likely due to the variation of rotor blade loading as the rotor speed is increased and decreased.



(a) Low-gain plant (b) High-gain plant

Figure 3.15: Measured output response (blue) and estimated output responses (red) to voltage changes for (a) a low-gain and (b) high-gain plant. The full plant set is described in Table 3.6.

$P_{\omega_{\Delta\alpha}}(s)$ is similarly estimated by exciting the servomotor angle of rotor k away from hover conditions, whilst keeping Δv_k constant at the determined nominal condition ($\Delta v_k^*(s) = 0$). The input-output data is shown in Figure 3.14b. As shown in Figure 3.14b, positive changes in the servomotor angle decrease the rotor speed,



(a) Low-gain plant

(b) High-gain plant

Figure 3.16: Measured output response (blue) and estimated output responses (red) to servomotor angle changes for (a) a low-gain and (b) high-gain plant. The full plant set is described in Table 3.6.

and vice versa, which is a result of increasing/decreasing the load torque. As with $P_{\omega_{\Delta v}}(s)$, $P_{\omega_{\Delta\alpha}}(s)$ is first estimated to be

$$P_{\omega_{\Delta\alpha}}(s) = \frac{-0.093}{1+0.3s}e^{-0.005s}, \tag{3.66}$$

with a goodness of fit of 76%, and a mean squared error of 23%. As with $P_{\omega_{\Delta v}}(s)$, a second-order transfer function model was also considered for $P_{\omega_{\Delta\alpha}}(s)$, but the fit improvement was found to be negligible. Following this, the parameters are manually refined in order to develop a suitable plant set. The simulated low- and high-gain plant responses are shown in Figure 3.16. The goodness of fit in Figure 3.16 is noticeably poorer than that of Figure 3.15, which is likely a result of the hysteresis in the blade actuation system (see Section 3.8 for more information).

The resulting differential behaviour of the rotor speed is described by

$$\begin{aligned}\Delta\omega(s) &= P_{\omega_{\Delta v}}(s)\Delta v_k^*(s) + P_{\omega_{\Delta\alpha}}(s)\Delta\alpha_k^*(s),\\ &= \frac{a_{\Delta v}e^{-st_{\Delta v}}}{1+sT_{\Delta v}}\Delta v_k^*(s) + \frac{a_{\Delta\alpha}e^{-st_{\Delta\alpha}}}{1+sT_{\Delta\alpha}}\Delta\alpha_k^*(s),\end{aligned} \tag{3.67}$$

with each parameter, and the corresponding variance, described in Table 3.6. As predicted, $P_{\omega_{\Delta v}}(s)$ and

$a_{\Delta v}$ (rad/s/16-bit)	$T_{\Delta v}$ (s)	$t_{\Delta v}$ (s)	$a_{\Delta\alpha}$ (rad/s/16-bit)	$T_{\Delta\alpha}$ (s)	$t_{\Delta\alpha}$ (s)
0.234 ± 0.016	0.28 ± 0.007	0.0025 ± 0.0025	−0.102 ± 0.007	0.29 ± 0.01	0.005

Table 3.6: Motor-rotor plant parameters with corresponding uncertainty bands.

$P_{\omega_{\Delta\alpha}}(s)$ have comparable time constants, irrespective of the transients in the servomotor system. This is likely because the servomotor time constant is far smaller than that of the unloaded motor time constant, and therefore can be approximated as adhering to time-scale separation.

With reference to Section 3.5.1, all four rotors are required to operate at a constant speed, regardless of torque load changes. The developed model in this Section therefore serves as the starting point for the rotor speed control. For more information, see Section 5.2.2.

3.8 Rotor thrust and torque model

As shown in Section 3.4, the rotor thrust and torque dictate the motion of the quadcopter, both in terms of translation and orientation. As such, a thorough understanding of their static and dynamic nature is required in order to successfully stabilise and control the robotic platform. With reference to Section 3.3, the thrust and torque behaviour are a function of the servomotor angle, rotor speed, and climb velocity:

$$f_k = f(\sigma_k, \omega_k, \epsilon_{c_k}), \tag{3.68}$$

$$\tau_k = \tau(\sigma_k, \omega_k, \epsilon_{c_k}). \tag{3.69}$$

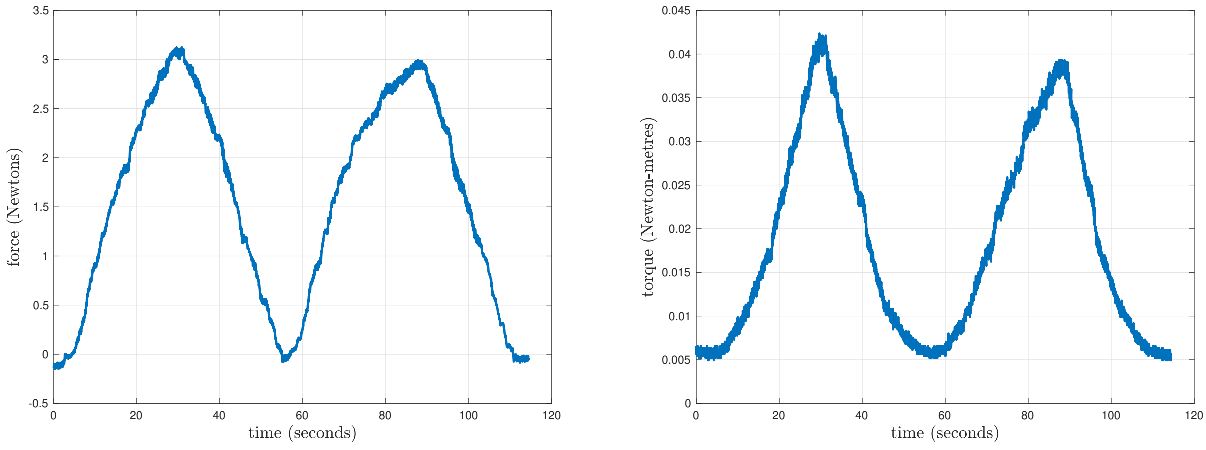
Assuming the rotor speed can be effectively regulated at a constant speed (see Section 5.2.2), and the climb velocity can be measured and accounted for (based on (3.38)), the thrust and torque dynamics will be dictated exclusively by the servomotor-rotor behaviour. However, the fact that the blade angles are not measurable necessitates development of an experimental model that relates the commanded servomotor angle to the corresponding rotor thrust and torque.

3.8.1 Steady-state profile

The thrust test rig is used to develop a static thrust and torque profile that describes the DC relationship between commanded servomotor angle and thrust/torque. The steady-state ($|\omega_k| = |\omega_0|$, $\dot{\sigma}_k = \epsilon_{c_k} = 0$, $\forall k$) mapping from commanded servomotor angle k, α_k, to the thrust, f_k, and torque, τ_k, is obtained by slowly sweeping through the servomotor angle's expected operating region whilst measuring the corresponding thrust and torque. The cut-off frequency of the digital low-pass filter of the load cell rig is set to 6 rad/s. This effectively removes high frequency noise from the amplification stage, without incurring phase lag (as a result of the sufficiently slow angle sweeping). The rotor thrust is directly measured using the load cell setup, whereas the rotor torque is estimated based on (3.61) when in steady-state ($\dot{\omega}_k = 0$), namely

$$\tau_k = k_\phi \left(i_k - i_{nl} \right). \tag{3.70}$$

The servomotor angle operating region is defined based on sweeping back and forward with a rotor thrust of $f_k \in [0\ 2f_0]$. In other words, from no-load conditions to double the per-rotor hover thrust. Figure 3.17 shows the resulting thrust and torque behaviour versus time, and Figure 3.18 shows the corresponding commanded

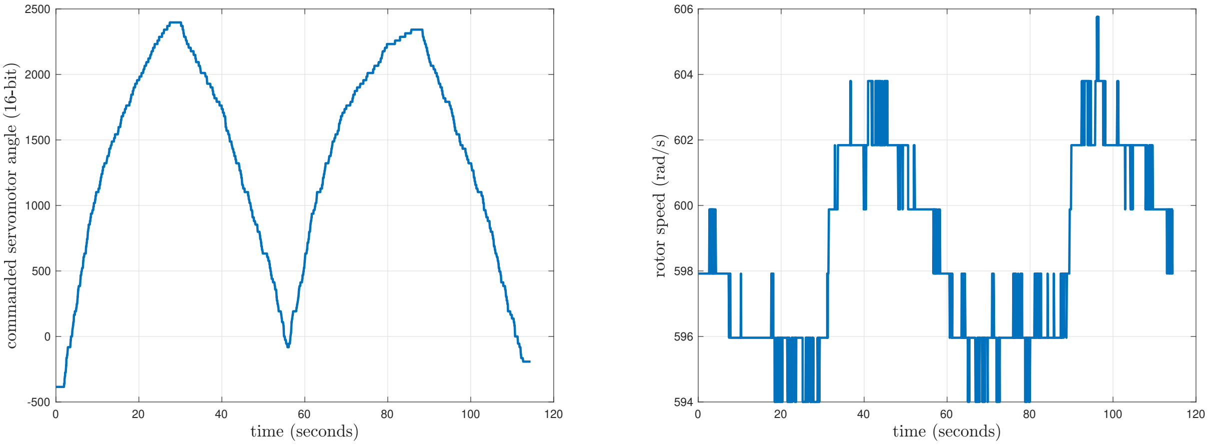


(a) f_k vs. t (b) τ_k vs. t

Figure 3.17: Measured rotor (a) thrust and (b) torque against time for multiple sweeps from $f_k = 0$ to $f_k = 2f_0$.

servomotor angle and rotor speeds from the same test. Note that the servomotor angle is repeatedly slewed in order to gauge the repeatability of the thrust and torque mappings. Figure 3.19 shows the corresponding thrust and torque profiles against the commanded servomotor angle. As evidenced in Figure 3.19, the steady-state mapping between the commanded servomotor angle and thrust/torque is not consistent. This can be attributed to nonlinear effects in the commanded servomotor-rotor subsystem. Specifically, hysteresis from gear backlash and play in the transmission system. As a result, a commanded servomotor angle will result in a corresponding thrust and torque that can have large uncertainty, depending on the particular operating point, with a measured hysteresis band of $h_\alpha = 200$ in 16-bit count. This is problematic for control systems, as this time-dependant uncertainty effectively acts as a plant input disturbance and can also lead to limit cycling and instability. While a variable rotor speed could also cause the inconsistent mapping, consultation of Figure 3.18b shows that the rotor speed is effectively constant during the entire test, with a worst-case change of 3 rad/s (or 0.5% of the nominal rotor speed). Figure 3.20 shows the worst-case varitation of the thrust and torque, as parameterised by the commanded servomotor angle, which is determined by finding the difference between the maximum and minimum thrust/torque for a given servomotor angle. The thrust uncertainty is somewhat constant for commanded servomotor angles greater than 1000, sitting at around 0.45 N, which is approximately 30% of the hover thrust of $f_0 = 1.55$ N. Given that the quadcopter control action comprises the thrust and torque across all four rotors, based on (3.45) and (3.51), the uncertainty band will become more problematic when controlling roll, pitch and yaw. This problem is ultimately addressed using high-bandwidth feedback on the rotor thrust, in order to suppress the low-level rotor thrust and torque uncertainty. Details of the thrust estimation can be found in Section 4.4, and information relating to the thrust control is presented in Section 5.2.3.

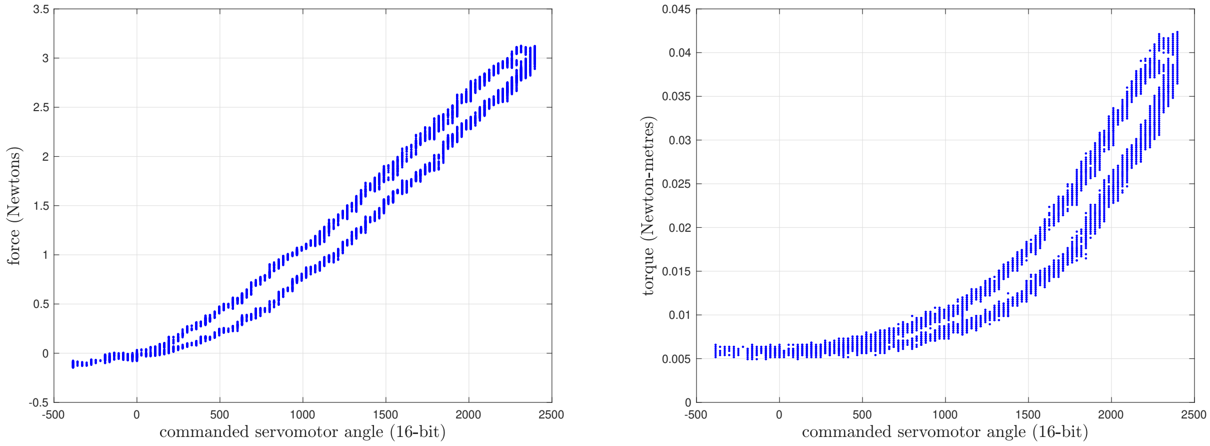
The mapping from α_k to f_k behaves approximately linearly around the hover thrust ($f_0 = 1.55$ N) when the commanded blade angle changes monotonically. With reference to Figure 3.21a, the nominal hover point,



(a) α_k vs. t

(b) ω_k vs. t

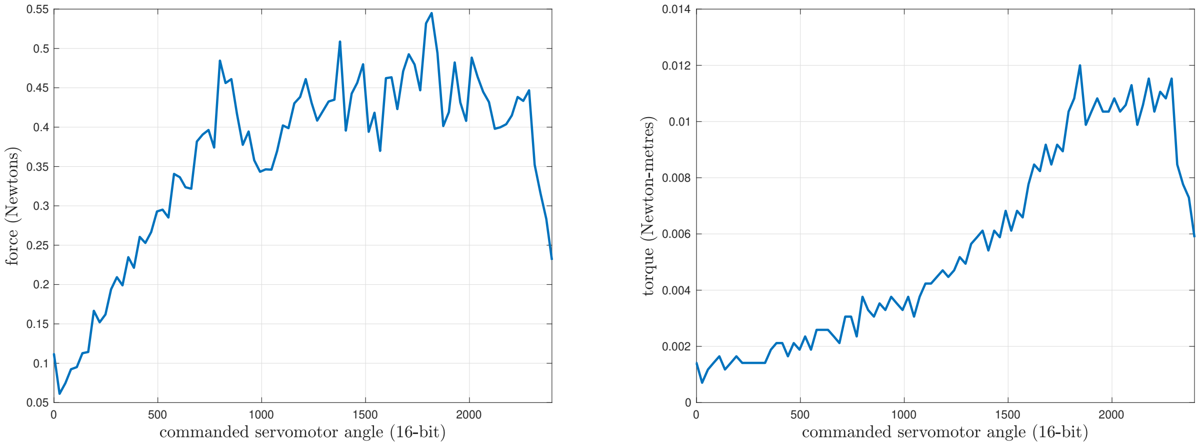
Figure 3.18: (a) commanded servomotor angle and (b) rotor speed against time for multiple sweeps from $f_k = 0$ to $f_k = 2f_0$.



(a) f_k vs. α_k

(b) τ_k vs. α_k

Figure 3.19: Measured rotor (a) thrust and (b) torque plotted against commanded servomotor angle for multiple sweeps from $f_k = 0$ to $f_k = 2f_0$.



(a) f_k (b) τ_k

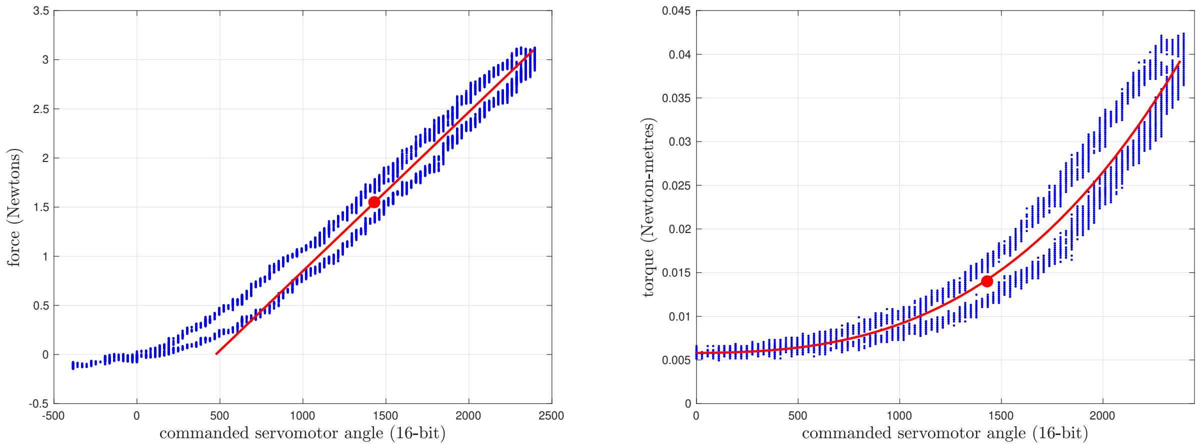
Figure 3.20: Band of (a) thrust and (b) torque uncertainty as a function of the commanded servomotor angle, as a result of the actuation backlash. This variation is calculated by finding the difference between the maximum and minimum thrust/torque for a given commanded servomotor angle.

(f_0, α_0) is identified, followed by fitting a straight line that matches the local linear behaviour near f_0. The resulting fit is used to describe the differential behaviour as

$$\Delta \hat{f}_k = k_f \Delta \alpha_k, \tag{3.71}$$

where $k_f = 0.0016$ is the gradient of the line of best fit through (f_0, α_0). The approximate relationship in (3.71) is notably only valid for sufficiently large commanded servomotor angles. Note that while f_k is found to be constant across all four rotors, the nominal hover servomotor angle, α_0, does vary across rotors. This is attributed to multiple factors, such as the initial rotor configuration, and play in the mechanism. As seen in Figure 3.21a, the relationship does not hold near the zero-lift angle ($\alpha_k \rightarrow 0$). However, the rotor is not expected to operate at/near the zero-lift angle for extended periods of time (likely only during brief, rapid manoeuvres). Finally, the measured steady-state rotor thrust can be compared to the expected rotor thrust from (3.42), which is derived using a combination of blade element theory and momentum theory. Figure 3.22 shows the expected and measured rotor thrust using a lift coefficient constant of $c_{L_\eta} = 4.8$, which closes matches the predicted lift coefficient constant of $c_{L_\eta} = 4.53$ from Section 3.3.2. The comparable profiles suggest that the developed mathematical model of the rotor thrust is valid within the expected operating region and can be used as part of the rotor thrust simulation model.

The steady-state servomotor angle to rotor torque behaviour can also be captured using a line of best fit. Unlike the rotor thrust profile, the rotor torque profile does not exhibit local linear behaviour around the hover thrust operating point. Note that this is an expected result, based on the nonlinear formulation in (3.36). As



(a) f_k vs. α_k (b) τ_k vs. α_k

Figure 3.21: Measured rotor (a) thrust and (b) torque plotted against commanded servomotor angle for multiple sweeps from $f_k = 0$ to $f_k = 2f_0$. The hover thrust operating point and lines of best fit are also shown on the sub-figures, with a determined hover servomotor angle of $\alpha_0 = 1430$.

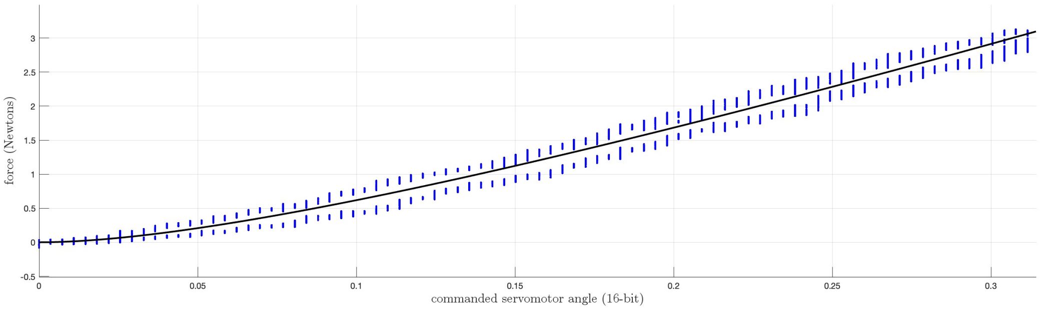


Figure 3.22: Comparison of measured (blue) and expected (black) rotor thrust. The expected rotor thrust is derived using a combination of blade element theory and momentum theory, as detailed in Section 3.3.

shown in Figure 3.21b, the torque profile is suitably captured using the absolute-value cubic polynomial

$$\hat{\tau}_k = \left|a_0 + a_1\alpha_k + a_2\alpha_k^2 + a_3\alpha_k^3\right|, \tag{3.72}$$

where the polynomial coefficients are provided in Table 3.7. Note that the absolute values preserve the stricly-positive torque production, regardless of whether the blade angle becomes negative.

3.8.2 Transfer function model

The steady-state thrust profile developed in the previous section is useful for understanding the DC mapping between α_k and f_k, but the transient behaviour also needs to be captured. The thrust test rig was used to generate step responses of the rotor thrust and torque as a result of instantaneous changes to the commanded

a_0	a_1	a_2	a_3
5.8×10^{-3}	1.376×10^{-7}	1.237×10^{-9}	1.937×10^{-12}

Table 3.7: Cubic polynomial coefficients used to describe steady-state mapping between commanded servomotor angle and rotor torque.

servomotor angle. Step inputs were applied in the vicinity of the hover thrust, f_0, as this is effectively the centre of the thrust's operating region. Starting from near-hover conditions ($f_k \approx f_0$), a series of bipolar step inputs are applied to the commanded servomotor angle. The step responses of the rotor thrust and torque are shown in Figure 3.23 and Figure 3.24, respectively. The corresponding commanded servomotor angle and rotor speed responses are shown in Figure 3.25. Using a similar process as in Section 3.7.4, a first-pass transfer function relating the commanded servomotor angle to thrust is found to be

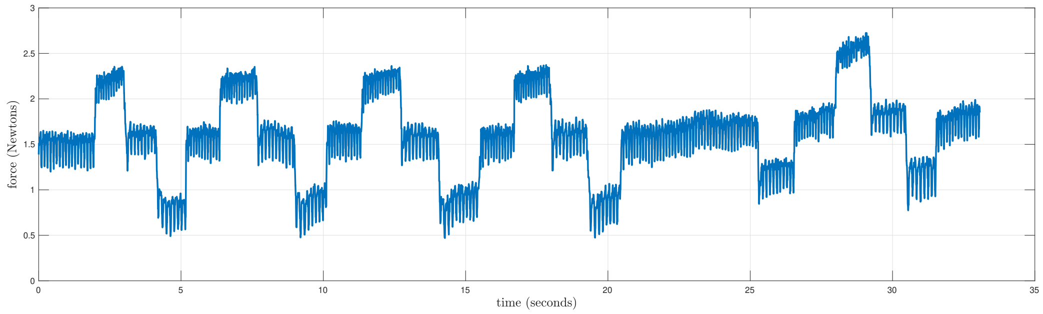


Figure 3.23: Time-domain output response of f_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$.

$$P_f(s) = \frac{a_f}{s^2 + 2\zeta_f \omega_f s + \omega_f^2} e^{-t_f s} = \frac{0.89}{s^2 + 26.49s + 773.5} e^{-0.005}, \tag{3.73}$$

with a damping ratio of $\zeta_f = 0.48$ and a natural frequency of $\omega_f = 27.81$ rad/s. The plant parameters are then adjusted to develop a plant set that suitably captures the gain and phase variations in the actual system. Figure 3.26 and Figure 3.27 compare the estimated thrust response against the measured response for two different

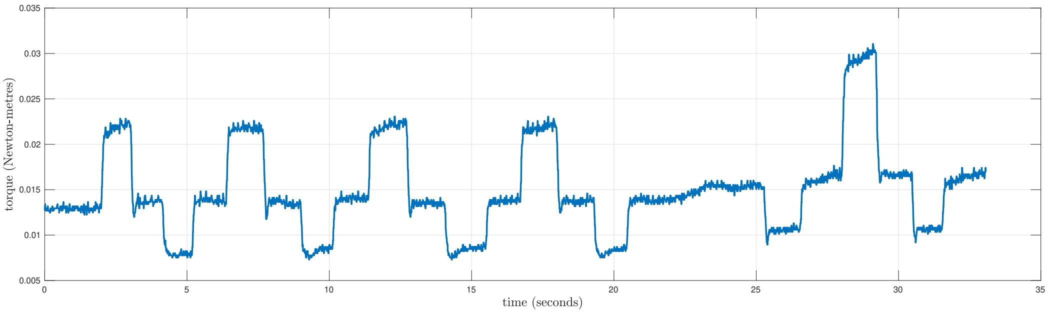
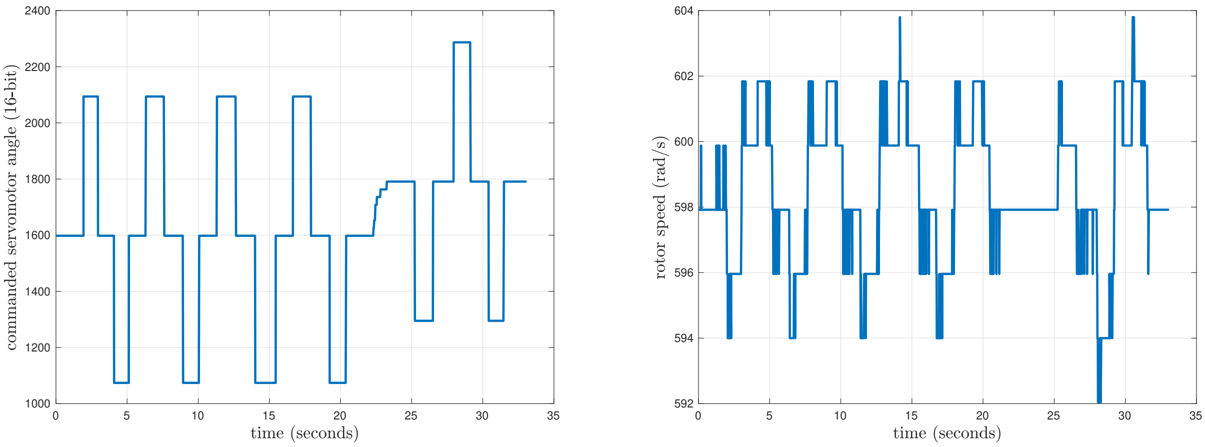


Figure 3.24: Time-domain output response of τ_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$.



(a) α_k vs. t (b) ω_k vs. t

Figure 3.25: (a) commanded servomotor angle and (b) rotor speed against time for multiple step inputs about $f_k \approx f_0$.

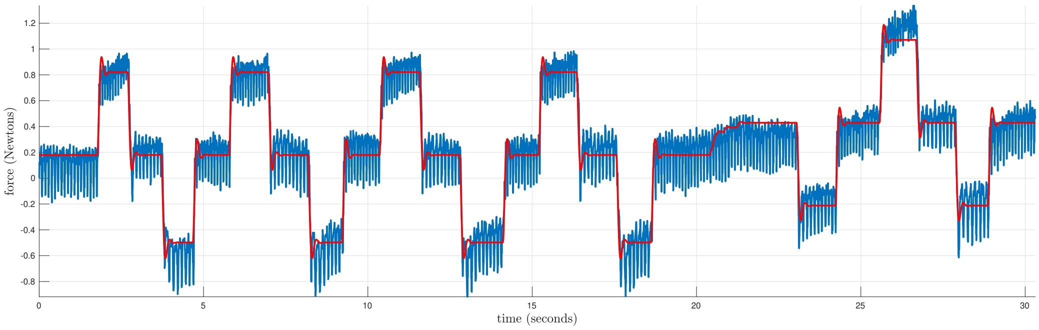


Figure 3.26: Time-domain output response of f_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$ (blue). The estimated thrust response of a plant instance with a low damping ratio (red) is overlaid to compare the transient behaviour.

plant instances. With reference to Figure 3.26 and Figure 3.27, the thrust response is seen to differ depending on the direction of the blade angle change. Specifically, the thrust response appears to be damped when the blade angle increases, whereas an underdamped response is seen when the blade angle change is negative. This is likely due to the aerodynamic loading on the blade surface that results in an asymmetric thrust model that depends on the angular direction. Specifically, when a positive blade angle change occurs, the blade pitch moves in the direction of the applied aerodynamic hinge torque, which maintains blade loading (thereby avoiding any backlash). However, when a negative blade angle change occurs, the blade rotates against the direction of the aerodynamic load. Once the servomotor angle reaches steady-state, the blade returns to being loaded only by aerodynamic hinge torque. As a result, the plant in Figure 3.26 represents positive angular changes, whereas the plant in Figure 3.27 captures negative angular changes. The full plant set is described in Table 3.8. System

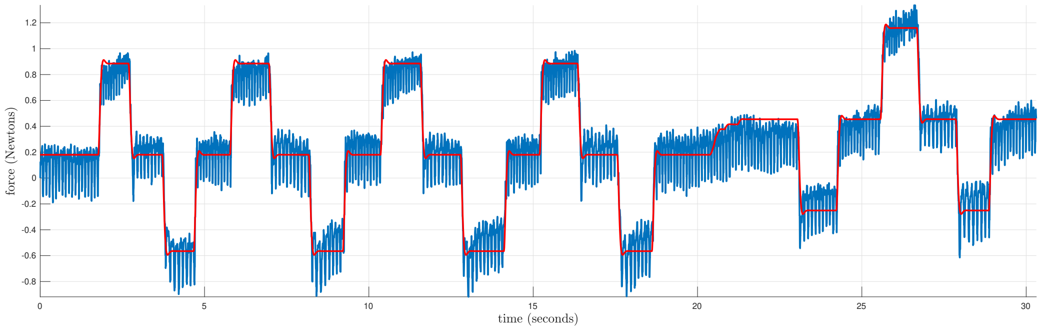


Figure 3.27: Time-domain output response of f_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$ (blue). The estimated thrust response of a plant instance with a high damping ratio (red) is overlaid to compare the transient behaviour.

a_f (N/16-bit)	ζ_f	ω_f (rad/s)	t_f (s)
1 ± 0.2	0.6 ± 0.12	28 ± 5	0.005

Table 3.8: Rotor thrust plant parameters with corresponding uncertainty bands.

identification of the rotor torque model is more challenging, as the local, near-hover mapping from commanded servomotor angle to torque is not linear. However, the transient behaviour of the torque is expected to be similar to that of the thrust model (aside from the nonlinear gain). Following this assumption, the structure of (3.73) is adopted for the torque plant as

$$P_\tau(s) = \frac{a_\tau}{s^2 + 2\zeta_\tau\omega_\tau s + \omega_\tau^2} e^{-t_\tau s}. \tag{3.74}$$

Using the first-pass thrust model from (3.73), and adjusting for the gain differences, the model parameters are then refined based on the torque step response data. The resulting torque plant set is described in Table 3.9, and responses from two of the estimated plant instances are shown in Figure 3.28 and Figure 3.29. As in the case of the rotor thrust model, the torque model is asymmetric, and dependant on the direction of the blade angle change. Figure 3.28 represents one of the plant instances that captures positive blade angle changes, whereas Figure 3.29 represents negative blade angle changes. With reference to Table 3.9, the gain uncertainty of the torque plant model is large relative to that of the thrust plant model in Table 3.8, owing to the fact that the transfer behaviour from commanded servomotor angle to rotor torque is fairly nonlinear. The rotor thrust and torque have comparable natural frequencies and damping ratios, with the rotor torque exhibiting a larger band of expected damping factors. This is somewhat expected, based on the equations developed in Section 3.3, and their common dependence on the rotor speed and servomotor dynamics.

a_τ (Nm/16-bit)	ζ_τ	ω_τ (rad/s)	t_τ (s)
0.01 ± 0.004	0.75 ± 0.25	27.5 ± 4	0.0025 ± 0.0025

Table 3.9: Rotor torque plant parameters with corresponding uncertainty bands.

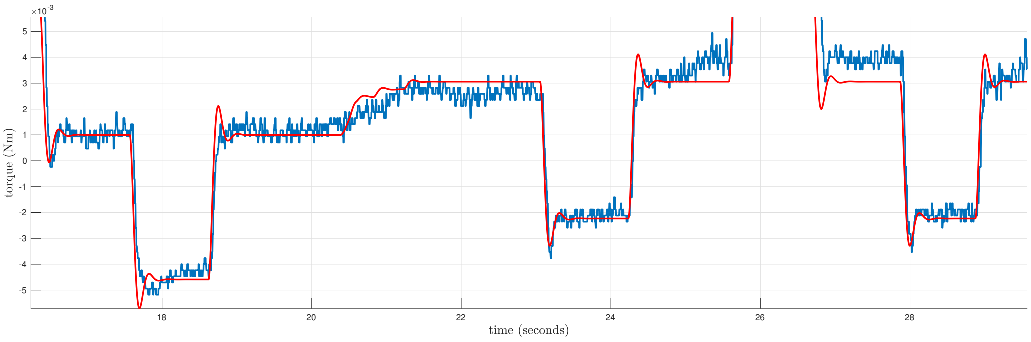


Figure 3.28: Time-domain output response of τ_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$. The estimated torque response of a plant instance with a low damping ratio is overlaid to compare the transient behaviour.

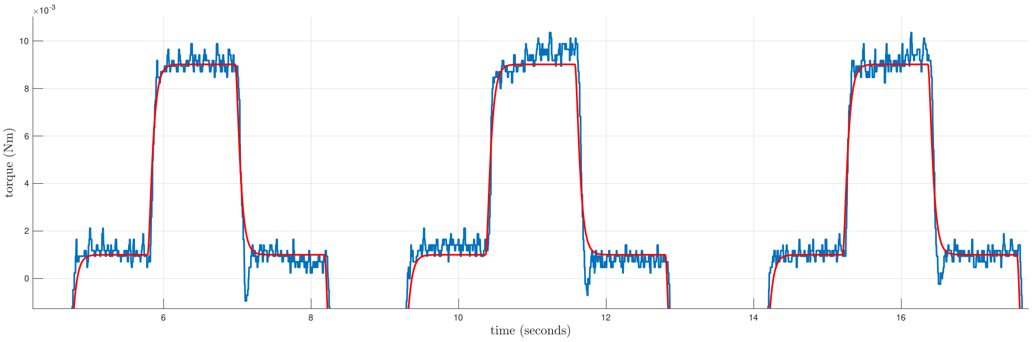


Figure 3.29: Time-domain output response of τ_k as a result of a series of step inputs applied to α_k about $f_k \approx f_0$. The estimated torque response of a plant instance with a high damping ratio is overlaid to compare the transient behaviour.

Chapter 4

State estimation

This chapter is based, in part, on the following first-authored and published paper:
Shutter delay estimation for a low-cost stand-alone visual tracking system [22]

Section 4.1 provides a review of the literature relevant to the state estimation of quadcopters. Section 4.2 details the visual tracking system, which includes the mathematical derivations, hardware considerations, and implementation of the motion capture system. Section 4.3 contains information about the on-board attitude estimation method that is used on the variable-pitch quadcopter to fuse the camera quaternion estimate with information from an onboard gyroscope. Section 4.4 provides a means of estimating the quadcopter rotor thrusts during operation using the measured motor currents.

4.1 Literature review

In the context of estimating the state of unmmanned aerial vehicles (UAVs), many solutions exist with varying accuracies and applicabilities [23]. A common method is to determine the translation and/or attitude using various combinations of on-board sensors, such as GPS modules (mainly outdoors) [55], stereo cameras [56], laser range-finding [57], inertial measurement units [44], and RGB-D camera modules [58]. These solutions commonly have conditional state observability [59] as well as other weaknesses, such as low accuracy, or unbounded drift.

On the other hand, off-board visual tracking solutions can give unconditional state information in a predefined operational volume [60]. Off-the-shelf camera systems are most commonly used to provide high-precision state estimation, and several offerings boast sub-millimetre accuracy at high frame rates [61]. For example, high-end Vicon camera systems can perform feature estimation at over 200 Hz with HD resolution [62]. However, such a set-up can easily cost over $100000, which is not feasible for many applications and research institutes.

Low-cost alternatives to expensive motion capture systems have also been investigated by several authors. The work in [63] uses two cameras, as well as IMU information, to localise a UAV with distinct hue-based features. Similarly, the authors of [64] were able to achieve sub-centimetre accuracy object tracking using PSEye cameras, which included IMU and ultrasonic data fusion. State estimation using static RGB-D cameras has also yielded promising results. In addition to visual 2D information, the feature depth is also estimated using a laser projector and CMOS sensor. These camera modules however generally have a limited depth range and frame-rate [65].

4.2 Visual tracking system

The software and hardware aspects of the motion capture system were briefly introduced in Section 2.3. Working details of the extended Kalman Filter (EKF) and correspondence matching will now be presented.

4.2.1 Camera Model

Generalised camera model

The camera model considers four distinct reference frames per camera, as summarised in Table 4.1 The generalised continuous-time camera model [66] for camera k is

Name	Notation	Location
Body-frame	$\{B\}$	Attached to quadcopter
Inertial-frame	$\{W\}$	Attached to flight test area
Camera-frame k	$\{C_k\}$	Attached to camera k lens
Image-plane k	$\{I_k\}$	Attached to camera k image plane

Table 4.1: Name, notation, and location of principle reference frames used in camera model.

$$^{I_k}w \; {}^{I_k}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = {}^{I_k}\mathbf{C}_W \begin{bmatrix} {}^W\mathbf{b} \\ 1 \end{bmatrix}, \tag{4.1}$$

where ${}^{I_k}\begin{bmatrix} u & v \end{bmatrix}^T$ is the normalised and undistorted measurement of an inertial-frame feature, ${}^W\mathbf{b}$, after projection into frame $\{I_k\}$ (the image plane of camera k). ${}^{I_k}w$ is the feature's unresolved metric depth, described in image plane k, and ${}^{I_k}\mathbf{C}_W$ is the camera projection matrix, specific to camera k, which projects ${}^W\mathbf{b}$ into frame $\{I_k\}$. The camera projection matrix can be written explicitly as

$${}^{I_k}\mathbf{C}_W = {}^{I_k}\mathbf{K}_{C_k} {}^{C_k}\mathbf{T}_W, \tag{4.2}$$

where ${}^{C_k}\mathbf{T}_W$ represents the homogeneous transformation matrix that describes the pose of frame $\{W\}$ with respect to frame $\{C_k\}$, and ${}^{I_k}\mathbf{K}_{C_k}$ represents the intrinsic matrix that maps information from frame $\{C_k\}$ to frame $\{I_k\}$. Matrix ${}^{I_k}\mathbf{K}_{C_k}$ comprises the (physically adjustable) camera focal length, pixel resolution and pixel centres, whereas matrix ${}^{C_k}\mathbf{T}_W$ contains the relative translation and orientation information of camera k, which is determined using co-planar feature calibration. Equation (4.1) can be written, without loss of generality, as

$$\left({}^{I_k}\mathbf{c}_3 \begin{bmatrix} {}^W\mathbf{b} \\ 1 \end{bmatrix} \right) {}^{I_k}\begin{bmatrix} u \\ v \end{bmatrix} = {}^{I_k}\mathbf{c}_{1:2} \begin{bmatrix} {}^W\mathbf{b} \\ 1 \end{bmatrix}, \tag{4.3}$$

where ${}^{I_k}\mathbf{c}_\mathbf{r}$ describes a matrix made up of row(s) $\mathbf{r}$ of ${}^{I_k}\mathbf{C}_W$.

Asynchronous camera model

In the case that multiple cameras are used for 3D triangulation, the various camera shutters need to be synchronised such that all cameras expose the scene at the same time. This assumes that the cameras possess global shuttering, or that the shuttering is fast enough relative to the feature(s) being captured. Without synchronisation, each camera's feature location will vary as a function of the differential shutter time and the feature's dynamics. With reference to Figure 4.1, camera 1 observes a feature's inertial-frame location ${}^W\mathbf{b}(t)$ at time t and is interpreted as the ray $\mathbf{r}_1(t)$. Given a relative shutter delay of τ_2 seconds, camera 2 observes the same feature at ${}^W\mathbf{b}(t+\tau_2)$ with the ray $\mathbf{r}_2(t+\tau_2)$ instead of $\mathbf{r}_2(t)$. The estimated feature location a time $t+\tau_2$ would therefore incorrectly be triangulated at ${}^W\hat{\mathbf{b}}(t+\tau_2)$.

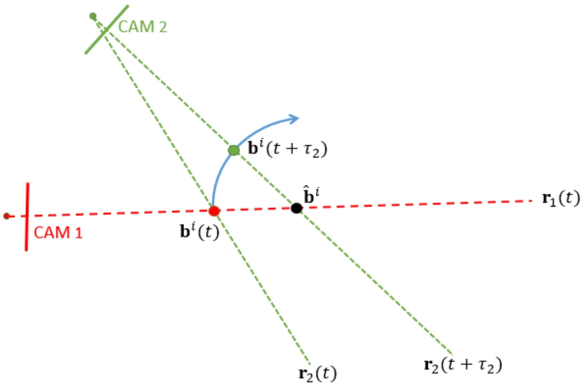


Figure 4.1: Effect of differential shutter times for two cameras when triangulating a feature's inertial-frame position. The blue arrow shows the feature's true trajectory.

Explicitly, a camera k, that is exposed τ_k seconds before/after an arbitrarily chosen reference camera will capture information which corresponds to an inertial-frame position, from Taylor series expansion, of

$${}^W\tilde{\mathbf{b}}_k = {}^W\mathbf{b} + {}^W\dot{\mathbf{b}}\tau_k + \frac{{}^W\ddot{\mathbf{b}}}{2}\tau_k^2 + \ldots, \tag{4.4}$$

where ${}^{W}\mathbf{b}$ is the inertial-frame location of the feature when the reference camera shuttering occurs. Note that τ_k is bounded in principle by the frame rate of the camera and as such cannot be outside ±1 frame (assuming the image processing frame buffering across all cameras is approximately equal). With a sufficiently high camera frame rate, relative to the tracked object's motion bandwidth, (4.4) can be suitably approximated up to its first order. Following from this, (4.3) is adjusted to include the shutter delay information as

$$\left({}^{I_k}\mathbf{c_3}\begin{bmatrix}{}^{W}\mathbf{b}+{}^{W}\dot{\mathbf{b}}\tau_k\\1\end{bmatrix}\right){}^{I_k}\begin{bmatrix}u\\v\end{bmatrix}={}^{I_k}\mathbf{c_{1:2}}\begin{bmatrix}{}^{W}\mathbf{b}+{}^{W}\dot{\mathbf{b}}\tau_k\\1\end{bmatrix}, \tag{4.5}$$

where $\tau_k = 0$ for the reference camera. Note that any camera can be chosen as the reference camera and will result in positive and/or negative shutter delay depending on the choice.

4.2.2 Resolving position and orientation

In order to estimate a non-arbitrary inertial-frame pose (position and orientation) of the quadcopter platform, at least three features, located at known positions in the object's body-frame, need to be determined. This is because three points will define a unique plane, whereas two points would define a line with orientation ambiguity when rotating about the line's vector (the feature locations would remain static and not detect the rotation). As explained in Section 2.3, three diffused IR LED arrays are used as the detectable features. Note that more features could be used to improve the feature tracking redundancy. Feature m's inertial-frame position ${}^{W}\mathbf{b}_m$ can be related to its constant (and known) location in the body-frame ${}^{B}\mathbf{b}_m$ as

$${}^{W}\mathbf{b}_m = {}^{W}\mathbf{p} + {}^{W}\mathbf{R}_B{}^{B}\mathbf{b}_m, \tag{4.6}$$

where ${}^{W}\mathbf{p}$ is the location of the quadcopter's centre of mass (coincident with the origin of frame $\{B\}$), with respect to the inertial-frame, ${}^{W}\mathbf{R}_B = \mathbf{R}({}^{W}\mathbf{q}_B)$ is the rotation matrix that captures the orientation of the quadcopter with respect to frame $\{W\}$, and

$${}^{W}\mathbf{q}_B = \begin{bmatrix}q_o\\q_x\\q_y\\q_z\end{bmatrix}^T \tag{4.7}$$

is the inertial-frame quaternion, which encodes the quadcopter attitude. For more information on quaternions, see Appendix A, or [48]. ${}^{W}\mathbf{R}_B$ can be written explicitly as

$${}^{W}\mathbf{R}_B = \begin{bmatrix}1-2(q_y^2+q_z^2) & 2(q_xq_y-q_oq_z) & 2(q_xq_z+q_oq_y)\\2(q_xq_y+q_oq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z-q_oq_x)\\2(q_xq_z-q_oq_y) & 2(q_yq_z+q_oq_x) & 1-2(q_x^2+q_y^2)\end{bmatrix}^i. \tag{4.8}$$

With reference to Appendix A, and noting that ${}^B\dot{\mathbf{b}}_m = 0$ (the location of feature m in the body-frame is always constant with respect to the body-frame), (4.6) is differentiated with respect to time to obtain the velocity of feature m as

$$ {}^W\dot{\mathbf{b}}_m = {}^W\dot{\mathbf{p}} + {}^W\dot{\mathbf{R}}{}^B\mathbf{b}_m = {}^W\dot{\mathbf{p}} + {}^W\mathbf{R}_B\lfloor{}^B\boldsymbol{\Omega}\rfloor_\times{}^B\mathbf{b}_m, \tag{4.9} $$

where ${}^B\boldsymbol{\Omega} = {}^B\!\begin{bmatrix}\Omega_x & \Omega_y & \Omega_z\end{bmatrix}^T$ is the body-frame rotational rate.

Equation (4.6) and (4.9) are used to populate (4.5) for each camera and feature pairing, which makes the camera equation a function of the state variables of interest. In this form, the camera equation can be used as a measurement equation as part of an extended Kalman filter routine, whereby the left-hand side of (4.5) serves as the output measurement, and the right-hand side represents the output measurement estimate. Note that $\tau_k = 0$ for the reference camera.

4.2.3 Extended Kalman Filter structure

State equations

The six-degree-of-freedom motion of the quadcopter can be described by the following state vector

$$ \mathbf{x} = \begin{bmatrix} {}^W\mathbf{p} \\ {}^W\mathbf{v} \\ {}^W\mathbf{q}_B \\ {}^B\boldsymbol{\Omega} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{4.10} $$

where ${}^W\mathbf{v} = {}^W\dot{\mathbf{p}}$ is the inertial-frame translational velocity of the quadrotor, and $\boldsymbol{\tau} = [\tau_2\ \tau_3\ \tau_4\ \tau_5]^T$ is the shutter delay vector containing each camera's shutter delay relative to camera 1. Note that ${}^B\boldsymbol{\Omega}$ is related to ${}^W\mathbf{q}_B$ using

$$ {}^W\dot{\mathbf{q}}_B = \frac{1}{2}{}^W\mathbf{q}_B \otimes \begin{bmatrix} 0 \\ {}^B\boldsymbol{\Omega} \end{bmatrix} = \frac{1}{2}\lfloor{}^B\boldsymbol{\Omega}\rfloor_\times^{4\times4}{}^W\mathbf{q}_B, \tag{4.11} $$

as explained in Appendix A. For sake of clarity, superscripts indicating reference frames will be omitted from this point onwards, unless explicitly required. Velocity states $\mathbf{v}$ and $\boldsymbol{\Omega}$ are driven by zero-mean noise signals $\mathbf{n}_v$ and $\mathbf{n}_\Omega$ respectively, chosen so as to match the dynamic capabilities of the designed quadcopter. Similarly, the shutter delay vector is modelled as a random walk state, driven by zero-mean noise term $\mathbf{n}_\tau$, which is chosen based on the knowledge that τ_k is bounded by ±20 milliseconds (equivalent to ±1 frame).

The fictitious discrete input vector at sample x, containing the aforementioned noise vectors, is

$$\mathbf{u}_x = \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_\Omega \\ \mathbf{n}_\tau \end{bmatrix}, \tag{4.12}$$

with corresponding standard deviations of $\boldsymbol{\sigma}_v$, $\boldsymbol{\sigma}_\Omega$, and $\boldsymbol{\sigma}_\tau$, respectively. The discrete-time Explicit Euler difference equations for a time step of Δt follow as

$$\mathbf{p}_x = \mathbf{p}_{x-1} + \Delta t \mathbf{v}_{x-1}, \tag{4.13}$$

$$\mathbf{v}_x = \mathbf{v}_{x-1} + \Delta t \mathbf{n}_v, \tag{4.14}$$

$$\mathbf{q}_x = \left(\mathbf{I}^{4\times 4} + \frac{\Delta t}{2} \lfloor \boldsymbol{\Omega}_{x-1} \rfloor_\times^{4\times 4}\right) \mathbf{q}_{x-1}, \tag{4.15}$$

$$\boldsymbol{\Omega}_x = \boldsymbol{\Omega}_{x-1} + \Delta t \mathbf{n}_\Omega, \tag{4.16}$$

$$\boldsymbol{\tau}_x = \boldsymbol{\tau}_{x-1} + \Delta t \mathbf{n}_\tau, \tag{4.17}$$

where $\mathbf{I}^{n\times n}$ represents an $n \times n$ identity matrix, and variable x indicates the current sample instance. Given a state estimate of $\hat{\mathbf{x}}_{x-1}$ at sample $x - 1$, the *a priori* state estimate at sample instance x is determined using $\hat{\mathbf{x}}_x^- = \mathbf{f}(\hat{\mathbf{x}}_{x-1})$, which is written explicitly as

$$\hat{\mathbf{p}}_x^- = \hat{\mathbf{p}}_{x-1} + \Delta t \hat{\mathbf{v}}_{x-1}, \tag{4.18}$$

$$\hat{\mathbf{v}}_x^- = \hat{\mathbf{v}}_{x-1}, \tag{4.19}$$

$$\hat{\mathbf{q}}_x^- = \left(\mathbf{I}^{4\times 4} + \frac{\Delta t}{2} \lfloor \hat{\boldsymbol{\Omega}}_{x-1} \rfloor_\times^{4\times 4}\right) \hat{\mathbf{q}}_{x-1}, \tag{4.20}$$

$$\hat{\boldsymbol{\Omega}}_x^- = \hat{\boldsymbol{\Omega}}_{x-1}. \tag{4.21}$$

$$\hat{\boldsymbol{\tau}}_x^- = \hat{\boldsymbol{\tau}}_{x-1}. \tag{4.22}$$

Note that the prediction-stage state updates assume a constant velocity in the absence of external measurements. Similarly, the shutter delay estimates remain unchanged during the prediction stage. With reference to (4.10) and (4.12), the linearised state difference equation is

$$\Delta \mathbf{x}_x = \mathbf{A}_x \Delta \mathbf{x}_{x-1} + \mathbf{L}_x \Delta \mathbf{u}_x. \tag{4.23}$$

The linearised state transition matrix from (4.23) is determined by partially differentiating the difference equations in (4.13)-(4.16) with respect to $\mathbf{x}_{x-1}$, followed by evaluating the result at the current *a priori* estimate,

$\hat{\mathbf{x}}_x^-$. The state transition matrix follows as

$$\mathbf{A}_x = \frac{\partial \mathbf{x}_x}{\partial \mathbf{x}_{x-1}}\bigg|_{\hat{\mathbf{x}}_x^-} = \begin{bmatrix} \mathbf{I}^{3\times3} & \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} \\ \mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} \\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{A}_q^q & \mathbf{A}_\Omega^q & \mathbf{O}^{4\times4} \\ \mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} & \mathbf{I}^{3\times3} & \mathbf{O}^{3\times4} \\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{O}^{3\times4} & \mathbf{O}^{4\times3} & \mathbf{I}^{4\times4} \end{bmatrix}_x, \tag{4.24}$$

where

$$\mathbf{A}_q^q = \frac{\partial \mathbf{q}_x}{\partial \mathbf{q}_{x-1}}\bigg|_{\hat{\mathbf{x}}_x^-}, \tag{4.25}$$

$$\mathbf{A}_\Omega^q = \frac{\partial \mathbf{q}_x}{\partial \mathbf{\Omega}_{x-1}}\bigg|_{\hat{\mathbf{x}}_x^-}, \tag{4.26}$$

and $\mathbf{O}^{m\times n}$ represents an $m \times n$ null matrix. The input noise covariance is $\mathbf{Q}_x = \mathrm{diag}\{\sigma_v^2, \sigma_\Omega^2, \sigma_\tau^2\}$, and the state noise Jacobian, which maps the input noise vector to the corresponding state estimates, is

$$\mathbf{L}_x = \frac{\partial \mathbf{x}_x}{\partial \mathbf{u}_x}\bigg|_{\hat{\mathbf{x}}_x^-} = \begin{bmatrix} \mathbf{O}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} \\ \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times3} & \mathbf{O}^{3\times4} \\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \mathbf{O}^{4\times4} \\ \mathbf{O}^{3\times3} & \Delta t\mathbf{I}^{3\times3} & \mathbf{O}^{3\times4} \\ \mathbf{O}^{4\times3} & \mathbf{O}^{4\times3} & \Delta t\mathbf{I}^{4\times4} \end{bmatrix}. \tag{4.27}$$

$\mathbf{A}_x$ and $\mathbf{L}_x$ are used to populate the *a priori* state error covariance matrix in (4.38).

Output equations

The output equation for camera k comprises all the detected feature locations in image plane k. Each feature has two output equations (as the measurement exists on the 2D image plane), which means that given n features, each camera will have at most $2n$ valid equations. With reference to (4.3) and (4.6), the output measurement equation and output estimate for camera 1 and feature m is

$$(\mathbf{y}_{1m})_x = \left(\mathbf{c}_{31}\begin{bmatrix} \hat{\mathbf{p}}_x^- + \mathbf{R}(\hat{\mathbf{q}}_x^-)^B\mathbf{b}_m \\ 1 \end{bmatrix}\right)\begin{bmatrix} u_m + n_x \\ v_m + n_y \end{bmatrix}_x, \tag{4.28}$$

$$(\hat{\mathbf{y}}_{1m})_x = \mathbf{c}_{1:21}\begin{bmatrix} \hat{\mathbf{p}}_x^- + \mathbf{R}(\hat{\mathbf{q}}_x^-)^B\mathbf{b}_m \\ 1 \end{bmatrix}, \tag{4.29}$$

respectively, where $\mathbf{n} = [n_x \; n_y]^T$ are the uncorrelated camera-frame noise values common across all cameras and features, with corresponding standard deviations of σ_x, and σ_y. Note that all shutter times are relative to camera 1, making camera 1 appear to not have shutter delay. With reference to (4.5), (4.6) and (4.9), the measurement equation and output estimate for cameras $k > 1$ and feature m can be written as

$$(\mathbf{y}_{km})_x = \left(\mathbf{c}_{\mathbf{3}k}\begin{bmatrix}\tilde{\mathbf{d}}_x\\ 1\end{bmatrix}\right)\begin{bmatrix}u_m + n_x\\ v_m + n_y\end{bmatrix}_x , \tag{4.30}$$

$$(\hat{\mathbf{y}}_{km})_x = \mathbf{c}_{\mathbf{1:2}k}\begin{bmatrix}\tilde{\mathbf{d}}_x\\ 1\end{bmatrix}, \tag{4.31}$$

where

$$\tilde{\mathbf{d}}_x = \hat{\mathbf{p}}_x^- + \hat{\mathbf{v}}_x^-\hat{\tau}_{kx}^- + \hat{\mathbf{R}}(\mathbf{q}_x^-)\left[{}^B\mathbf{b}_m - \lfloor {}^B\mathbf{b}_m\rfloor_\times^{3\times3}\hat{\boldsymbol{\Omega}}_x^-\hat{\tau}_{kx}^-\right], \tag{4.32}$$

and

$$\lfloor \mathbf{b}_m^b\rfloor_\times^{3\times3} = \begin{bmatrix}0 & -b_z & b_y\\ b_z & 0 & -b_x\\ -b_y & b_x & 0\end{bmatrix}_m^b . \tag{4.33}$$

Note that when $\tau_k = 0$, (4.28) and (4.29) are equivalent to that of (4.30) and (4.31), respectively. The normalised pixel noise values, n_x and n_y, are modelled based on the quantisation effect from the cameras having finite resolution, as well as the pixel re-projection error given from camera calibration.

The output measurement covariance is determined as $\mathbf{V} = \text{diag}\{\sigma_x^2, \sigma_y^2, ..., \sigma_x^2, \sigma_y^2\} \in \mathbb{R}^{30\times30}$. The output measurement equations of (4.29) and (4.31) are a function of the state variables, and as such requires finding its sensitivity to the output noise signals. It is described by

$$\mathbf{M}_x = \left.\frac{\partial\hat{\mathbf{y}}_x}{\partial\mathbf{n}}\right|_{\hat{\mathbf{x}}_x^-} = \left(\mathbf{c}_{\mathbf{3}k}\begin{bmatrix}\tilde{\mathbf{d}}_x\\ 1\end{bmatrix}\right)\mathbf{I}^{30\times30}, \tag{4.34}$$

where $\tau_k = 0$ for $k = 1$. Lastly, the observation matrix, $\mathbf{H}_x \in \mathbb{R}^{30\times13}$, is required to populate the Kalman gain and error covariance, and is described by

$$\mathbf{H}_x = \left.\frac{\partial\hat{\mathbf{y}}_x}{\partial\mathbf{x}_{x-1}}\right|_{\hat{\mathbf{x}}_x^-}. \tag{4.35}$$

The full derivation of the observation matrix in (4.35) is trivial, but is spared for sake of readability. In the interest of brevity, the full set of projection and update equations [67] are

$$\hat{\mathbf{x}}_x^- = \mathbf{f}(\hat{\mathbf{x}}_{x-1}), \tag{4.36}$$

$$\hat{\mathbf{y}}_x = \mathbf{h}(\hat{\mathbf{x}}_{x-1}), \tag{4.37}$$

$$\mathbf{P}_x^- = \mathbf{A}_x\mathbf{P}_{x-1}\mathbf{A}_x^T + \mathbf{L}_x\mathbf{Q}_x\mathbf{L}_x^T, \tag{4.38}$$

$$\mathbf{K}_x = \mathbf{P}_x^-\mathbf{H}_x^T\left[\mathbf{H}_x\mathbf{P}_x^-\mathbf{H}_x^T + \mathbf{M}_x\mathbf{V}_x\mathbf{M}_x^T\right]^{-1}, \tag{4.39}$$

$$\hat{\mathbf{x}}_x = \hat{\mathbf{x}}_x^- + \mathbf{K}_x \left[\mathbf{y}_x - \hat{\mathbf{y}}_x\right], \tag{4.40}$$

$$\mathbf{P}_x = \left[\mathbf{I}^{13\times13} - \mathbf{K}_x\mathbf{H}_x\right] \mathbf{P}_x^-. \tag{4.41}$$

The quaternion estimate is normalised in every iteration after evaluating (4.40). Given that the time step is sufficiently small relative to the flight dynamics of the quadcopter platform, the normalisation operation has a comparable effect to that of rounding error in a system with float precision.

4.2.4 Filter analysis

The noise terms in (4.28) and (4.30) are scaled by the depth estimate. This means that the elements of the output equation covariance $\mathbf{M}_x\mathbf{V}_x\mathbf{M}_x^T$ corresponding to a specific feature and camera will naturally increase as the feature moves further from said camera. In other words, the feature depth estimate of each camera acts as a trust metric, which exploits the fact that the 3D triangulation resolution of a camera is inversely proportional to the feature depth.

Observability of a specific feature's state (inertial-frame position) requires at least two cameras to detect the feature. If three features on an object are observable, a plane is defined. This makes the position and orientation of the object resolvable. On the other hand, the shutter delay estimate for a particular camera is not observable when the detected features either do not move, or move along their respective camera-frame rays (thereby appearing to be stationary to a particular camera). Explicitly, this requires that $|\mathbf{c}_{\mathbf{1}:\mathbf{2}k}(\hat{\mathbf{v}}_x^- - \hat{\mathbf{R}}(\hat{\mathbf{q}}_x^-)\lfloor {}^B\mathbf{b}_m \rfloor_\times^{3\times3} \hat{\mathbf{\Omega}}_x^-)| > 0$ for at least one of camera k's features, for τ_k to be observable. However, this does not pose a real-world issue, as the shutter delay is expected to be slowly time-varying, if at all, over the relatively short flight periods. One option when performing flight tests is to first artificially move the quadcopter (or an equivalent calibration wand) within the flight area to calibrate the shutter delay estimates, prior to performing flight tests. Note that $\mathbf{n}_\tau$ contributes to the shutter delay estimate's rate of convergence and can be adjusted accordingly.

4.2.5 Correspondence Matching

The choice of using IR markers (instead of active markers with different hues) has the added challenge that each feature is not immediately distinct. In other words, the camera cannot arbitrarily capture an image and then know how to match the IR features in the image with those on the quadcopter. This requires feature matching and ordering across all cameras from one frame to the next. The method of correspondence matching in this project is as follows: At the beginning of a flight, the quadcopter begins in a predefined pose on the ground, which places the three features in known inertial-frame locations. These known inertial-frame locations are projected into each camera's undistorted image plane using (4.1), and the Euclidean distance from each detected feature to the back-projected feature estimate is determined. For three detected features there will be a 3×3 distance matrix $\mathbf{D}$, where $\mathbf{D}(i, j)$ corresponds to the distance from feature i to back-projected feature prediction j . Feature i corresponds to feature prediction j if row i's minimum is in column j. This minimum pixel distance needs to be below a chosen error threshold in order to validate it as a correct correspondence. The threshold is scaled based on the estimated depth of the feature relative to the particular camera. Additionally, two feature predictions cannot correspond to a single measured feature, and vice versa. If only two features are detected, the height and width of the feature is analysed, as two features may be lying close enough in the camera-frame to appear as a single larger feature. The same check is conducted in the event of only one feature being detected. All valid feature measurements are then sent to the EKF to populate the output measurement equation. If a particular feature is invalid, the corresponding measurement covariance is set to a sufficiently large value to discourage use of the incorrect reading. This is done instead of restructuring the output equations and corresponding matrices as it is computationally simpler. After the filter's iteration, the new *a priori* state estimate is used to find the inertial-frame features' new location estimates based on (4.6), which is then projected into each camera's frame for use in the next iteration of correspondence matching.

4.2.6 Preliminary results

Simulation results

The visual tracking system is first simulated using the MATLAB/Simulink environment. The calculated camera intrinsics and extrinsics of the physical set-up are used in the simulation with an update rate of 50 Hz. Table 4.2 shows the noise values used in the simulation. σ_v and σ_ω are measured in m/s^2 and rad/s^2 respectively, σ_τ is measured in seconds, whilst σ_x, and σ_y are unitless. The three feature locations are generated by back-projecting the true inertial-frame position through each camera's extrinsics, and then adding the appropriate noise and quantisation. The object to be localised starts at an initial position of ${}^{W}\mathbf{p}_0 = \begin{bmatrix}0 & 1 & 1\end{bmatrix}^T$ metres. After 0.1 seconds, the feature begins to move in a 1 metre radius circle in the inertial-frame x-y plane, whilst maintaining a constant altitude. The attitude of the object constantly varies during the manoeuvre. Figure 4.2 and 4.3 show the inertial-frame position and attitude estimate tracking over the first 1.8 seconds. Figure 4.4 shows the convergent behaviour of the shutter delay estimates.

Table 4.2: Standard deviations used for the input and output measurements

σ_v	σ_ω	σ_τ	σ_x	σ_y
10	5	1×10^{-3}	1.2×10^{-3}	1.9×10^{-3}

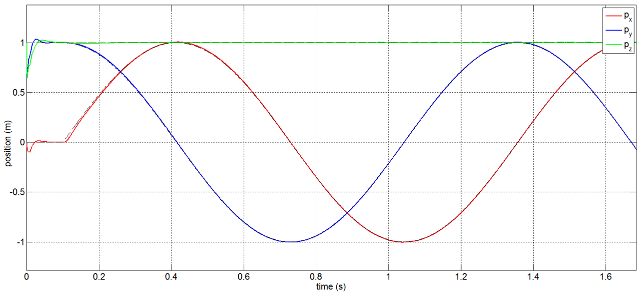


Figure 4.2: Position estimate (solid lines) compared to true position (dotted lines) over 1.8 seconds of manoeuvre

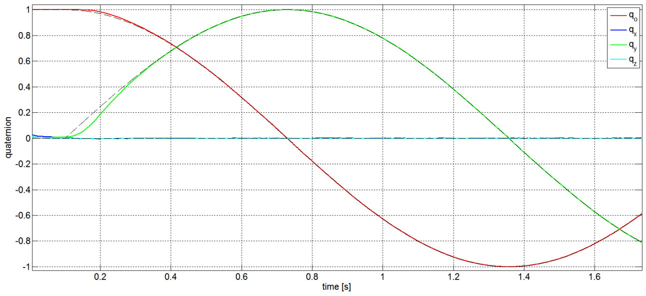


Figure 4.3: Quaternion estimate (solid lines) compared to true quaternion (dotted lines) over first 1.8 seconds of manoeuvre

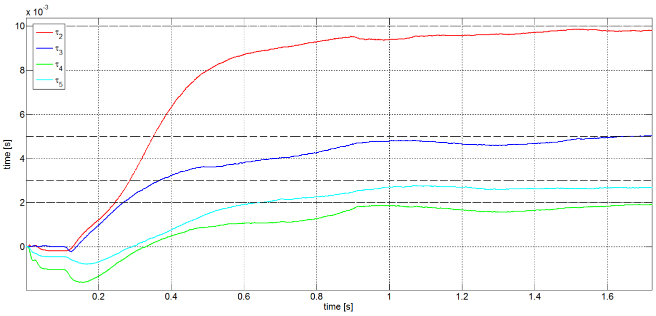


Figure 4.4: Shutter delay estimate (solid lines) compared to true shutter delay (dotted lines) over first 1.8 seconds of manoeuvre

Single feature tracking

As a simple proof of concept, the physical camera system is made to track a single feature, moving at varying speeds, in a fixed circle in the inertial-frame x-y plane. The cameras are purposefully initialised at different times. Figure 4.5 shows the filter's position estimates and corresponding error ellipses, with and without delay estimation, over a single rotation, with the feature moving at $v_f = 2.6$ m/s. Figure 4.6 shows the same information, but with the feature moving at $v_f = 7$ m/s. Table 4.3 contains the root-mean-square (RMS) error of the two tests, with and without shutter delay estimation. In the case of no delay estimation, uniform noise of ±1 frame, correctly scaled by the projected velocity estimate, is added to the camera equations. Both versions of the estimators in Figure 4.5 and 4.6 capture the true path of the object within their respective error ellipses, but the estimator that includes the shutter delay states has a noticeably smaller error ellipse. Figure 4.7 shows the shutter delay estimates over the first 1300 iterations, resulting in eventual steady-state of the shutter delay estimates. Figure 4.8 shows the diagonal position error standard deviations, which is obtained by square

Table 4.3: Root-mean-square error for the two single-feature tests, with and without shutter delay estimation

v_f (m/s)	$e_{\tau=0}$ (mm)	$e_{\tau\neq 0}$ (mm)
2.6	8.3	4.2
7	19.7	5.5

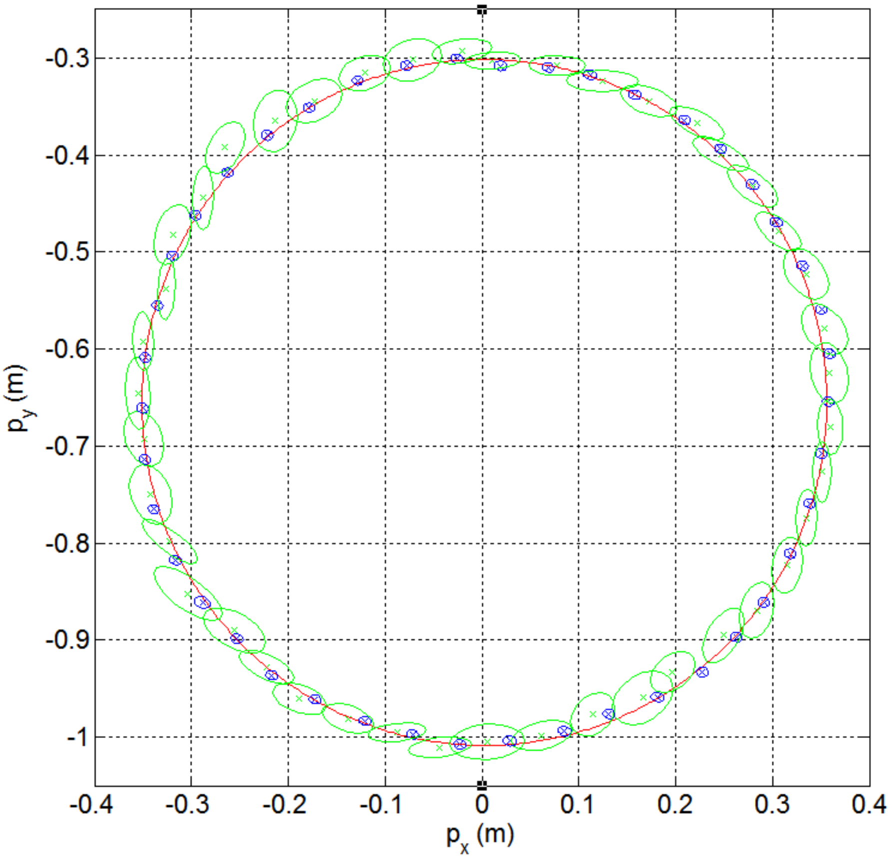


Figure 4.5: Inertial-frame x-y position estimate with delay estimation (blue cross) and without delay estimation (green cross). The corresponding error ellipses are also shown for each iteration. The true path is shown in red for a feature moving clockwise at 2.6 m/s.

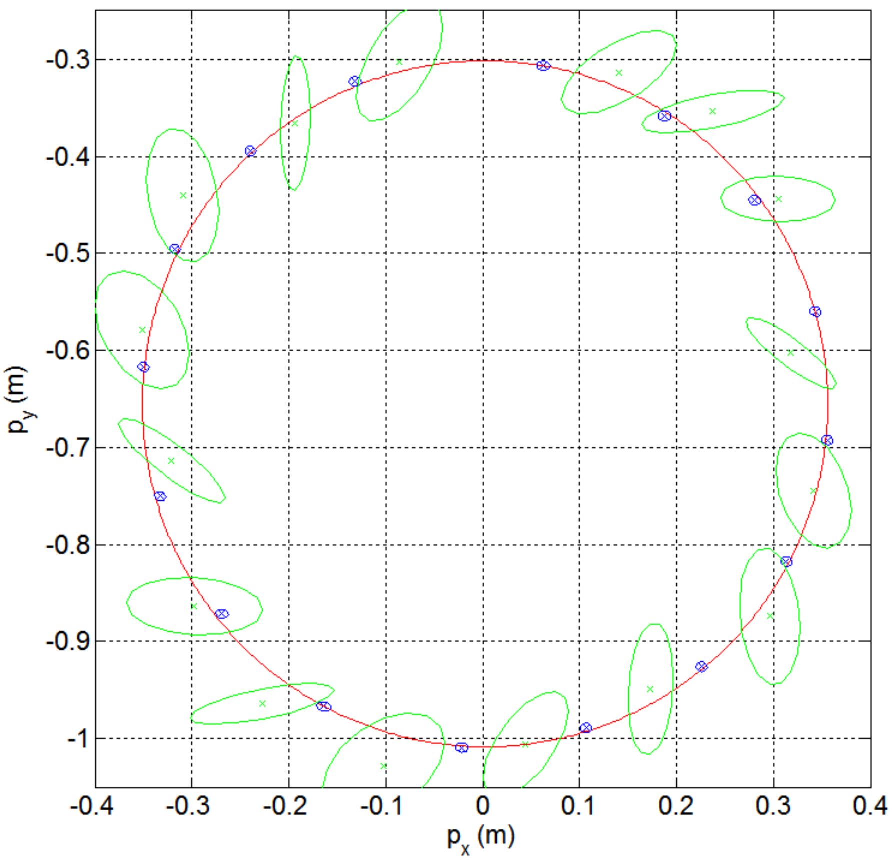


Figure 4.6: Inertial-frame x-y position estimate with delay estimation (blue cross) and without delay estimation (green cross). The error ellipses are also shown for each iteration. The true path is shown in red for a feature moving clockwise at 7 m/s.

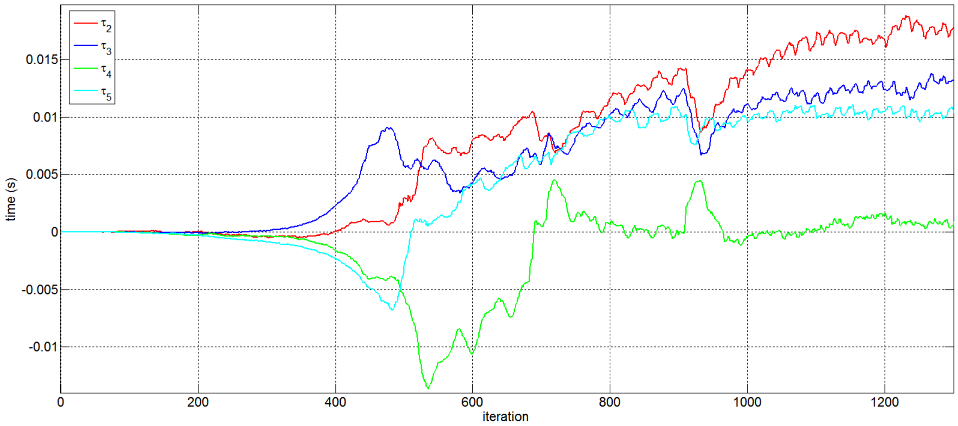


Figure 4.7: Shutter delay estimates over first 1300 iterations for test with $v_f = 7$ m/s

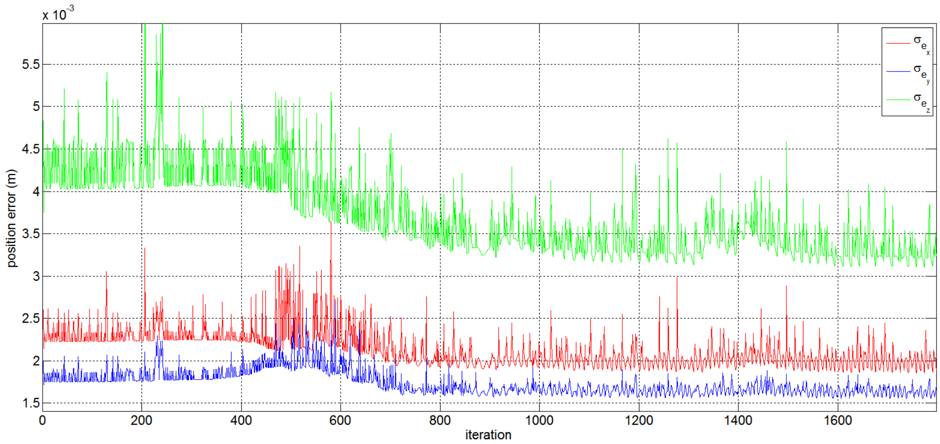


Figure 4.8: Diagonal position error standard deviations during first 26 seconds for test with $v_f = 7$ m/s

rooting the diagonal position error covariances. Noticeably, the ${}^{W}\mathcal{Z}$-axis errors in Figure 4.8 are larger than that of the ${}^{W}\mathcal{X}$- and ${}^{W}\mathcal{Y}$-axis errors. This can be attributed to the general camera orientations, as most of the cameras are configured such that their longitudinal axes, with lower resolutions, are approximately parallel with the inertial-frame $z-$axis. The spikes present in Figure 4.8 are a result of artificially increasing the output noise covariance of a camera when a specific feature is invalid, as explained in Section 4.2.5.

Multi-feature tracking

As a final demonstration of the camera's full state estimation capabilities, a quadcopter mock-up, with three features located on its body, is used to emulate a 360° flip manoeuvre. The quadcopter starts near the inertial-frame origin and then rises to an altitude of about 1 metre. The platform then proceeds to perform a rapid 360° roll manoeuvre at approximately 10 seconds into the experiment, before landing back near the origin. The same noise values from Table 4.2 are used in the experiment. Figure 4.9 shows the inertial-frame position estimate of the quadcopter, and Figure 4.10 shows the inertial-frame attitude estimate, described using intrinsic 3-2-1 Euler

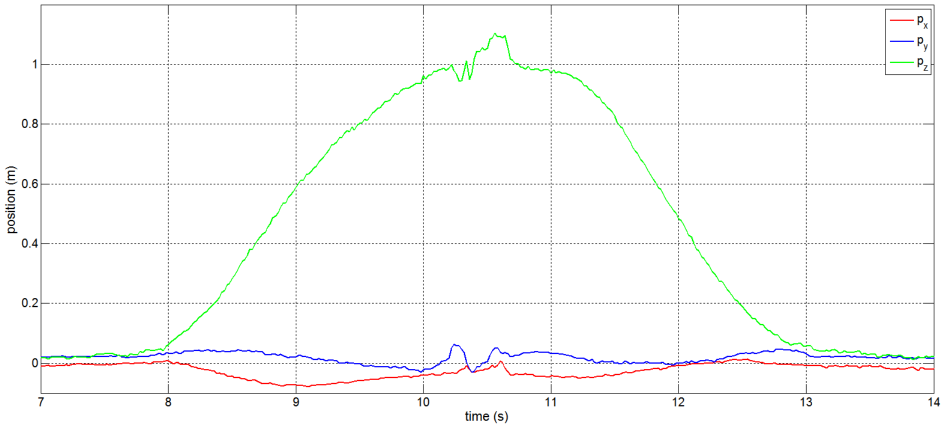


Figure 4.9: Inertial-frame position estimate of quadcopter during a 360° flip manoeuvre.

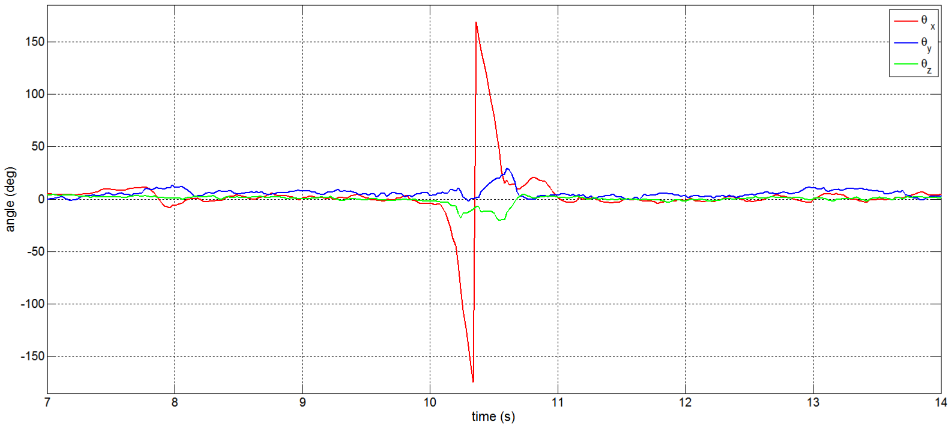


Figure 4.10: Inertial-frame orientation estimate of quadcopter during a 360° flip manoeuvre. Intrinsic $3-2-1$ Euler angles are used to represent the angular behaviour.

angles.

4.2.7 Asynchronous extended Kalman filter structure

The shutter delay estimation scheme from the previous sections is convenient when the camera shuttering time cannot be accurately synced. As demonstrated, this is especially important when the features to be tracked are moving sufficiently fast such that non-negligible triangulation errors will occur without shutter delay estimation. However, the addition of more cameras to the motion capture system requires the addition of a corresponding shutter delay state. As such, the state vector, and by extension, Kalman gain matrix dimension, will increase as more cameras are added.

An alternative formulation of the extended Kalman filter scheme does not attempt to estimate the shutter delay. Instead, the filter is run in a quasi-continuous-discrete mode, at a rate of 1 kHz. When new data from

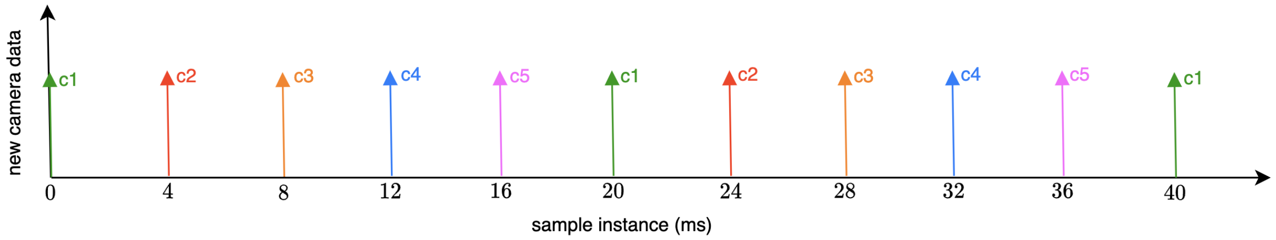


Figure 4.11: An illustration of how receiving camera data asynchronously, as a result of relative shutter delays, can increase the effective update rate of the EKF. In this particular example, the camera shutter delays are spaced such that new camera data is received every four milliseconds, which corresponds to an update rate of 250 Hz.

a camera is available, it is then included in the innovation stage of the filter, ignoring all other cameras that have not provided new data in that same sampling instance. In this way, the cameras provide asynchronous data to the Kalman filter and the filter fusion step takes place immediately (within 1 millisecond in the low-cost experimental setup, based on the filter's sampling time) when new camera data is available.

In this structure, the state is not observable using any one camera feed. However, the state will become observable over time, provided that subsequent measurement feeds are available from other cameras (assuming the features are not occluded from all cameras).

An interesting benefit of this scheme is that the effective update rate increases depending on the relative shutter delays between cameras. With reference to Figure 4.11, if the relative shutter times of the five cameras are equally spaced within a 20 millisecond window, the EKF update will take place every four milliseconds, which is equivalent to a 250 Hz update rate — five times greater than the previous filter scheme.

The following relevant files can be viewed in the flightControl GitHub repository:

- Motion capture source file
- Motion capture header file
- Camera EKF source file
- Camera EKF header file

4.3 On-board attitude estimation

In the previous section, a robust position and attitude estimator was developed, which relies on visual information from a vision-based motion capture system. Because this scheme relies solely on the low resolution camera information, the attitude estimates have undesirable quantisation noise. Specifically, the relatively small distance between any two IR markers on the quadcopter means that small angular changes result in small pixel location changes in the camera images. The attitude estimation can be improved by using on-board IMU information from the gyroscope. Note that one can either fuse the information on-board (cascaded filters) or at the ground station (centralised filter). The cascaded on-board solution is appealing for this research project, as it adds an extra layer of fidelity to the attitude estimation. This is because if there are any periods of communication failure, the on-board attitude estimator can still function in open-loop mode. Additionally, fusing the data on-board results in the smallest post-fusion latency.

The chosen method of data fusion is an adaptation of the explicit complementary filter (ECF) [68]. This estimation algorithm is simple to implement on a micro-controller, has negligible computational load, and easily incorporates the nonlinear nature of the quaternion. This method of measurement fusion is equivalent to a steady-state Kalman filter, whereby the state of interest is the body-frame angular velocity. The orientation, captured using the inertial-frame quaternion, follows as a numerical integration from said angular velocity state. This implies that the error is accceptably small in the integration from rate to orientation. This is a valid assumption when the update rate is sufficiently fast compared to the bandwidth of the quadcopter's angular velocity. In the case of the quadcopter, the bandwidth of the servomotors are in the region of 60 rad/s, or ≈ 10 Hz. The on-board ECF is run at 200 Hz (20 times faster) and constitutes sufficient time-scale separation. Given the camera-derived quaternion ${}^W\mathbf{q}_c$, obtained from the base station EKF in Section 4.2, the aim is to generate a fused quaternion ${}^W\mathbf{q}$, which yields the benefits of both the camera system and on-board IMU. In particular, the orientation estimate from the camera system is stable, but possesses quantisation noise from the low resolution cameras, and relies on camera data that is updated at 50 Hz. In contrast, the IMU-based quaternion estimate is high resolution, has a high update rate of up to 1 kHz, and does not possess significant noise, but is unstable, and will drift over time. With reference to Section A.2.2, the body-frame filter error is defined as

$$ {}^B\mathbf{e} = {}^B\alpha_e {}^B\mathbf{v}_e, \tag{4.42} $$

where ${}^B\alpha_e$ and ${}^B\mathbf{v}_e$ are extracted from the fused quaternion error, which is defined as

$$ {}^B\mathbf{q}_e = {}^W\mathbf{q}^{-1} \otimes {}^W\mathbf{q}_c = \begin{bmatrix} \cos(\frac{\alpha_e}{2}) \\ \sin(\frac{\alpha_e}{2})\mathbf{v}_e \end{bmatrix}. \tag{4.43} $$

With reference to Figure 4.12, the steady-state innovation equation, for iteration k, is

$$ {}^B\mathbf{\Omega}_k = {}^B\mathbf{\Omega}_k^- + \mathbf{K}_\Omega {}^B\mathbf{e}_k \tag{4.44} $$

where ${}^B\hat{\mathbf{\Omega}}_k^-$ is the conditioned (filtered and temperature de-biased) IMU angular velocity measurement (effectively serving as an *a priori* estimate), ${}^B\mathbf{\Omega}_k$ is the updated angular velocity estimate, and $\mathbf{K}_\Omega$ is the steady-state

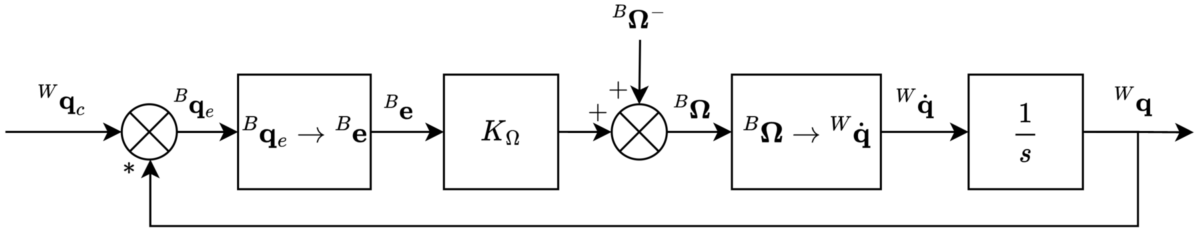


Figure 4.12: Block diagram of quaternion-based explicit complementary filter, represented in the continuous time-domain.

Kalman gain, explicitly written as

$$\mathbf{K}_\Omega = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{bmatrix}. \tag{4.45}$$

$^{B}\hat{\mathbf{\Omega}}_k^{-}$ is a new measurement source as it is not used in the ground station EKF. Note that as $^{B}\mathbf{e}$ and $^{B}\mathbf{\Omega}_k$ are both described in the body-frame, there is no cross-coupling between channels, and this allows for a diagonal $\mathbf{K}_\Omega$ matrix. After obtaining the fused angular velocity estimate in (4.44), the result is numerically integrated using the following 1^{st} order quaternion propagation equation (see Appendix A)

$$^{W}\mathbf{q}_k = \left(\mathbf{I}^{4\times 4} + \frac{\Delta t}{2}\lfloor {}^{B}\mathbf{\Omega}_k \rfloor_\times\right) {}^{W}\mathbf{q}_{k-1}. \tag{4.46}$$

The result in (4.46) is then normalised, in order to preserve the unit-norm structure of the encoded pose information.

Selection of the Kalman gain elements is performed by logging the camera quaternion and IMU gyroscope measurements, and then performing the fusion offline, with varying Kalman gain values. Larger gains will prioritise the camera information, while lower gains will place more trust in the gyroscope measurements. In principal, one could add integral action in order to account for gyroscope biases [68]. However, this is not necessary in practice, as the flight tests are short enough to routinely de-bias the gyroscope readings on start-up (prior to a flight test), as well as rely on the active temperature compensation using the thermistor measurement on the IMU.

As previously mentioned, a notable benefit of the cascaded filter approach is that if a communication issue occurs between the quadcopter and base station (e.g. partial loss of information), the on-board estimator can still function with sufficient reliability over a short period of time. Note that extended periods of time would result in on-board attitude estimation degradation, as the integrated gyroscope readings will naturally experience arbitrary drift. In the case that the quadcopter does not receive new data within a specified time-frame, the on-board attitude estimator will automatically set $\mathbf{K}_\Omega = 0$, which is equivalent to only using gyroscope information as part of the quaternion update.

The explicit complementary filter class is available for viewing in the floki GitHub repository.

4.4 Rotor thrust observer

With reference to Section 3.8, the rotor thrust profile is subject to hysteresis, as a result of the mechanical backlash of the rotor pitch actuation mechanism. This has the effect of an uncertain command signal, which can result in poor tracking performance, limit cycling, and even instability. In stark contrast, conventional speed-controlled quadcopters have the distinct benefits in that the rotor thrust production is directly related to the squared rotor speed, and can therefore be accurately estimated in open-loop [35].

Given that each servomotor-rotor unit of the designed variable-pitch quadcopter is a closed system, with a regulated speed, one can make use of the motor current measurement to estimate (and eventually close the loop on) the thrust production. For example, if the servomotor angle is changed, but the effective angle of attack of the blades remains the same, the torque demand on the motor will not change. The observer can then deduce on-board that the differential thrust is zero. With reference to Section 3.3 and Section 3.8, the rotor thrust and torque have comparable transient dynamics. By extension, the rotor thrust and motor current will share the same transient properties, based on (3.70). If the motor speed controller is effective in regulating the nominal rotor speed ($|\omega_k| = |\omega_0|$), changes in τ_k (as a result of changes in σ_k) will cause measurable changes in i_k, which can be used to estimate the thrust.

Using the thrust test rig from Section 2.6, the rotor thrust is compared with the motor current measurement, for a range of servomotor commands (within the expected quadcopter operating region), whilst keeping the rotor speed constant. With reference to Figure 4.13, the rotor thrust is approximately related to the associated motor current measurement using a quartic polynomial of the form

$$\hat{f}_k = b_0 + b_1 i_k + b_2 i_k^2 + b_3 i_k^3 + b_4 i_k^4, \tag{4.47}$$

with the polynomial coefficients provided in Table 4.4. The fourth-order polynomial is selected to balance order with goodness of fit. The corresponding time-domain comparison of the estimated and measured rotor thrust is provided in Figure 4.14.

b_0	b_1	b_2	b_3	b_4
-2.253	3.2521	-0.974	0.141	-0.0076

Table 4.4: Parameters of quartic polynomial fitting used to estimate rotor thrust from motor current.

The estimated and measured thrust signals in Figure 4.14 have a mean-squared error of 0.008 N, which suggests that the rotor current is an effective measurand to approximate rotor thrust onboard the quadcopter. However,

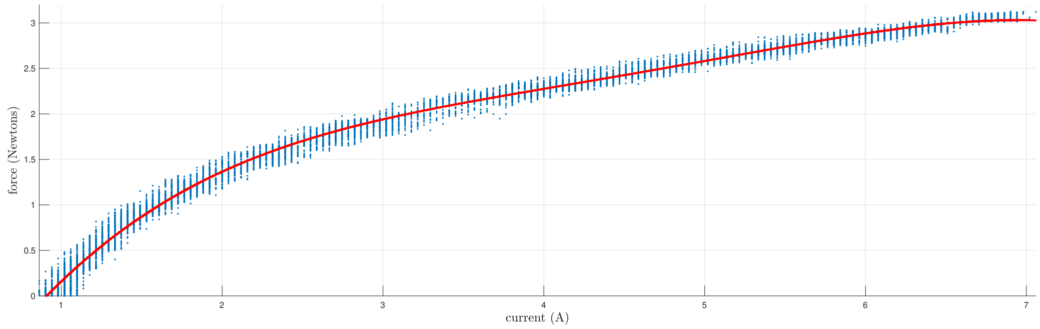


Figure 4.13: Steady-state mapping between motor measured current and rotor thrust (blue). A cubic polynomial fitting is also shown (red). The polynomial coefficients can be found in Table 4.4.

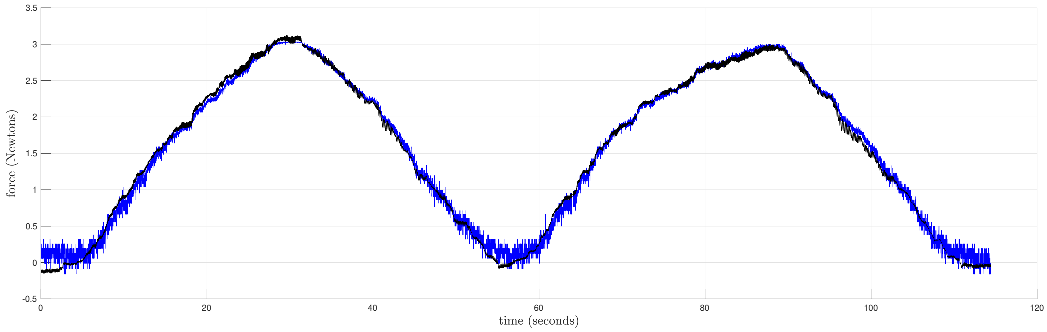


Figure 4.14: Time-domain comparison of the measured (black) and estimated (blue) rotor thrust for a slowly time-varying servomotor angle. The estimated thrust is determined using (4.47), with the polynomial coefficients in Table 4.4.

the signals in Figure 4.14 are effectively in steady-state, as the commanded servomotor angle is changed slowly, and in small increments. Additionally, the mapping should be consistent across all four rotors. Adequate rotor thrust estimation requires that the mapping be reliable not only during steady-state, near-hover flight, but also during dynamic flight manoeuvres. Figure 4.15 shows a comparison of the estimated and measured rotor thrust, using the estimation model in (4.47), during a series of servomotor angle steps. Note that a different quadcopter motor-rotor subsystem is used in this experiment. The transients in Figure 4.15 are comparable, with a mean-squared error of 0.012 N, and suggests that the sensor dynamics introduced using the current sensor can be treated as negligible. The noise profile of the rotor thrust observer is also noticeably better than that of the load cell thrust rig.

The ability to accurately estimate rotor thrust onboard the quadcopter can be used to circumvent the hysteresis present in the servomotor to blade angle transmission (as first seen in Figure 3.19). Specifically, a rotor thrust controller can be designed to remove the predominant uncertainty from the servomotor backlash. The added benefit is that high-bandwidth feedback can be achieved at the lowest dynamic chain of the quadcopter

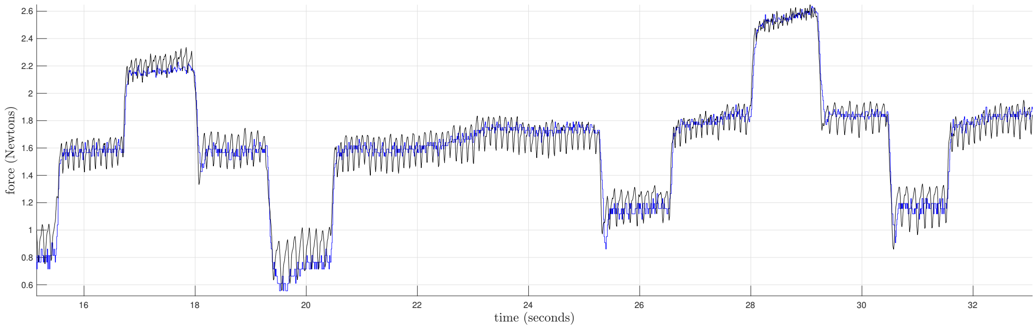


Figure 4.15: Time-domain comparison of the measured (black) and estimated (blue) rotor thrust for a series of step signals applied to the commanded servomotor angle. The estimated thrust is determined using (4.47), with the polynomial coefficients in Table 4.4.

— where disturbances (and disturbance-like effects) arise. This concept is explored in Section 5.2.3.

The thrust estimation function can be found in the thrust control class, which forms part of the floki GitHub repository.

Chapter 5

Preliminary control design

This chapter is based, in part, on the following first-authored and published paper:
Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres [21].

Section 5.1 provides an overview of the proposed control scheme of the variable-pitch quadcopter system, which includes the various feedback controllers, reference frame mappings, and control authority allocation functions. Section 5.2 presents low-level individual controller designs for rotor speed and rotor thrust control, with the latter employing a novel thrust to motor current mapping to infer thrust without direct measurement. Section 5.3 presents the mapping from the desired heave and attitude controller control action to individual rotor thrust commands. Section 5.4 makes use of platform symmetry and imposed state constraints to simplify the attitude dynamics. Section 5.5 determines the roll, pitch, and yaw channel plant sets, using information from the low-level control loops. Section 5.6 presents the thrust vectoring approach that is used to map inertial-frame position control action into the desired body-frame heave and attitude reference signals. Section 5.7 provides suggestions on the selection of the attitude and position controllers. Section 5.8 offers a closed-loop simulation of the variable-pitch quadcopter during a translation change.

5.1 Proposed control architecture

The proposed control architecture of the quadcopter is shown in Figure 5.1, whereby the signals and systems are summarised in Table 5.1. The subsequent sections provide a breakdown of each element in the control scheme. The only exception is the quadcopter dynamics block, represented with $\dot{\mathbf{x}} = \mathbf{g}(\mathbf{f})$, which was introduced in Chapter 3.

5.2 Low-level control

With reference to Figure 5.1, this Section entails the feedback control design methodologies that form part of the closed-loop rotor speed vector plants, $\mathbf{T}_{\omega}(s)$, and rotor thrust vector plants, $\mathbf{T}_{\mathbf{f}}(s)$. As alluded to in Chapter 3,

Name	Description	Size
${}^{W}\mathbf{p}^{*}$	Desired inertial-frame location of quadcopter	3×1
${}^{W}\mathbf{p}$	Measured inertial-frame location of quadcopter	3×1
$\mathbf{X}_{\mathbf{p}}$	Inertial-frame feedforward filter	3×3
$\mathbf{G}_{\mathbf{p}}$	Inertial-frame position controller	3×3
${}^{W}\mathbf{f}_{\Sigma}^{*}$	Desired inertial-frame thrust vector	3×3
${}^{h^{*},\mathbf{\Theta}^{*}}\mathcal{M}_{\mathbf{f}_{\Sigma}}$	Mapping from desired inertial-frame thrust to corresponding desired heave and orientation (described as a quaternion)	N/A
h^{*}	Desired body-frame heave thrust	1×1
${}^{B}\mathbf{\Theta}_{e}$	Body-frame angular error	3×1
${}^{W}\mathbf{q}_{B}$	Measured body-frame orientation with respect to the inertial-frame (described as a quaternion)	4×1
$\mathbf{G}_{\eta}$	Attitude controller	3×3
$\mathbf{u}_{\eta}$	Attitude control action	3×1
${}^{\mathbf{f}^{*}}\mathcal{M}_{\mathbf{u}}$	Mapping from desired heave and atittude control action to desired rotor thrust vector	N/A
$\mathbf{f}^{*}$	Desired rotor thrust vector	4×1
$\mathbf{f}$	Measured rotor thrust vector	4×1
$\mathbf{T}_{\mathbf{f}}$	Closed-loop rotor thrust vector plant	4×4
$\boldsymbol{\omega}^{*}$	Desired rotor thrust vector	4×1
$\boldsymbol{\omega}$	Measured rotor thrust vector	4×1
$\mathbf{T}_{\omega}$	Closed-loop rotor speed vector plant	4×4
$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{f})$	Quadcopter dynamics	N/A
$\mathbf{x}$	Quadcopter state vector	18×1

Table 5.1: Summary of signals and systems from Figure 5.1. The signal/system size is also included, along with a reference to the particular subsection that provides more information.

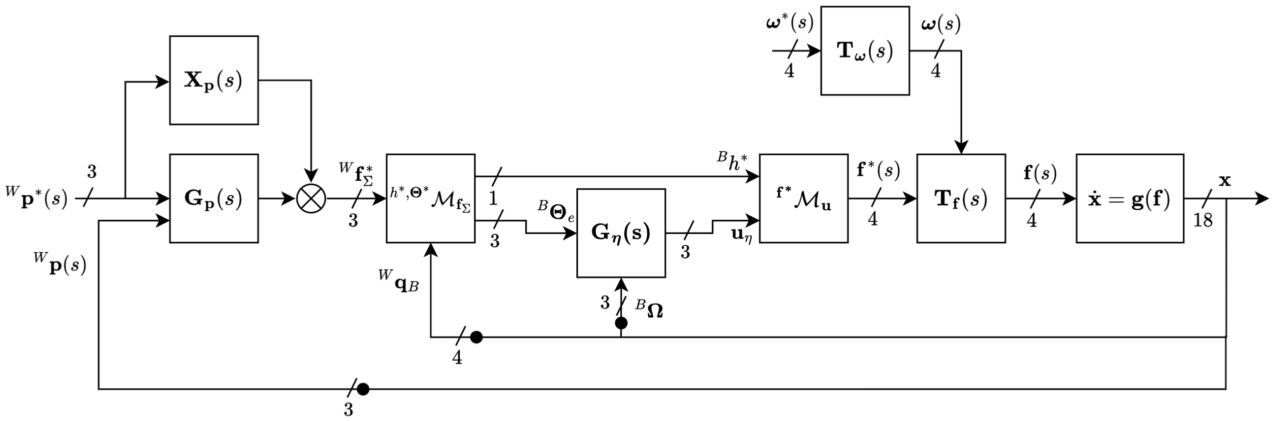


Figure 5.1: Abstracted block diagram of quadcopter control scheme, which contains the various control signal mappings, feedback controllers, and closed-loop subsystems. The signals and systems are detailed in Table 5.1
.

the design choice was made to separate the low-level control into two subsystems, namely: rotor speed control, and rotor thrust control. While the two subsystems are inherently coupled, experimental results from Section 3.8 show that enforcing approximately constant rotor speeds can decouple the two subsystems.

5.2.1 Control allocation between rotor speed and thrust

The directly controllable input signals for rotor k are the desired motor voltage, v_k^*, and the commanded servomotor angle, α_k^*. While there exist many control action mixing schemes that can achieve the aforementioned low-level control, a simple and intuitive approach is to use v_k^* to regulate the rotor speed, and α_k^* to adjust the rotor thrust as needed. With reference to Figure 3.21 in Section 3.8, the relationship between the commanded servomotor angle and rotor thrust is approximately linear around the hover thrust condition (when monotonically increasing or decreasing). This suggests that using α_k^* to control f_k is plausible when the rotor velocity is maintained at the designed constant operating speed. Additionally, the transfer behaviour from commanded motor voltage to rotor speed is well known, based on the experimental work in Section 3.7.4. Notably, α_k^* is known to have non-negligible transfer behaviour to the rotor speed, and this should be accounted for as part of a rigorous control design.

5.2.2 Rotor speed control

Plant template

Section 3.7 developed a small signal transfer function model of the quadcopter rotor speed system, which takes the form of

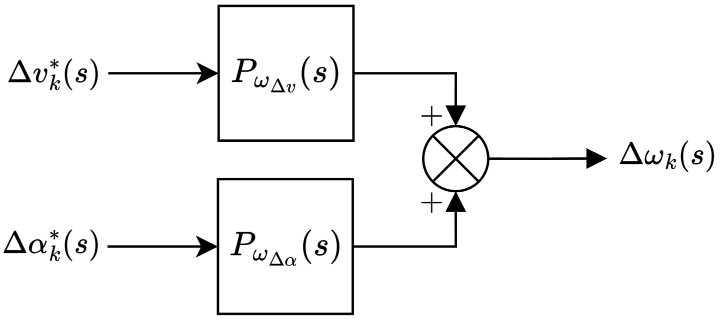


Figure 5.2: Block diagram of linearised open-loop rotor speed model.

$$\begin{aligned}\Delta\omega(s) &= P_{\omega_{\Delta v}}(s)\Delta v_k^*(s) + P_{\omega_{\Delta\alpha}}(s)\Delta\alpha_k^*(s),\\ &= \frac{a_{\Delta v}e^{-st_{\Delta v}}}{1+sT_{\Delta v}}\Delta v_k^*(s) + \frac{a_{\Delta\alpha}e^{-st_{\Delta\alpha}}}{1+sT_{\Delta\alpha}}\Delta\alpha_k^*(s),\end{aligned} \tag{5.1}$$

as depicted in Figure 5.2, with each parameter and the corresponding uncertainty range described in Table 5.2 (repeated from Section 3.7 for sake of convenience). In order to capture the phase lag behaviour that results from

$a_{\Delta v}$	$T_{\Delta v}$	$t_{\Delta v}$	$a_{\Delta\alpha}$	$T_{\Delta\alpha}$	$t_{\Delta\alpha}$
0.234 ± 0.016	0.28 ± 0.007	0.0025 ± 0.0025	-0.102 ± 0.008	0.29 ± 0.01	0.005

Table 5.2: Motor-rotor plant parameters with corresponding uncertainty ranges.

a digital feedback control implementation, the plant set is mapped into the w-domain [69], which is represented as

$$\begin{aligned}\tilde{P}_{\omega_{\Delta v}}(w) &\approx P_{\omega_{\Delta v}}(w)(1-\frac{w}{2f_\omega}),\\ \tilde{P}_{\omega_{\Delta\alpha}}(w) &\approx P_{\omega_{\Delta\alpha}}(w)(1-\frac{w}{2f_\alpha}),\end{aligned} \tag{5.2}$$

where $f_\omega = 95.49$ and $f_\alpha = 200$ are the sampling frequencies (in Hz) of the rotor speed and servomotor angle, respectively. Note that the PWM frequency of the motor ESCs is also 200 Hz. The approximation in (5.2) requires that the respective frequencies must be greater than the highest corner frequency (in rad/s) of the corresponding plant that has significant residual.

Using the representation from (5.2) and Table 5.2, the open-loop plant templates are generated for a range of frequencies of interest, as shown in Figure 5.3-5.4. The design choice of enforcing a constant motor-rotor speed in Section 3.5 means that the rotor speed control problem takes the shape of a disturbance regulator, whereby the reference rotor speed remains constant, but the effective disturbances from blade angle changes need to be mitigated. Additionally, given that the blade angle changes are measurable in some form (specifically, the commanded servomotor angles are known at the current sampling instance), this information can be used in a feedforward structure to preemptively account for load torque variations (this only works as far as the model

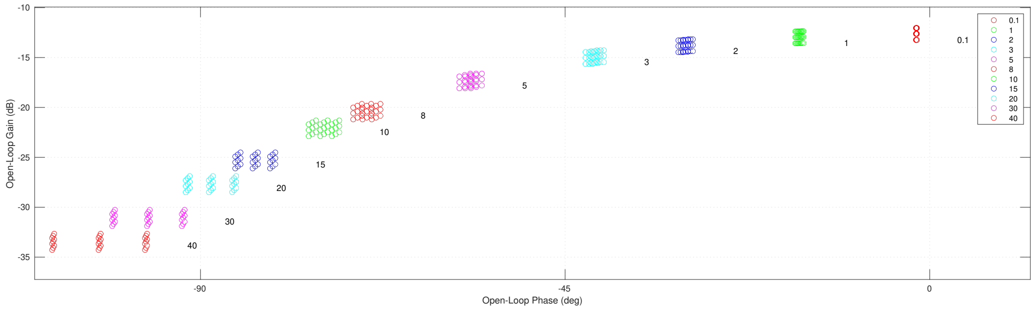


Figure 5.3: Open-loop plant template of $\tilde{P}_{\omega_{\Delta v}}(jv)$, displayed in the log-polar plane.

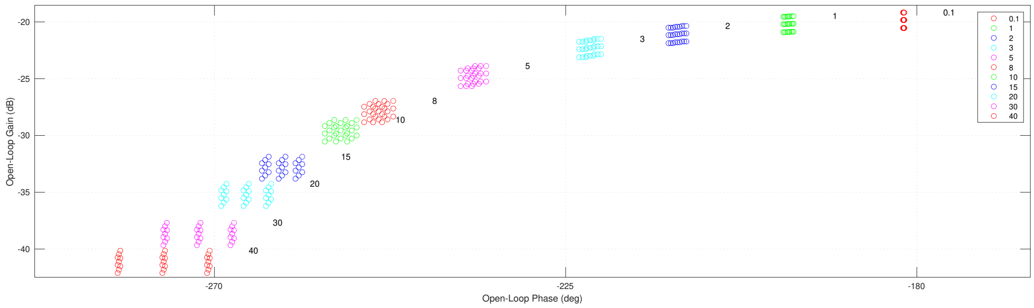


Figure 5.4: Open-loop plant template of $\tilde{P}_{\omega_{\Delta\alpha}}(jv)$, displayed in the log-polar plane.

is certain). This has the benefit of offloading control action from the feedback controller, which means that the feedback component of the input (action) signal is smaller, reducing the overall demand on the actuator.

Control architecture

Following the control allocation insights in Section 5.2.1, the control scheme in Figure 5.5 is proposed, where $G_\omega(w)$ is the feedback controller acting on the rotor speed error, $X_\omega(w)$ is a feedforward element acting on the commanded servomotor angle, and $F_\omega(w)$ is a pre-filter that acts on the desired rotor speed, $\omega_k^*(w) = \omega_0$. Note that sensor dynamics of the rotor speed system are inherently captured as part of the system identification in Section 3.7.4. While $\tilde{P}_{\omega_{\Delta v}}(w)$ and $\tilde{P}_{\omega_{\Delta\alpha}}(w)$ are modelled based on differential signal changes, the inclusion of the feedback control loop means that residual signal behaviour will be removed if $G_\omega(w)$ is at least Type-1 (containing one integrator). This is also necessitated by the requirement to enforce $\omega_k = \omega_0$ at steady-state under disturbances and plant uncertainty.

Selection of pre-filter, $F_\omega(w)$

$F_\omega(w)$ is used to steadily ramp up/down the rotor speed before/after testing the system. In the case of the case of ramping up the speed, this allows all the rotors to reach their predefined nominal operating conditions without experiencing rapid speed changes. Otherwise, the excessive rotational accelerations on the rotors can

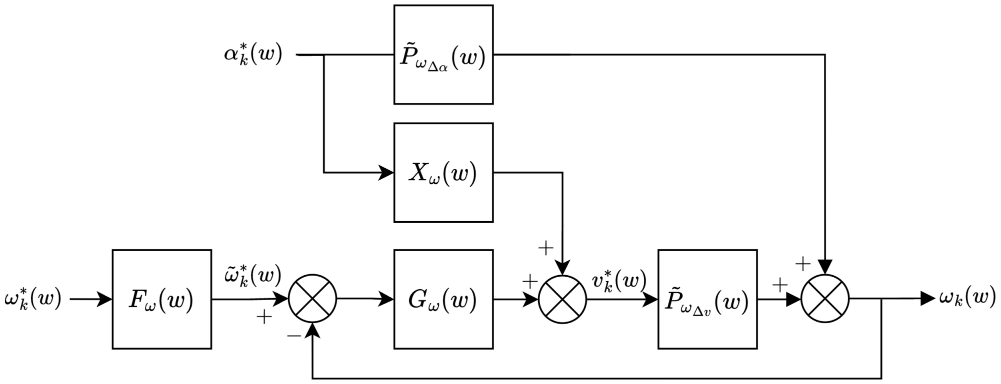


Figure 5.5: Block diagram of proposed rotor speed control scheme with disturbance feedforward element, $X_\omega(w)$.

induce moments on the quadcopter body, as explained in Section 3.4. At steady-state, the filtered reference rotor velocity is required to approach the nominal rotor velocity ($\tilde{\omega}_k^* \to \omega_0$), thereby requiring that $F_\omega(w)$ has DC gain of unity. The rotor speeds are then ramped down after testing has completed. This is done to avoid rapidly braking the motor shaft, which would be detrimental to the various mechanical elements on the quadcopter. Following this, the pre-filter is chosen as,

$$F_\omega(w) = \begin{cases} \frac{1}{1+w/0.35} & \omega_k^* = \omega_0, \\ \frac{1}{1+w/1} & \omega_k^* = 0, \end{cases} \tag{5.3}$$

which corresponds with a ±2% settling time of around 11 seconds when the reference rotor speed is $\omega_k^* = \omega_0$ (when the rotor speeds are ramping up before testing), and 4 seconds when $\omega_k^* = 0$ (when ramping down the rotor speeds after testing).

Selection of feedforward filter, $X_\omega(w)$

Design of $X_\omega(w)$ can be performed using necessary and sufficient existence conditions, for example using a disturbance-equivalent method presented in Section 6.4-6.6. However, Table 3.6 suggests that $\tilde{P}_{\omega_{\Delta v}}(w)$ and $\tilde{P}_{\omega_{\Delta\alpha}}(w)$ have near-identical transients, with the only substantial difference being their DC gain terms. Based on this observation, an appropriate feedforward filter gain is

$$X_\omega(w) = -\frac{a^0_{\Delta\alpha}}{a^0_{\Delta v}}, \tag{5.4}$$

where $a^0_{\Delta\alpha} = -0.102$ and $a^0_{\Delta v} = 0.234$ are the centroids of the respective parameters sets in Table 3.6 and represent the nominal parameters. Following this, the open-loop representation of (5.1) can be adjusted to

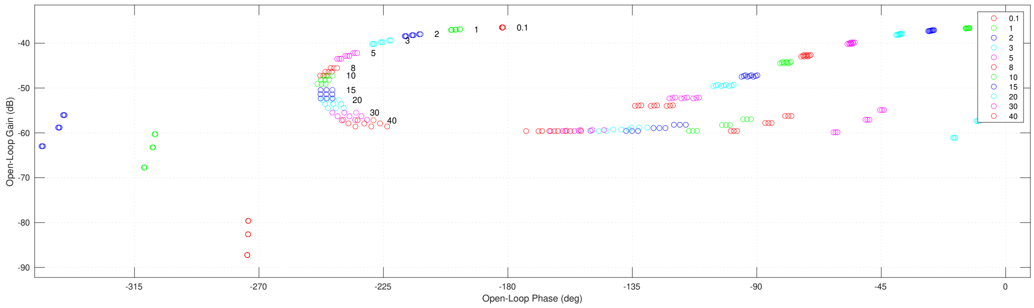


Figure 5.6: Open-loop plant template of $\overline{P}_{\omega_{\Delta\alpha}}(j\nu)$, displayed in the log-polar plane.

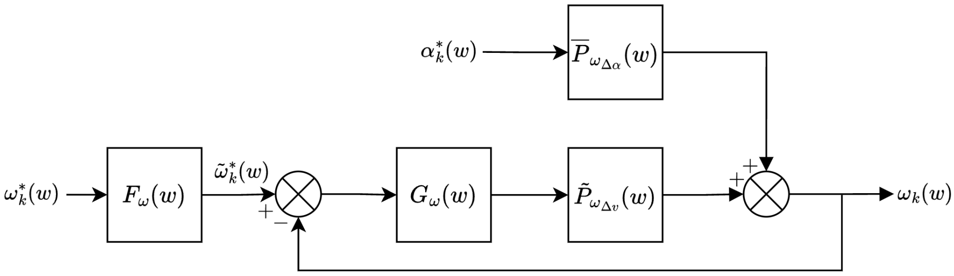


Figure 5.7: Block diagram of proposed rotor speed control scheme, adapted from that of Figure 5.5.

include the feedforward information as

$$\begin{aligned}\Delta\omega(w) &= \tilde{P}_{\omega_{\Delta v}}(w)\Delta v_k^*(w) + \left(\tilde{P}_{\omega_{\Delta\alpha}}(w) + X_\omega(w)\tilde{P}_{\omega_{\Delta v}}(w)\right)\Delta\alpha_k^*(w),\\ &= \tilde{P}_{\omega_{\Delta v}}(w)\Delta v_k^*(w) + \left(\tilde{P}_{\omega_{\Delta\alpha}}(w) - \frac{a_{\Delta\alpha}^0}{a_{\Delta v}^0}\tilde{P}_{\omega_{\Delta v}}(w)\right)\Delta\alpha_k^*(w),\\ &= \tilde{P}_{\omega_{\Delta v}}(w)\Delta v_k^*(w) + \overline{P}_{\omega_{\Delta\alpha}}(w)\Delta\alpha_k^*(w),\end{aligned} \tag{5.5}$$

where

$$\overline{P}_{\omega_{\Delta\alpha}}(w) = \tilde{P}_{\omega_{\Delta\alpha}}(w) - \frac{a_{\Delta\alpha}^0}{a_{\Delta v}^0}\tilde{P}_{\omega_{\Delta v}}(w) \tag{5.6}$$

captures the gross transfer behaviour from commanded servomotor angle to rotor speed. Note that in the nominal plant case ($a_{\Delta\alpha} = a_{\Delta\alpha}^0$, $a_{\Delta v} = a_{\Delta v}^0$), commanded servomotor angle changes will have a negligible effect on the rotor speed: $\overline{P}_{\omega_{\Delta\alpha}}(w) \approx 0$. The updated plants set is shown in Figure 5.6, after inclusion of the feedforward information. As shown in Figure 5.6, the gain of plant set $\overline{P}_{\omega_{\Delta\alpha}}(w)$ is shown to significantly reduce, relative to $\tilde{P}_{\omega_{\Delta\alpha}}(w)$ in Figure 5.4, as a result of the feedforward term, thereby making the local linear rotor speed dynamics approximately independent of the commanded servomotor angle changes.

Disturbance rejection specifications

The block diagram in Figure 5.5 can be represented compactly as shown in Figure 5.7, whereby the closed-loop transfer behaviour from commanded servomotor angle, $\alpha_k^*(w)$, to rotor speed error, $e_{\omega_k}(w) = \tilde{\omega}_k^*(w) - \omega_k(w)$, is

$$\frac{e_{\omega_k}(w)}{\alpha_k(w)} = \frac{-\overline{P}_{\omega_{\Delta\alpha}}(w)}{1 + \tilde{P}_{\omega_{\Delta\nu}}(w)G_\omega(w)}. \tag{5.7}$$

Importantly, the one caveat of the representation in Figure 5.7 is the incorrect depiction of the true plant input, which is subject to saturation. Enforcing an approximately constant rotor speed under load torque variations requires that the frequency response magnitude of $\frac{e_{\omega_k}(w)}{\alpha_k(w)}$ is sufficiently small. The disturbance rejection design problem follows as

$$\left|\frac{e_{\omega_k}(j\nu)}{\alpha_k(j\nu)}\right| \leq D_\alpha(\nu), \tag{5.8}$$

where $D_\alpha(\nu)$ is the prescribed magnitude bound as a function of the design frequencies. The structure of (5.7) suggests that $\frac{e_{\omega_k}(w)}{\alpha_k(w)}$ will band-passing within the frequency range of interest. Following this, the disturbance rejection bound is specified using the canonical band-passing structure of

$$D_\alpha(\nu) = \left| k_\alpha \frac{2\zeta_\alpha \omega_\alpha w}{w^2 + 2\zeta_\alpha w + \omega_\alpha^2} \right|_{w \to j\nu}. \tag{5.9}$$

The disturbance rejection model is idealised based on a worst-case unit step on the desired rotor force, which is approximately 65% of the hover thrust demand of $f_0 = 1.55$ N. With reference to (3.21a), the commanded servomotor angle corresponding to a 1 N force change is $\Delta\alpha = \frac{1}{0.0016} = 625$ (when measured in 16-bit count). A realistic expectation is to reduce the worst-case angular rate deviation to 5 rad/s away from the nominal operating point of $\omega_0 = 600$ rad/s (an allowance of less than 1% deviation). If $G_\omega(w)$ is designed to be strictly proper, the high-frequency behaviour of (5.7) will tend to $-\overline{P}_{\omega_{\Delta\alpha}}(w)$. This knowledge can be used to shape the high-frequency component of the disturbance bound. Figure 5.8 shows the Bode magnitude response of $-\overline{P}_{\omega_{\Delta\alpha}}(w)$. The transfer function behaviour of the closed-loop disturbance to error function at high frequencies is expected to tend towards $2k_\alpha\zeta_\alpha\omega_\alpha/w$. Following this, k_α, ζ_α and ω_α can be chosen such that the magnitude

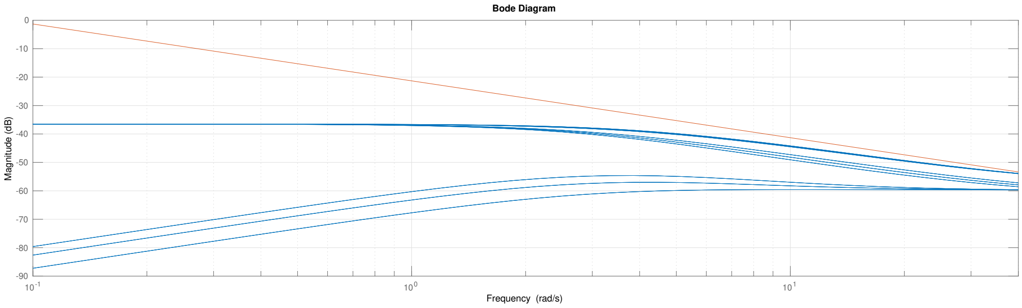


Figure 5.8: Bode magnitude plot of $-\overline{P}_{\omega_{\Delta\alpha}}(w)$ over frequency range of interest.

k_α	ζ_α	ω_α
0.011	1	4

Table 5.3: Chosen parameters of disturbance rejection specification in (5.9).

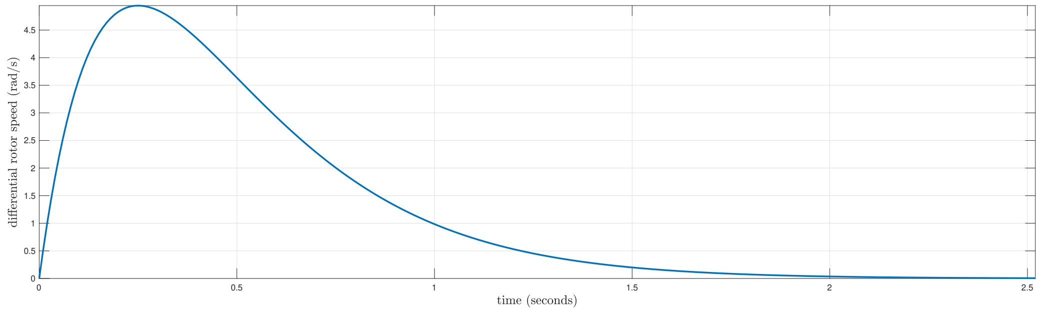


Figure 5.9: Time-domain plot, showing the step response of the disturbance model for a 1 N commanded rotor thrust, based on (5.9) and Table 5.3.

response of $2k_\alpha\zeta_\alpha\omega_\alpha/w$ overbounds the Bode magnitude plot of $-\overline{P}_{\omega_{\Delta\alpha}}(w)$ over the design frequencies of interest. An appropriate choice of $2k_\alpha\zeta_\alpha\omega_\alpha/w$ is also overlaid in Figure 5.8. Using a damping ratio of $\zeta_\alpha = 1$, k_α and ω_α are then varied until the time-domain response of $S_\alpha(w)$ has a maximum peak of 5 rad/s (for a commanded servomotor angle change of $\Delta\alpha = 625$) and a sufficiently small settling time. The parameters are summarised in Table 5.3, and the corresponding (1 N rotor force equivalent) step response of the disturbance model is shown in Figure 5.9.

Design constraints

Combining (5.7) and (5.8), and omitting the frequency-domain parameter ω for sake of clarity, the design constraint follows as

$$\left|\frac{-\tilde{P}_{\omega_{\Delta\alpha}}}{1+P_{\omega_{\Delta v}}G_\omega}\right| \le D_\alpha. \tag{5.10}$$

Equation (5.10) can be approached with standard QFT toolboxes, as it is a linear fractional mapping on G_ω. The constraint in (5.10) can be represented as

$$\frac{\left|\tilde{P}_{\omega_{\Delta\alpha}}\right|}{\left|P_{\omega_{\Delta v}}\right|Dc_\alpha} \le \left|G_\omega - \left(-\frac{1}{P_{\omega_{\Delta v}}}\right)\right|, \tag{5.11}$$

which represents the discoidal solution space of $G_\omega(j\omega)$ when visualised in the arithmetic-complex plane. The valid solution space of $G_\omega(j\omega)$ exists outside of the disc with a centre of $-\frac{1}{P_{\omega_{\Delta v}}}$ and a corresponding radius of $\frac{|\tilde{P}_{\omega_{\Delta\alpha}}|}{|P_{\omega_{\Delta v}}|D_\alpha}$. As $P_{\omega_{\Delta v}}$ and $\tilde{P}_{\omega_{\Delta\alpha}}$ describe an ordered pair of plant sets, the resulting global solution space of $G_\omega(j\omega)$ will comprise the intersection of the stay-out discoidal regions across all ordered plant instances. Notably, a sufficiently large feedback controller gain, $|G(j\omega)|$, will satisfy the design constraint in (5.11).

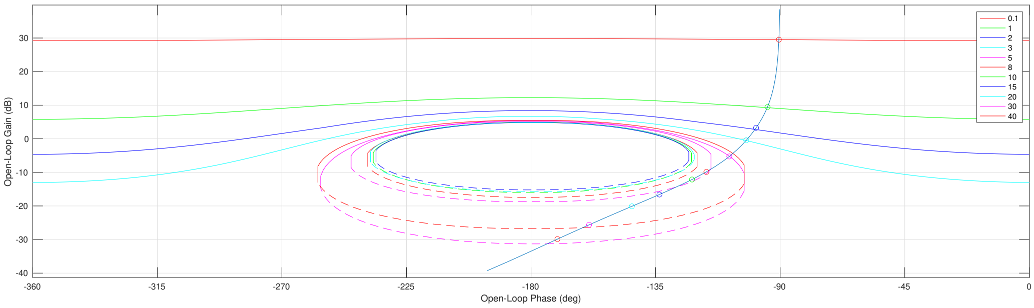


Figure 5.10: Log-polar plane, showing the design regions from (5.11). The frequency response of the nominal open-loop transfer function is also shown.

However, a pragmatic design that does not unduly amplify sensor noise necessitates a low controller gain in the mid- to high-frequency band. As such, the feedback control design proceeds by solving the set of design constraints in (5.11) with the smallest possible gain, whilst also trying to select a low-order controller that is strictly proper.

Feedback control design

A graphical visualisation of the constraint set in (5.10) is facilitated using an open-source MATLAB toolbox [70], which displays the *stay-above* and *stay-below* bounds of the nominal open-loop frequency response in the log-polar plane. Figure 5.10 shows the resulting design regions, as well as the frequency response of the nominal loop transfer function, $\tilde{P}^0_{\omega_{\Delta v}}(w)G_\omega(w)$, where

$$\begin{aligned} G_\omega(w) &= K_\omega \frac{1 + w/\alpha_z}{w(1 + w/\alpha_p)}, \\ &= 12.77 \frac{1 + w/4}{w(1 + w/21)}, \end{aligned} \tag{5.12}$$

and

$$\begin{aligned} \tilde{P}^0_{\omega_{\Delta v}}(w) &= \frac{a^0_{\Delta v} e^{-w t_{\Delta v}}}{1 + w T^0_{\Delta v}} (1 - \frac{w}{2 f_\omega}), \\ &= \frac{0.234 e^{-0.005 w}}{1 + 0.28 w} (1 - \frac{w}{190.99}). \end{aligned} \tag{5.13}$$

The feedback controller in (5.12) ends up taking the form of a PI controller with additional gain-phase rolloff at the higher frequencies from the auxillary pole (sometimes referred to as a PIF controller). The effect is likely small, but a benefit of $G_\omega(w)$ rolling off is that this cancels the roll-up from the sampling effect. While the loop shaping naturally dictates this shape for low gain-phase design at the mid-frequencies, the strictly proper structure of $G_\omega(s)$ is also beneficial in terms of reducing sensor noise amplification in the high-frequency band (where the loop transfer behaviour is effectively open). As previously mentioned, the integrator in the loop transfer function is required to fully reject disturbances at steady-state. Adherence to the disturbance rejection requirements across all frequencies is verified by comparing the Bode magnitude response of the disturbance rejection bound in (5.9) with that of (5.7), which is shown in Figure 5.11. The closed-loop transfer behaviour from commanded servomotor angle to rotor speed error is band-passing as expected and is shown to conform to the design constraint within the frequency band of interest.

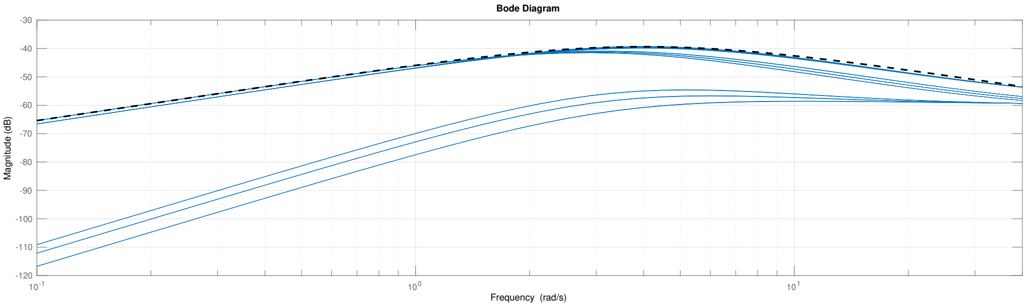


Figure 5.11: Bode magnitude plot, comparing the prescribed rejection bounds (dotted), with that of the resulting disturbance rejection behaviour of the closed-loop system.

The simulated time-domain disturbance rejection is shown in Figure 5.12 for a commanded servomotor angle change equivalent to a 1 Newton rotor force change. The disturbance rejection behaviour of the closed-loop system in Figure 5.12 is comparable to that of the time-domain specification and validates the selection of $G_\omega(w)$ and $X_\omega(w)$. Notably, the majority of the control action is contributed by the feedforward element, $X_\omega(w)$, which preemptively compensates for the commanded servomotor angle change.

Gain scheduling during rotor spool up

As explained in Section 2.2.2, the rotor speed measurement update rate is dependant on the rotor speed. The nominal rotor speed of $\omega_k = \omega_0 = 600$ rad/s results in a measurement update rate of 95.49 Hz. However, during

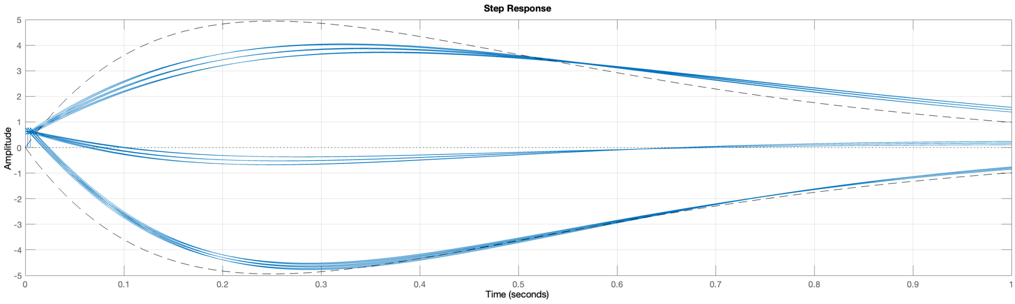


Figure 5.12: Time-domain plot, showing the disturbance rejection behaviour of the closed-loop system to a commanded servomotor angle change equivalent to a 1 N rotor force change.

the initial spool up of the rotor speed to the nominal operation condition, the corresponding update rate will also increase in proportion to the rotor speed. Explicitly, the update rate of rotor speed k is

$$f_{\omega_k} = \frac{\omega_k}{2\pi}\ \text{Hz}. \tag{5.14}$$

As such, during the initial spool up period, digital sampling effects from the low measurement update rate will introduce significant phase lag (and likely send the control loop into limit cycling or instability). To counter this, the feedback controller gain is initially reduced, at start-up of the rotors, to K_{ω_0}. Once the rotor speed is within a specified threshold of the nominal rotor speed, $|\omega_0 - \omega_k| \leq \Delta\omega_K$, the feedback controller gain then begins to monotonically increase in proportion to the rotor speed using the approach in [71], namely

$$K_{\omega_k} = K_\omega \frac{\omega_k}{\omega_0}. \tag{5.15}$$

As explained in [71], scheduling the gain in this way preserves the original stability properties of the feedback controller, whilst reducing the bandwidth during spool up. In particular, when visualising the open-loop transfer function in the log-polar plane after inclusion of the gain scheduling term, the loop profile remains the same regardless of ω_k (for $\omega_k > 0$). The gain scheduling parameters are summarised in Table 5.4. Note that the feedback controller pole and zero locations remain unchanged.

K_{ω_0}	$\Delta\omega_K$	ω_0
6.4	300	600

Table 5.4: Gain scheduling parameters used when adjusting rotor speed gain during spool up of rotors.

Implementation

Implementation of the controllers on the quadcopter necessitates an algorithmic form that can be deployed on the microcontroller. This is achieved by first representing the PI-type controller from (5.12) in the equivalent anti-reset windup form from [72], as shown in Figure 5.13. The structure in Figure 5.13 is able to account for integrator windup, whereby clamping occurs in the forward path when the control action becomes too large,

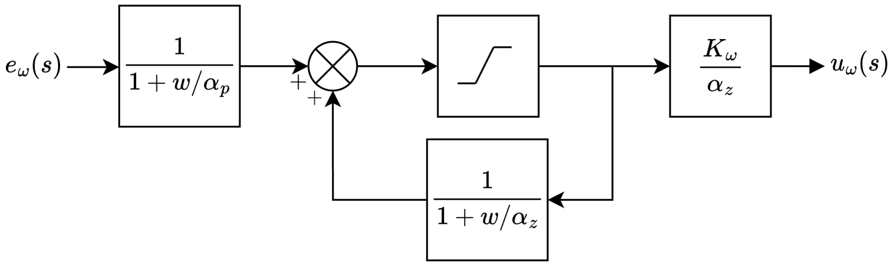


Figure 5.13: Block diagram representing modular implementation of rotor speed feedback controller from (5.12), including a saturation block to prevent integrator wind-up.

thereby removing the integral action (achieved using the positive feedback path). Based on Section 3.7.1, the minimum and maximum allowable 16-bit signals are $[13107, 26214]$, which serves as the control action limit. Working backwards, the saturation block is then required to clamp when outside the range of $\frac{\alpha_z}{K_\omega}[13107, 26214]$.

The Tustin transform [53], $w = \frac{2}{T}\frac{z-1}{z+1}$, is applied to the low-pass filter elements in Figure 5.13, followed by conversion into corresponding difference equations that can be implemented in code, namely

$$y[n] = \frac{2 - T\alpha}{2 + T\alpha} y[n-1] + \frac{T\alpha}{2 + T\alpha}\left(u[n] + u[n-1]\right), \tag{5.16}$$

where $T = 0.005$ is the sample time, α is the pole location of the particular low-pass filter element (α_p or α_z), and $u[n]$ and $y[n]$ are the respective input and output signals of the low-pass filter element at sample instance n. The remainder of the control scheme comprises linear operators (addition, subtraction, scalar multiplication), and a bipolar saturation function.

The performance of the implemented rotor speed controller is demonstrated in Figure 3.18b and 3.25b, and is shown to exhibit similar disturbance rejection qualities to that of the simulated system.

The motor speed control class is available for viewing in the floki GitHub repository.

5.2.3 Rotor thrust control

Plant template

The (estimated) linearised transfer behaviour from commanded servomotor angle (in 16-bit form) to rotor thrust (in Newtons) was presented in Section 3.8.2 and found to be

$$P_f(s) = \frac{a_f}{s^2 + 2\zeta_f\omega_f s + \omega_f^2} e^{-t_f s}, \tag{5.17}$$

with the parameter sets described in Table 5.5. The corresponding plant templates are displayed in Figure 5.14.

a_f (N/16-bit)	ζ_f	ω_f (rad/s)	t_f (s)
1 ± 0.2	0.6 ± 0.12	28 ± 5	0.005

Table 5.5: Rotor thrust plant parameters with corresponding uncertainty ranges.

As with the rotor speed control, digital sampling effects can be represented using a right-half plane zero at twice the sampling frequency (in Hz). However, given that the rotor thrust estimates (by virtue of the motor current measurements) are taken at a sampling frequency of 200 Hz (which is also equivalent to the servomotor PWM frequency), the resulting right-half plane zero will be located at 400 rad/s — far removed from the frequency range of interest for rotor thrust control (more than a decade higher than the natural frequency of the open-loop

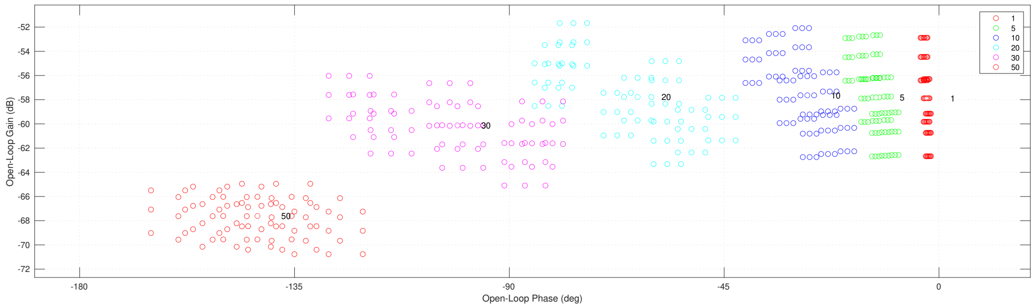


Figure 5.14: Open-loop plant template of $P_f(j\omega)$, displayed in the log-polar plane.

plant set). As such, sampling effects are treated as negligible within the expected (servomotor rate-limited) bandwidth of the rotor thrust.

Control architecture

The rotor thrust control design challenge is aimed at achieving a suitable balance between fast response (high bandwidth), and acceptable overshoot. Following from the previous section, the control loop is to operate under the assumption that the rotor speed is approximately static during operation, with the commanded servomotor angle, $\alpha_k^*(s)$, acting as the rotor thrust plant input signal. A desired rotor thrust, $f_k^*(s)$, serves as the sole signal entering the system, with the objective of driving the actual rotor thrust to the desired thrust within a suitable timeframe and performance profile ($f_k(t) \rightarrow f_k^*(s)$). With reference to Figure 5.15, the proposed control architecture comprises a feedback element, $G_f(s)$, and a feedforward filter, $X_f(s)$. $G_f(s)$ acts on the rotor thrust error, whereas $X_f(s)$ maps information from the desired thrust to the plant input. As such, the feedback controller is primarily responsible for dealing with plant uncertainty (quantified in Table 5.5) and the nonlinear hysteresis in the blade pitch actuation (see Section 3.8.1). The on-board thrust estimator (using motor current), will be prone to uncertainty and other effects. However, a comparison of the thrust estimates from the thrust rig and onboard estimator, shown in Section 4.4, infers that the two estimates are comparable in terms of transient and steady-state behaviour. As such, the design that follows assumes that the sensor dynamics are

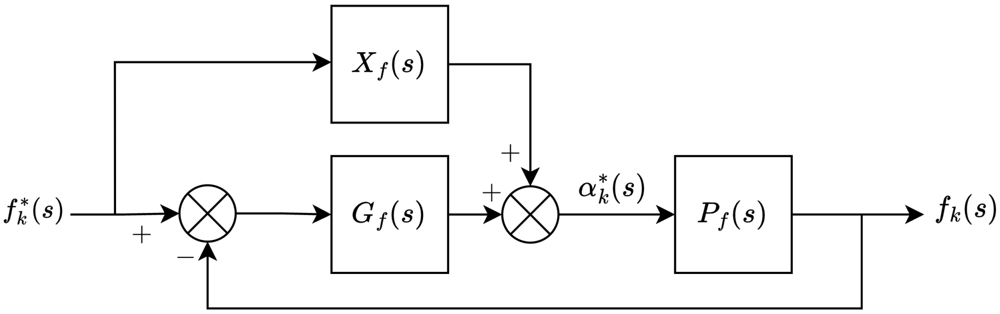
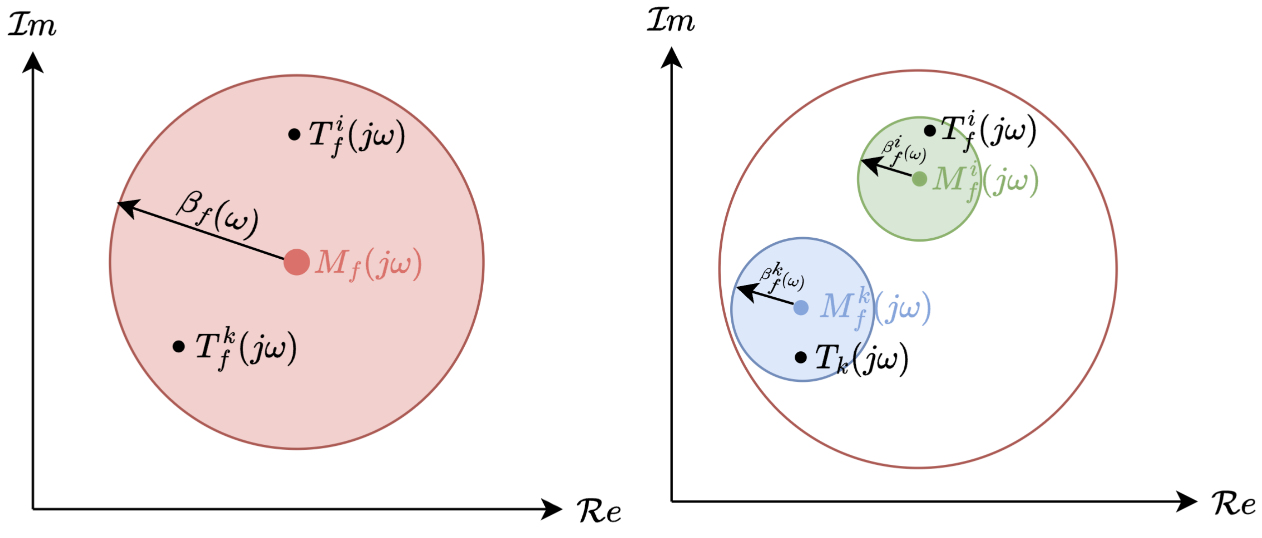


Figure 5.15: Block diagram representing proposed rotor thrust control scheme.



(a) (b)

Figure 5.16: Arithmetic-complex plane, showing geometric representation of reference model-based tracking error problem, using a (a) single-model, and (b) multi-model approach.

equal to unity ($H(s) \approx 1$). In reality, non-negligible measurement uncertainty, especially in the low frequency band, may present over time, but the outer loop attitude and position controllers, shown in Figure 5.1, will be able to account for these drift-like rotor thrust errors.

Tracking error design specifications

Design of $G_f(s)$ requires realistic performance specifications that can be used to shape the frequency-dependent profile. One such approach is to make use of an idealised transfer function model that is then compared against the actual reference-to-output transfer behaviour. Transfer function $M_f(s)$ is defined as the desired reference-to-output transfer behaviour (or reference model), and

$$T_f^i(s) = \frac{f_k^i(s)}{f_k^*(s)} = \frac{P_f^i(s)\left[G_f(s) + X_f(s)\right]}{1 + P_f^i(s)G_f(s)} \tag{5.18}$$

represents the actual closed-loop transfer behaviour from Figure 5.15 for plant instance i. The feedback control objective is to force the set of $T_f^i(s)$ elements towards $M_f(s)$ in the frequency domain to within some realistic, achievable tolerance. This can be expressed mathematically as

$$\left|M_f(j\omega) - T_f^i(j\omega)\right| \leq \beta_f(\omega),\ \forall i, \tag{5.19}$$

where β_f is the model-erorr tolerance. The geometric representation of (5.19) is shown in an arithmetic-complex plane in Figure 5.16a, whereby satisfying (5.19) for plant instance i necessitates that $T_f^i(j\omega)$ reside in the disk of centre $M_f(j\omega)$ and radius $\beta_f(\omega)$ in order to adhere to the design constraint.

With reference to Table 5.5 and Figure 3.26-3.27, the rotor thrust plant varies depending on the direction of the blade angle changes (due to the aerodynamic blade loading). Additionally, as explained in Section 3.6.3, the rotor thrust, by extension of the servomotor actuation, is expected to experience load-dependant rate limiting. An example of this is shown in Figure 5.17, where equal-sized positive and negative commanded servomotor angle steps result in different thrust rate responses of $m_+ = 37.3$ N/s and $m_- = 16.61$ N/s, respectively. Notably, the backlash in the servomotor actuation is also a contributing factor to this thrust-rate disparity. Based on the thrust response asymmetry, a per-plant reference model approach [73] is instead adopted, as shown in Figure 5.16b, whereby each plant instance has a corresponding reference model that can be specified based on realistic expectations of the particular plant's characteristics. The mathematical constraint follows as

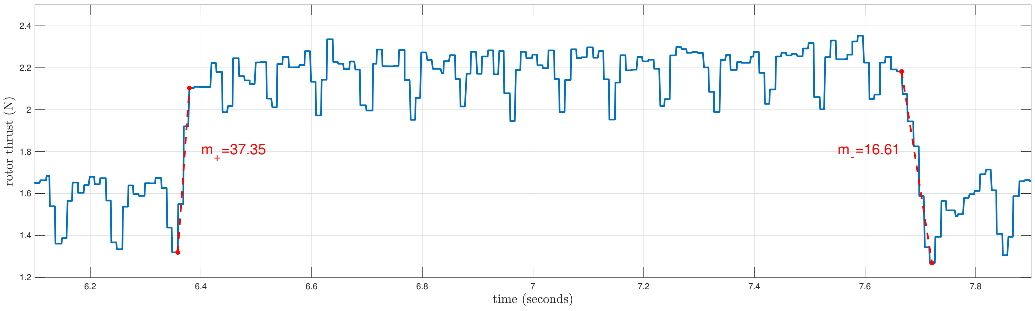


Figure 5.17: Time-domain plot.

$$\left| M_f^i(j\omega) - T_f^i(j\omega) \right| \leq \beta_f^i(\omega), \ \forall i. \tag{5.20}$$

For sake of simplicity, a common tracking tolerance is used across all plant instances ($\beta_f^i(\omega) = \beta_f(\omega), \ \forall i$). In effect, this gives each plant instance equal weighting in meeting the prescribed reference-to-output transfer behaviour — something that is absent in the single-model matching approach.

The plant-instanced reference models are selected to match the pole structure of the corresponding open-loop rotor thrust plant from Table 5.5 (implying that the rotor thrust plant models are implicitly in rate saturation from the servomotors), albeit with unitary gain

$$M_f^i(s) = \frac{(\omega_f^i)^2}{s^2 + 2\zeta_f^i \omega_f^i s + (\omega_f^i)^2} e^{-t_f s}. \tag{5.21}$$

Expanding (5.20) results in

$$\begin{aligned}\left|M_f^i(j\omega)-\frac{P_f^i(s)\left[G_f(s)+X_f(s)\right]}{1+P_f^i(s)G_f(s)}\right| &\leq \beta_f(\omega),\ \forall i,\\ \left|M_f^i(j\omega)\left[1+P_f^i(j\omega)G_f(j\omega)\right]-P_f^i(j\omega)\left[G_f(j\omega)+X_f(j\omega)\right]\right| &\leq \beta_f(\omega)\left|1+P_f^i(j\omega)G_f(j\omega)\right|,\ \forall i,\\ \left|M_f^i(j\omega)-P_f^i(j\omega)X_f(j\omega)+\left[M_f^i(j\omega)-1\right]P_f^i(j\omega)G_f(j\omega)\right| &\leq \beta_f(\omega)\left|1+P_f^i(j\omega)G_f(j\omega)\right|,\ \forall i.\end{aligned} \tag{5.22}$$

Unlike single-model design regions, the solution space of $G(j\omega)$ arising from (5.22) may not necessarily contain the infinite gain solution ($|G(j\omega)| \to \infty$). This can be shown by taking the limit as $|G(j\omega)|$ tends to infinity on either side of the constraint set in (5.22), which results in

$$\left|M_f^i(j\omega)-1\right| \leq \beta_f(\omega),\ \forall i. \tag{5.23}$$

In other words, $|G(j\omega)| \to \infty$ will only be a valid solution if (5.23) holds true for every plant instance. It is worth emphasizing that if (5.23) is invalid for any one plant case ($\left|M_f^i(j\omega)-1\right| \geq \beta_f(\omega)$), the resulting global (across all plant cases) solution space of $G(j\omega)$ will appear as a *stay-in* design region. This is because the intersection of a *stay-out* and *stay-in* design region will result in another *stay-in* design region (assuming the intersection is nonempty). The benefit of the per-plant reference model specifications is that as the model-error tolerance, $\beta_f(\omega)$, becomes smaller, the *stay-in* region monotonically decreases (and vice versa), until such time that the solution space of $G_f(j\omega)$ is represented by a single point for each design frequency. The set of point-wise design frequencies then represents the best-case fit of $G_f(j\omega)$ (for the given specifications), which may not necessarily be achievable with a causal, finite-order feedback controller. In order to assist with the pragmatic design of $G_f(j\omega)$, a level set of model-error tolerances are used, which results in a contour map of admissible design regions for each design frequency of interest. As explained in [73], this provides the designer with a transparent visualisation of the design trade-off between meeting the prescribed tolerances and increasing the feedback controller gain and complexity.

Synthesis of prototype feedforward filter, $X_f(s)$

The design of $X_f(s)$ is comparable to the approach introduced in [18]. Specifically, $X_f(s)$ is first synthesized based on a nominal plant selection, which enables direct design of $G_f(s)$. Once the feedback controller has been defined, $X_f(s)$ is then refined using the original model-error constraint set. The feedforward structure in Figure 5.15 means that selecting $X_f(s)=\frac{M_f^0(s)}{P_f^0(s)}$ will result in ideal reference to output behaviour for the nominal plant instance. Given the choice to incorporate per-plant reference models with equivalent pole locations, the feedforward filter is set as

$$X_f(s)=\frac{M_f^0(s)}{P_f^0(s)}=\frac{\omega_f^0(s)}{a_f^0}=28. \tag{5.24}$$

Following this, the tracking design constraint set of (5.22) becomes solely dependant on $G_f(j\omega)$ and allows for the generation of log-polar boundaries to guide the control practioner.

Selection of feedback controller, $G_f(s)$

Based on the describing function method, the nonlinear backlash can be represented by stretching the critical point in the log-polar point, such that it represents a critical line spanning from $0\text{dB}\angle-180^\circ$ to $\infty\text{dB}\angle-90^\circ$. If the open-loop transfer function frequency response crosses the critical line, the system will experience limit cycling. Given that the open-loop plant is Type-0, the feedback controller can contain up to one integrator without risk of crossing the critical line (and inducing limit cycling or instability). This also highlights the benefit of using thrust control, as loop closure from the quadcopter attitude level only would entail two integration operations, which would cause the loop transfer function to cross the critical line and induce limit cycling.

The level set approach requires selection of the $\beta(\omega)$ values. This is performed by assessing the resulting log-polar boundaries that arise from (5.22) and then altering the tolerances in order to generate a set of per-frequency design regions that are feasible in terms of gain-phase constraints. The selected model-error tolerances are summarised in Table 5.6. Following this, (5.22) is converted into design boundaries on $G(j\omega)$ for each tolerance in Table 5.6. The tracking error design regions are shown in Figure 5.18 using an arbitrarily selected nominal plant to present the design regions in the form of the classical Nichols chart. As seen in Figure 5.18, the per-frequency *stay-in* design regions expand as the $\beta_f(\omega)$ is increased based on Table 5.6. At such time that (5.23) is true across all plant cases, the particular design regions opens up, resulting in a *stay-out* design region. This can be seen for the low- to mid-frequency specifications of $\omega < 50$ rad/s.

Level set	$\beta_f(0.1)$	$\beta_f(1)$	$\beta_f(5)$	$\beta_f(10)$	$\beta_f(20)$	$\beta_f(30)$	$\beta_f(50)$	$\beta_f(80)$
1	0.006	0.06	0.25	0.45	0.8	1.04	0.3	0.15
2	0.0075	0.075	0.31	0.56	1	1.3	0.38	0.19
3	0.009	0.093	0.39	0.70	1.24	1.61	0.47	0.23

Table 5.6: Level set of model-reference tolerances used over discrete frequency range of interest during design of $G_f(s)$.

Robust disturbance rejection requires overbounding of the closed-loop sensitivity, which is enforced with a 6 dB tolerance

$$\left|\frac{1}{1+P_f^i(j\omega)G_f(j\omega)}\right| \leq 10^{6/20}, \forall i. \tag{5.25}$$

The resulting sensitivity bounds are displayed in Figure 5.19. After consideration of the design regions in Figure 5.18 and Figure 5.19, the feedback controller is selected as

$$G_f(s) = 10628\frac{(1+s/26)(1+s/44)}{s(1+s/247)}. \tag{5.26}$$

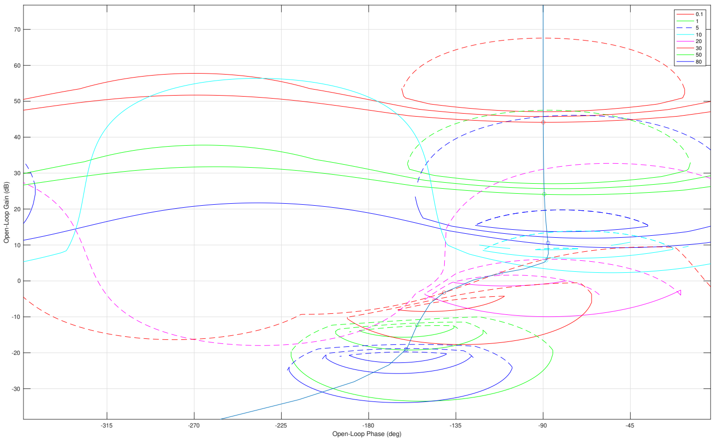


Figure 5.18: Log-polar plane, showing admissible design regions of $G(j\omega)$ arising from (5.22). Solid lines indicated *stay-above* boundaries, whereas dotted lines show *stay-below* boundaries.

The corresponding nominal open-loop frequency response, $P_f^0(j\omega)G_f(j\omega)$, is also shown in Figure 5.18 and Figure 5.19 for the design frequencies of interest. The rotor thrust controller in (5.26) takes the form of a PID controller with realisable differential action implemented as a lead element (albeit formed naturally based on the frequency-domain boundaries).

Selection of feedforward filter, $X_f(s)$

The prototype feedforward filter synthesis, $X_f(j\omega)$, serves as a simple means of proceeding with the univariate feedback control design stage, but the sub-optimal selection places conservatism on the design regions of $G_f(s)$, which limits the achievable model-error tolerance. However, following the designation of $G_f(s)$, one can make use of (5.22) to design $X_f(s)$ in earnest, which will generate *stay-in* design regions at each frequency of interest. At this stage, the model-error tolerance, $\beta(\omega)$, is also adjusted as needed, depending on the design rigour in joining the various frequency-dependant *stay-in* solution spaces. Following visual inspection of the resulting design regions of $X_f(j\omega)$, the tracking tolerances are specified as shown in Table 5.7. The corresponding log-polar design regions of $X_f(j\omega)$ are shown in Figure 5.20, along with the frequency response of the selected feedforward filter of

Level set	$\beta_f(0.1)$	$\beta_f(1)$	$\beta_f(5)$	$\beta_f(10)$	$\beta_f(20)$	$\beta_f(30)$	$\beta_f(50)$	$\beta_f(80)$
1	0.007	0.06	0.29	0.45	0.72	0.8	0.24	0.08
2	0.0075	0.0750	0.3125	0.62	0.92	0.9	0.285	0.125

Table 5.7: Level set of model-reference tolerances used over discrete frequency range of interest during design of $X_f(s)$.

$$X_f(s) = \frac{358}{s^2/21^2 + 2(0.38/21)s + 1}. \tag{5.27}$$

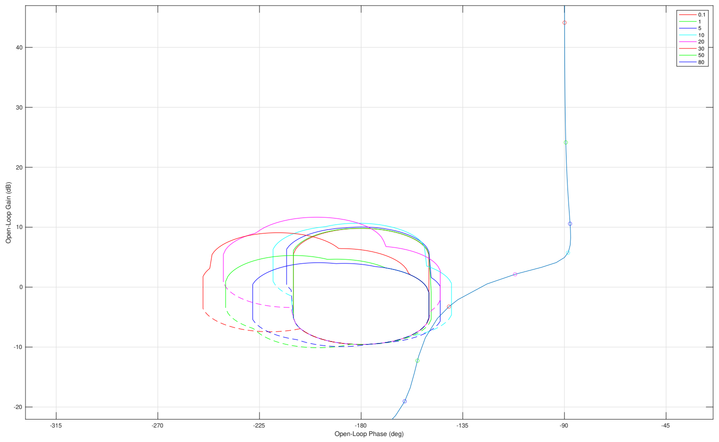


Figure 5.19: Log-polar plane, showing admissible design regions of $G(j\omega)$ arising from sensitivity constraint in (5.25). Solid lines indicated *stay-above* boundaries, whereas dotted lines show *stay-below* boundaries.

The DC gain of $X_f(s)$ varies significantly to that of the prototype design in (5.24), which suggests that the feedforward element is ineffectual at low-frequencies (where the integral feedback control action will predominantly contribute to the control action. The benefit of $X_f(s)$ is in the mid-frequency gain contribution, near the peak frequency of $P_f(s)$ based on Table 5.5.

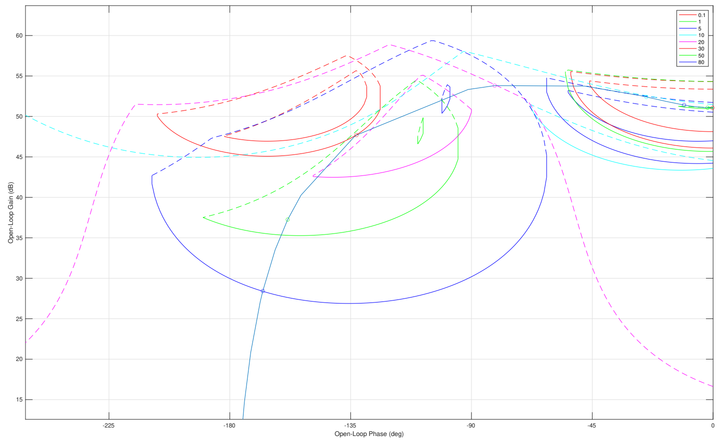


Figure 5.20: Log-polar plane, showing admissible design regions of $X(j\omega)$ arising from tracking error constraint in (5.22). Solid lines indicated *stay-above* boundaries, whereas dotted lines show *stay-below* boundaries.

Time-domain simulation and implementation

Following the selection of $G_f(s)$ and $F(s)$, the closed-loop performance can be ratified by comparing the step response behaviour to that of the plant-dependant reference models. Note that the use of multiple reference models and the level set of model-error tolerances means that assessing (5.23) using a Bode magnitude response (commonplace in QFT designs) is not useful. The representative subset of step responses of the reference models and corresponding reference to output transfer behaviours are displayed in Figure 5.21. Notably, the simulation in Figure 5.21 assumes the plant set is linear and does not encapsulate servomotor saturation or backlash effects. Implementation of the thrust control scheme on the quadcopter microcontroller follows a similar process to that of the rotor speed control, such as using Shinskey's anti-reset windup approach [72], and conversion into appropriate difference equations. Following this, the time-domain performance of the thrust controller is assessed by applying thrust variations within the expected operating region of the quadcopter during flight. The resulting closed-loop rotor thrust responses are shown in Figure 5.22 and Figure 5.23, for two different time periods. The transient behaviour of the rotor thrust responses in Figure 5.22 and Figure 5.23 are comparable to the simulated behaviour in Figure 5.21 when operating in the vicinity of the hover thrust (f_0 = 1.55 N), where the plant modelling was performed. Increased overshoot and oscillatory behaviour can be seen in Figure 5.23 when the reference rotor thrust becomes sufficiently small (< 1 N). This is presumably as a result of the commanded servomotor angle to thrust mapping from Figure 3.21a, which behaves increasingly

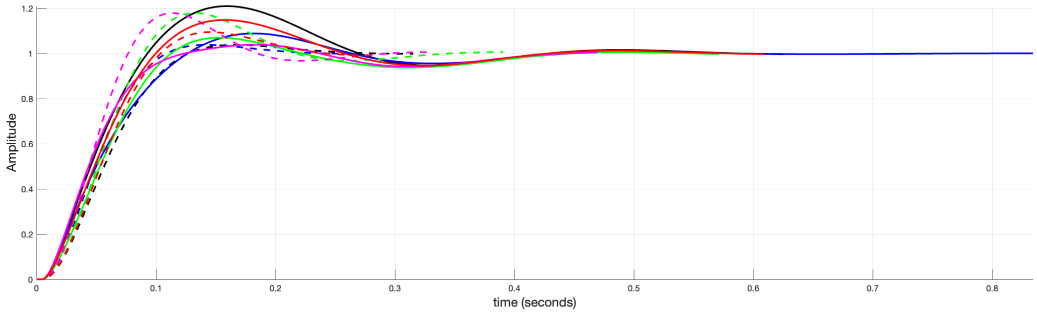


Figure 5.21: Time-domain comparison of per-plant reference model step responses (dotted lines) and simulated closed-loop responses (solid lines) for a representative subset of the closed-loop rotor thrust plant set. A particular colour corresponds to a model and plant instance pair

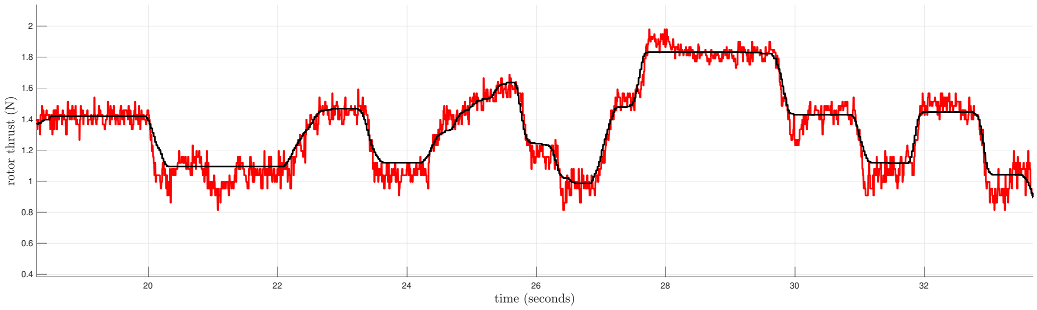


Figure 5.22: Time-domain plot showing rotor thrust reference signal (black) and corresponding closed-loop thrust response (red) for time period of $t \in [18, 34]$ seconds. The rotor thrust measurement is obtained from the thrust test rig.

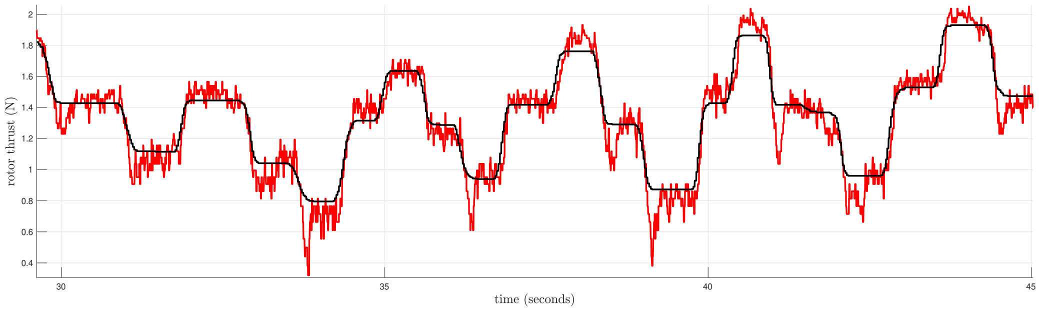


Figure 5.23: Time-domain plot showing rotor thrust reference signal (black) and corresponding closed-loop thrust response (red) for time period of $t \in [30, 45]$ seconds.

nonlinear as the rotor thrust approaches zero, as well as the aforementioned backlash effects.

5.3 Controller command to desired rotor thrust mapping

With reference to Section 5.2.3, the closed-loop rotor thrust serves as the inner-most loop of the overarching quadcopter control structure. The rotor thrust control loop is predicated on knowledge of the desired rotor thrust, f_k^*, which up until this point has been treated as a directly controllable quantity. In order to link the rotor thrust loop with that of the quadcopter heave and attitude systems, one can first make use of (3.45) and (3.51) to define the on-board control action

$$\mathbf{u} = \begin{bmatrix} u_h \\ \mathbf{u}_\theta \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{4} f_k^* \\ l(f_4^* - f_2^*) \\ l(f_3^* - f_1^*) \\ \sum_{k=1}^{4} (-1)^{k+1} \tau_k^* \end{bmatrix}, \tag{5.28}$$

where l is the rotor lever arm (the distance from each rotor axis to the quadcopter centre of mass), introduced in Section 3.4.2. $u_h = {}^{B}h^*$ represents the open-loop control action from the heave controller (equivalent to the desired body-frame heave, ${}^{B}h^*$), and $\mathbf{u}_\theta$ is the control action for the attitude controller. Equation (5.28) contains the desired rotor torque, τ_k^*, which does not have a corresponding torque control loop. With reference to Section 4.4, a suitable mapping was found that relates motor current to rotor thrust. The rotor torque is directly proportional to the motor current over the operating region of interest., based on (3.70). In addition, the relationship between blade angle and both lift and drag is well characterised from aerodynamics based on Section 3.3. As a result, there is a well defined mapping between rotor thrust and motor current. The steady-state rotor thrust is plotted against the rotor torque in Figure 5.24 for the expected servomotor angle operation region,

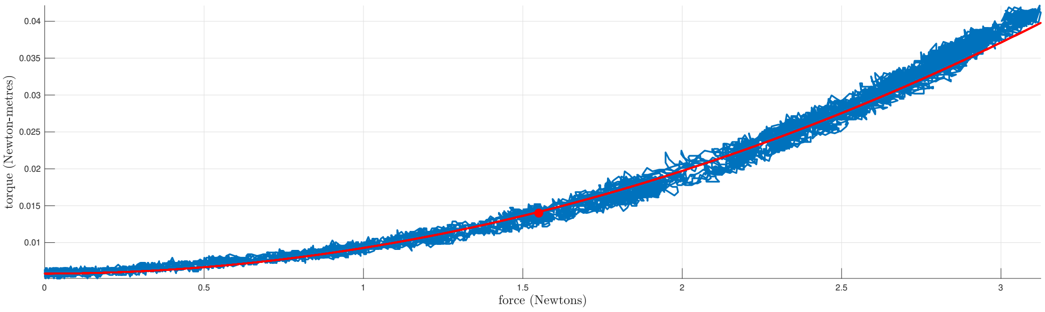


Figure 5.24: Steady-state rotor thrust to torque relationship (blue), measured on thrust test rig. A quadratic line of best fit is also shown (red), as well as the hover thrust-torque point (red dot).

and is shown to exhibit the approximate (pure) quadratic relationship of

$$\begin{aligned}\tau_k &\approx c_0 + c_1 f_k^2, \\ &\approx 5.8 \times 10^{-3} + 3.5 \times 10^{-3} f_k^2.\end{aligned} \tag{5.29}$$

The generated rotor torque at zero lift ($f_k = 0$), c_0, will be approximately constant across all four rotors. Given that the quadcopter uses two pairs of counter-rotating rotors, the net zero-lift torque (across all four rotors) sums to zero. As such, substituting the relationship in (5.29) into (5.28) results in

$$\mathbf{u} = \begin{bmatrix} u_h \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} \approx \begin{bmatrix} \sum_{k=1}^{4} f_k^* \\ l(f_4^* - f_2^*) \\ l(f_3^* - f_1^*) \\ c_1 \sum_{k=1}^{4} (-1)^{k+1} (f_k^*)^2 \end{bmatrix}. \tag{5.30}$$

The desired per-rotor forces can be extracted by inverting the relationship in (5.30), which yields

$$\mathbf{f}^* = \begin{bmatrix} f_1^* \\ f_2^* \\ f_3^* \\ f_4^* \end{bmatrix} = {}^{\mathbf{f}^*}\mathcal{M}_{\mathbf{u}}(\mathbf{u}) = \frac{1}{4 c_1 l^2 u_h} \begin{bmatrix} c_1 l^2 u_h^2 + 2 l^2 u_\psi - 2 c_1 l u_h u_\theta + c_1 u_\phi^2 - c_1 u_\theta^2 \\ c_1 l^2 u_h^2 - 2 l^2 u_\psi - 2 c_1 l u_h u_\phi - c_1 u_\phi^2 + c_1 u_\theta^2 \\ c_1 l^2 u_h^2 + 2 l^2 u_\psi + 2 c_1 l u_h u_\theta + c_1 u_\phi^2 - c_1 u_\theta^2 \\ c_1 l^2 u_h^2 - 2 l^2 u_\psi + 2 c_1 l u_h u_\phi - c_1 u_\phi^2 + c_1 u_\theta^2 \end{bmatrix}, \tag{5.31}$$

as shown compactly in Figure 5.1. A conditional singularity exists when the commanded quadcopter heave thrust, u_h, is equal to zero. However, in practice this is not problematic, as the nominal heave thrust is equal to mg (the quadcopter weight) and "typical" flights are not expected to require a heave thrust that approaches zero. Note that if the commanded roll, pitch, and yaw moments are negligible ($u_\phi = u_\theta = u_\psi = 0$), the mapping in (5.31) will not possess a singularity.

5.4 Simplifying the attitude dynamics

With reference to (3.56), the high-level attitude dynamics (ignoring unmodelled disturbance effects) are described by

$$\mathbf{J}^B\dot{\mathbf{\Omega}} = {}^B\mathbf{m} - {}^B\mathbf{\Omega} \times \mathbf{J}^B\mathbf{\Omega}. \tag{5.32}$$

Note that the simplified representation in (5.32) is only valid when the rotor speeds are all equivalent ($|\omega_1| = |\omega_2| = |\omega_3| = |\omega_4|$) which is suitably enforced using the rotor speed control detailed in Section 5.2.2. Given that the mass moment of inertia matrix from (3.55) is diagonally dominant (with a Perron root of $0.04 \ll 1$), (5.32) can be represented as

$$\begin{bmatrix} J_{xx}\dot{\Omega}_x \\ J_{yy}\dot{\Omega}_y \\ J_{zz}\dot{\Omega}_z \end{bmatrix} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} - \begin{bmatrix} (J_{zz} - J_{yy})\Omega_y\Omega_z \\ (J_{xx} - J_{zz})\Omega_x\Omega_z \\ (J_{yy} - J_{xx})\Omega_x\Omega_y \end{bmatrix}, \tag{5.33}$$

where

$$\mathbf{J} \approx \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} = \begin{bmatrix} 3.016 & 0 & 0 \\ 0 & 2.956 & 0 \\ 0 & 0 & 5.290 \end{bmatrix} \times 10^{-3}. \tag{5.34}$$

Noting that the quadcopter inertia is symmetric about $\mathcal{X}^b$ and $\mathcal{Y}^b$ ($J_{yy} \approx J_{xx}$), imposing the constraint of $\Omega_z \approx 0$ simplifies the dynamic relationship in (5.33) to

$$\begin{bmatrix} J_{xx}\dot{\Omega}_x \\ J_{yy}\dot{\Omega}_y \\ J_{zz}\dot{\Omega}_z \end{bmatrix} = \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix}. \tag{5.35}$$

Note that the simplification in (5.35) is, in general, valid when $\Omega_x\Omega_z \approx \Omega_y\Omega_z \approx 0$, which suggests that yaw actuation is not problematic when the pitching and rolling is negligible. The body-frame yaw-rate constraint of $\Omega_z \approx 0$ will remove a degree of freedom from the system. However, generalised position control is still achievable using the three remaining degrees of freedom (heave, roll, and pitch). In fact, based on the relatively large J_{zz} term from (5.34), along with the relatively low achievable rotor torque production (based on Figure 3.21b), the yaw channel has the smallest effective gain, making it the least effective channel for pose corrections.

With the body-frame yaw-rate constraint in place, the body-frame roll and pitch channels become solely responsible for correcting the orientation of the quadcopter. As a result, the attitude control task transitions from resolving the 3D orientation, to a normalised thrust vectoring problem. In other words, seeking to orient the ${}^B\mathcal{Z}$-axis in the correct inertial-frame direction (based on tracking some desired inertial-frame setpoint). For more information, see Section 5.6. The angular yaw behaviour becomes of no consequence to the operation of the quadcopter, as long as the body-frame yaw-rate is sufficiently small to not induce cross-coupling from (5.33). For this reason, it makes sense to close the loop around the yaw-rate, as modelling imperfections and disturbances will otherwise cause unbalanced torques on the yaw channel. The resulting body-frame yaw angular behaviour is likely to slowly vary, but this has no effect on the thrust vectoring capability of the quadcopter (as long as the pose can be estimated with sufficient fidelity and the thrust vectoring is correctly mapped from the inertial frame to the body frame.).

5.5 Determining the attitude plants

5.5.1 Rotor thrust plant set

The rotor thrust control approach in Section 5.2.3 facilitates a regulated input-output relationship between desired thrust and achieved thrust. The closed-loop rotor thrust plant is expected to contain non-negligible uncertainty, as evidenced in the varying time-domain responses from Figure 5.22 and 5.23. Additionally, the rotor thrust control loop is only able to regulate the motor-generated thrust, and does not account for thrust from external sources (such as aerodynamic drag). Following this, system identification methods were used to deduce an appropriate plant set that captures the closed-loop rotor thrust behaviour under the various operation conditions. A similar process to the system identification in Chapter 3 is used, whereby a first-pass model is determined using software to obtain the general pole-zero structure, followed by manual refinement to develop the plant set. Closed-loop thrust plant instance i follows as

$$T_f^i = \frac{a_{T_f}^i \left(1 + s/z_{T_f}^i\right)}{s^2/{\omega_{T_f}^i}^2 + 2\zeta_{T_f}^i/\omega_{T_f}^i s + 1} e^{-0.005s}, \tag{5.36}$$

with the corresponding parameters uncertainty bands shown in Table 5.8. The system response of two closed-loop plant instances is shown in Figure 5.25. As with the open-loop rotor thrust plant, the closed-loop thrust behaviour is asymmetrical and responds differently depending on whether the blade angle needs to actuate with or against the aerodynamic loading. The closed-loop plant set captures this substantial plant variation, mainly using the minimum-phase zero, z_{T_f}.

a_{T_f} (N/N)	z_{T_f} (rad/s)	ω_{T_f} (rad/s)	ζ_{T_f}
1 ± 0.1	4.4 ± 0.6	11.35 ± 4	0.87 ± 0.2

Table 5.8: Closed-loop rotor thrust plant parameters with corresponding uncertainty bands.

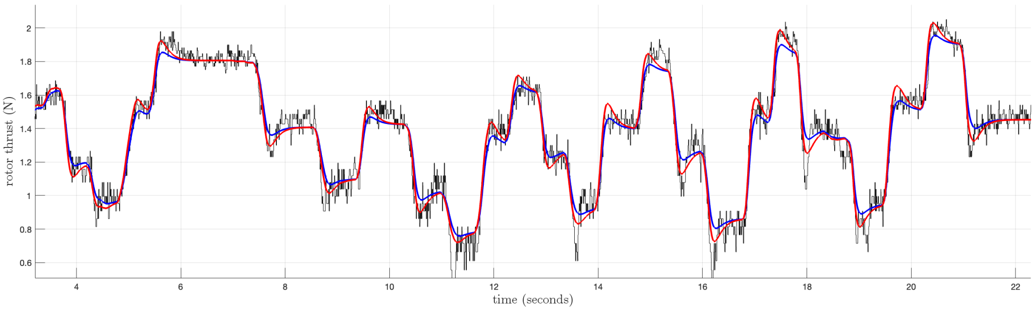


Figure 5.25: Time-domain plot, showing the measured rotor thrust (black). The two extreme closed-loop system responses from the plant set are also shown, for $z_{T_f} = 5.0$ rad/s (blue), and $z_{T_f} = 3.8$ rad/s (red) from Table 5.8.

5.5.2 Roll and pitch channel

With reference to (3.51), (5.35), and Section 5.2.3, the open-loop transfer behaviour of the body-frame roll- and pitch-rate plants can be written explicitly as

$$\begin{bmatrix} \Omega_x(s) \\ \Omega_y(s) \end{bmatrix} = \begin{bmatrix} \frac{m_x(s)}{J_{xx}} \\ \frac{m_y(s)}{J_{yy}} \end{bmatrix} = \begin{bmatrix} \frac{l}{J_{xx}s}\left(T_f^i(s) f_4{}^*(s) - T_f^k(s) f_2{}^*(s)\right) \\ \frac{l}{J_{yy}s}\left(T_f^i(s) f_3{}^*(s) - T_f^k(s) f_1{}^*(s)\right) \end{bmatrix}, \ \forall i, k, \tag{5.37}$$

where $T_f^i(s)$ and $T_f^k(s)$ are two distinct plant instances within the closed-loop rotor thrust plant set, as defined in Section 5.5.1. Both the roll and pitch channels incorporate rotor thrust from two opposing rotors (rotor 2 and 4 for roll, and rotor 1 and 3 for pitch), which necessitates quantifying the potential plant variations across the opposing rotors. This is exemplified by considering a desired positive roll motion about ${}^B\mathcal{X}$, which would require that f_4 increase and f_2 decrease. By extension, this would demand an increase in the blade angle of rotor 4 ($\Delta\sigma_4 > 0$), and decrease in the blade angle of rotor 2 ($\Delta\sigma_2 < 0$). With reference to Figure 5.25, this would result in different transient behaviours for the two rotors (as σ_4 rotates in the direction of the aerodynamic loading and σ_2 rotates against it).

With reference to Figure 5.1, we seek the transfer behaviour from the defined roll and pitch input signal of $\{u_\phi(s), u_\theta(s)\}$ to the respective angular rate outputs of $\{\Omega_x(s), \Omega_y(s)$. Refering to (5.30), the roll command, $u_\phi(s)$, can be parameterised as

$$u_\phi(s) = l\left[f_4{}^*(s) - f_2{}^*(s)\right] = \gamma_\phi u_\phi(s) - (\gamma_\phi - 1)u_\phi(s), \tag{5.38}$$

which implies

$$\begin{aligned} l f_4{}^*(s) &= \gamma_\phi u_\phi(s), \\ l f_2{}^*(s) &= (\gamma_\phi - 1)u_\phi(s), \end{aligned} \tag{5.39}$$

with $\gamma_\phi \in [0, 1]$. Solving (5.39) in terms of the desired thrusts and substituting into the first row of (5.37) yields

$$\Omega_x(s) = P^{ik}_{\Omega_x}(s)u_\phi(s) = \frac{T^i_f(s)\gamma_\phi - T^k_f(s)\left[\gamma_\phi - 1\right]}{J_{xx}s}u_\phi(s), \tag{5.40}$$

where $P^{ik}_{\Omega_x}(s)$ represents the transfer behaviour from the desired torque about ${}^B\mathcal{X}$, $u_\phi(s)$, to the body-frame roll-rate, $\Omega_x(s)$, for plant pair $\{i, k\}$. γ_ϕ serves as an additional uncertain parameter to sweep through, in order to sufficiently map the uncertainty profile from the appropriate rotors, to the roll channel. This is very important, as the load sharing between $f_4(s)$ and $f_2(s)$ will vary depending on the reference signal and current pose and pose rate of the quadcopter. The transfer behaviour in the pitch channel (from desired torque to the corresponding angular rate) is similarly expressed as

$$\Omega_y(s) = P^{ik}_{\Omega_y}(s)u_\theta(s) = \frac{T^i_f(s)\gamma_\theta - T^k_f(s)\left[\gamma_\theta - 1\right]}{J_{yy}s}u_\theta(s), \tag{5.41}$$

where $\gamma_\theta \in [0, 1]$. Following this, the set of body-frame roll- and pitch-rate plants are described as

$$\begin{aligned}\frac{\Omega_x(s)}{u_\phi(s)} &= P^{ik}_{\Omega_x}(s) = \frac{T^i_f(s)\gamma_\phi - T^k_f(s)\left[\gamma_\phi - 1\right]}{J_{xx}s}, \ \forall i, k,\\ \frac{\Omega_y(s)}{u_\theta(s)} &= P^{ik}_{\Omega_y}(s) = \frac{T^i_f(s)\gamma_\theta - T^k_f(s)\left[\gamma_\theta - 1\right]}{J_{yy}s}, \ \forall i, k.\end{aligned} \tag{5.42}$$

By extension, the relationship between the desired torques and corresponding incremental body-frame angular displacements is determined by

$$\begin{aligned}\frac{\Delta\phi(s)}{u_\phi(s)} &= P^{ik}_{\Delta\phi}(s) = \frac{T^i_f(s)\gamma_\phi - T^k_f(s)\left[\gamma_\phi - 1\right]}{J_{xx}s^2}, \ \forall i, k,\\ \frac{\Delta\theta(s)}{u_\theta(s)} &= P^{ik}_{\Delta\theta}(s) = \frac{T^i_f(s)\gamma_\theta - T^k_f(s)\left[\gamma_\theta - 1\right]}{J_{yy}s^2}, \ \forall i, k.\end{aligned} \tag{5.43}$$

Notably, (5.42) can be used to represent the open-loop plant if angular rate is the output of interest, whereas (5.43) is relevant when regulating the orientation.

5.5.3 Yaw-rate plant

With reference to (5.30), the yaw channel control action is equal to the sum of the rotor torques, which is related to the sum of the squares of the rotor thrusts

$$u_\psi = c_1 \sum_{k=1}^{4}(-1)^{k+1}(f^*_k)^2. \tag{5.44}$$

This relationship is first linearised around the hover condition ($f_k = f_0$), which gives

$$\begin{aligned} u_\psi &\approx c_1 \sum_{k=1}^{4} (-1)^k (f_0^*)^2 + 2c_1 f_0 \sum_{k=1}^{4} (-1)^{k+1} (f_k^*), \\ &\approx 0 + 2c_1 f_0 \sum_{k=1}^{4} (-1)^{k+1} (f_k^*). \\ &\approx 2c_1 f_0 \left(f_1^* - f_2^* + f_3^* - f_4^*\right). \end{aligned} \tag{5.45}$$

Applying the same logic to (5.35), the linearised yaw-rate plant set description is

$$\Omega_z(s) = \frac{m_z(s)}{J_{zz}s} \approx \frac{2c_1 f_0}{J_{zz}s} \left[T_f^a(s) f_1^*(s) - T_f^b(s) f_2^*(s) + T_f^c(s) f_3^*(s) - T_f^d(s) f_4^*(s)\right], \; \forall a, b, c, d. \tag{5.46}$$

Similar to (5.38), (5.45) can be parameterised as

$$u_\psi = 2c_1 f_0 \left(f_1^* + f_3^*\right) - 2c_1 f_0 \left(f_2^* + f_4^*\right) = \gamma_\psi u_\psi - \left(\gamma_\psi - 1\right) u_\psi(s), \tag{5.47}$$

which implies

$$\begin{aligned} 2c_1 f_0 \left(f_1^* + f_3^*\right) &= \gamma_\psi u_\psi, \\ 2c_1 f_0 \left(f_2^* + f_4^*\right) &= \left(\gamma_\psi - 1\right) u_\psi, \end{aligned} \tag{5.48}$$

with $\gamma_\psi \in [0, 1]$. The relationships in (5.48) can be further parameterised as

$$\begin{aligned} 2c_1 f_0 f_1^* &= \beta_1 \gamma_\psi u_\psi, \\ 2c_1 f_0 f_3^* &= (1 - \beta_1)\, \gamma_\psi u_\psi, \end{aligned} \tag{5.49}$$

and

$$\begin{aligned} 2c_1 f_0 f_2^* &= \beta_2 \left(\gamma_\psi - 1\right) u_\psi, \\ 2c_1 f_0 f_4^* &= (1 - \beta_2) \left(\gamma_\psi - 1\right) u_\psi, \end{aligned} \tag{5.50}$$

where $\beta_1, \beta_2 \in [0, 1]$. Solving for f_k^* and substituting the results into (5.46) gives

$$\Omega_z(s) = \frac{1}{J_{zz}s} \left[T_f^a(s)\beta_1\gamma_\psi - T_f^b(s)\,(1 - \beta_1)\,\gamma_\psi + T_f^c(s)\beta_2 \left(1 - \gamma_\psi\right) - T_f^d(s)\,(1 - \beta_2)\left(\gamma_\psi - 1\right)\right] u_\psi, \tag{5.51}$$

and by extension

$$\frac{\Omega_z(s)}{u_\psi} = P_{\Omega_z}^{abcd}(s) = \frac{1}{J_{zz}s} \left[T_f^a(s)\beta_1\gamma_\psi - T_f^b(s)\,(1 - \beta_1)\,\gamma_\psi + T_f^c(s)\beta_2 \left(1 - \gamma_\psi\right) - T_f^d(s)\,(1 - \beta_2)\left(\gamma_\psi - 1\right)\right], \tag{5.52}$$

The newly introduced parameters, γ_ψ, β_1, and β_2, are used to generate the set of plant cases, which depend on the yaw torque contribution of each rotor. The linearisation introduced in (5.45) is expected to result in an imperfect description of the yaw-rate plant set. However, because the yaw-rate is to be regulated at zero (using integral control), the implications are expected not to affect the quadcopter performance significantly.

5.6 Thrust vectoring

With reference to Section 3.4.1, the quadcopter is only able to produce a controllable thrust in the body-frame z-axis, ${}^{B}\mathcal{Z}$. However, when combining the controllable orientation of the quadcopter (as a result of the differential thrusts from each rotor), the thrust vector can be pointed in any arbitrary direction (albeit with an associated settling time). This insight is used as part of the control allocation from the outer loop position control (in the inertial-frame), to the inner loop heave and attitude controllers.

Equation (3.47) is repeated below for sake of convenience, after omitting the unmodelled disturbance vector,

$$\begin{aligned} m{}^{W}\ddot{\mathbf{p}} &= {}^{W}\mathbf{R}_{B}{}^{B}\mathbf{f}_{\Sigma} + m{}^{W}\mathbf{g}, \\ &= {}^{W}\mathbf{f}_{\Sigma} + m{}^{W}\mathbf{g}. \end{aligned} \tag{5.53}$$

Given a desired translational acceleration, ${}^{W}\ddot{\mathbf{p}}^{*}$, (for example from a trajectory planner, such as one detailed in Appendix B), the corresponding desired inertial-frame thrust vector follows as

$${}^{W}\mathbf{f}_{\Sigma}{}^{*} = m{}^{W}\ddot{\mathbf{p}}^{*} - m{}^{W}\mathbf{g}. \tag{5.54}$$

Uncertainty in the various system parameters, as well as unmodelled disturbances acting on the translational system (such as aerodynamic drag) means that the relationship in (5.54) may not always hold true. To counteract this, the control action from the position controller, $\mathbf{u}_p$, is augmented to the desired inertial-frame thrust vector description

$${}^{W}\mathbf{f}_{\Sigma}{}^{*} = m{}^{W}\ddot{\mathbf{p}}^{*} - m{}^{W}\mathbf{g} + \mathbf{u}_p. \tag{5.55}$$

The control signal, $\mathbf{u}_p$, is generated from the position controller, $\mathbf{G_p}$, as shown in Figure 5.1. ${}^{W}\mathbf{f}_{\Sigma}{}^{*}$ contains information for both the desired body-frame heave, ${}^{B}h^{*}$, and desired z-axis configuration in the inertial frame, ${}^{W}\mathcal{Z}^{*}$. The desired heave is determined by calculating the magnitude of the desired inertial-frame thrust vector

$${}^{B}h^{*} = \left|{}^{W}\mathbf{f}_{\Sigma}{}^{*}\right|, \tag{5.56}$$

and is used as the heave control action, $u_h = {}^B h^*$, that is sent to the rotor thrust mapping from (5.31). The desired inertial-frame z-axis configuration is obtained by normalising the desired inertial-frame thrust vector

$$ {}^W\mathcal{Z}^* = \frac{{}^W\mathbf{f}_\Sigma{}^*}{\left|{}^W\mathbf{f}_\Sigma{}^*\right|} = \frac{{}^W\mathbf{f}_\Sigma{}^*}{{}^B h^*}. \tag{5.57} $$

Notably, vector ${}^W\mathcal{Z}^*$ can also be derived using

$$ \begin{aligned} {}^W\mathcal{Z}^* &= {}^W\mathbf{R}_D{}^D\mathcal{Z} \\ &= \mathbf{R}({}^W\mathbf{q}_B^*)^D\mathcal{Z} \\ &= \mathbf{R}({}^W\mathbf{q}_B^*)\begin{bmatrix}0\\0\\1\end{bmatrix}, \end{aligned} \tag{5.58} $$

where ${}^W\mathbf{q}_B^* = {}^W\mathbf{q}_D$ describes the desired (quaternion-encoded) orientation of frame $\{B\}$ with respect to frame $\{W\}$, and ${}^D\mathcal{Z} = [0\ 0\ 1]^T$ is the z-axis of frame $\{D\}$.

5.6.1 Deriving a yaw-free body-frame angular error

Following the design choice in Section 5.4 of regulating the body-frame yaw-rate at zero ($\Omega_z \approx 0$), a suitable angular error needs to be defined that does not activate the yaw channel. A yaw-free angular error in the body-frame corresponds to finding the rotation vector that rotates the body-frame z-axis, ${}^B\mathcal{Z} = [0\ 0\ 1]^T$, onto the desired body-frame z-axis, ${}^B\mathcal{Z}^*$,

$$ \begin{aligned} {}^B\boldsymbol{\Theta}_e &: {}^B\mathcal{Z} \to {}^B\mathcal{Z}^*, \\ &: \begin{bmatrix}0\\0\\1\end{bmatrix} \to {}^B\mathcal{Z}^*. \end{aligned} \tag{5.59} $$

${}^B\mathcal{Z}^*$ can be formed by mapping ${}^W\mathcal{Z}^*$ into the body frame, which is expressed as

$$ \begin{aligned} {}^B\mathcal{Z}^* &= \mathbf{R}^T({}^W\mathbf{q}_B)^W\mathcal{Z}^*, \\ &= \mathbf{R}({}^B\mathbf{q}_W)^W\mathcal{Z}^*, \\ &= \mathbf{R}({}^B\mathbf{q}_W)\mathbf{R}({}^W\mathbf{q}_B^*)^D\mathcal{Z}, \\ &= \mathbf{R}({}^B\mathbf{q}_B^*)^D\mathcal{Z}, \\ &= \mathbf{R}({}^B\mathbf{q}_B^*)\begin{bmatrix}0\\0\\1\end{bmatrix}. \end{aligned} \tag{5.60} $$

${}^{B}\mathbf{q}_{B}^{*}$ describes the desired body-frame orientation with respect to the body frame, thereby encoding the orientation change required to align ${}^{B}\mathcal{Z}$ with ${}^{B}\mathcal{Z}^{*}$, albeit while potentially activating the yaw channel. Note that ${}^{B}\mathcal{Z}^{*}$ in (5.60) can either be populated using (5.57) (which would originate from the position controller), or using the desired inertial-frame quaternion, ${}^{W}\mathbf{q}_{B}^{*}$ (which could be the case if the quadcopter is flown in attitude control mode only without outer loop control). Regardless, knowledge of the current inertial-frame quaternion, ${}^{W}\mathbf{q}_{B}$, is required in both cases to map the information from the inertial frame into the body frame.

Using (5.60), (5.59) can be written explicitly as

$$ {}^{B}\mathbf{\Theta}_{e} \ : \ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow \mathbf{R}({}^{B}\mathbf{q}_{B}^{*}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5.61} $$

With reference to Appendix A.2.2, ${}^{B}\mathbf{\Theta}_{e}$ can be constructed using rotation vector ${}^{B}\mathbf{v}_{e}$, which lies perpendicular to ${}^{B}\mathcal{Z}$ and ${}^{B}\mathcal{Z}^{*}$, and rotation angle θ_{e}, which describes the radial distance between the two vectors,

$$ {}^{B}\mathbf{\Theta}_{e} = \theta_{e}{}^{B}\mathbf{v}_{e}. \tag{5.62} $$

Note that there exist two applicable rotation angles within $[-\pi, \pi)$ that will rotate ${}^{B}\mathcal{Z}$ onto ${}^{B}\mathcal{Z}^{*}$. However, the smallest angle change is preferable for time- and energy-efficient flight manoeuvres. The smallest rotation angle, θ_{e}, can be found by evaluating the dot product of the two vectors, namely

$$ {}^{B}\mathcal{Z} \cdot {}^{B}\mathcal{Z}^{*} = \left|{}^{B}\mathcal{Z}\right| \left|{}^{B}\mathcal{Z}^{*}\right| \cos(\theta_{e}) = \cos(\theta_{e}), \tag{5.63} $$

which follows as

$$ \theta_{e} = \cos^{-1}\left({}^{B}\mathcal{Z} \cdot {}^{B}\mathcal{Z}^{*}\right), \tag{5.64} $$

The rotation vector, ${}^{B}\mathbf{v}_{e}$, is found by employing the cross product,

$$ {}^{B}\mathcal{Z} \times {}^{B}\mathcal{Z}^{*} = \left|{}^{B}\mathcal{Z}\right| \left|{}^{B}\mathcal{Z}^{*}\right| \sin(\theta_{e}){}^{B}\mathbf{v}_{e} = \sin(\theta_{e}){}^{B}\mathbf{v}_{e}. \tag{5.65} $$

in which the rotation vector follows as

$$ {}^{B}\mathbf{v}_{e} = \frac{{}^{B}\mathcal{Z} \times^{B} \mathcal{Z}^{*}}{\sin(\theta_{e})}. \tag{5.66} $$

Combining (5.64) and (5.66), the yaw-free angular error can be described as

$$ {}^{B}\mathbf{\Theta}_{e} = \frac{\cos^{-1}\left({}^{B}\mathcal{Z} \cdot {}^{B}\mathcal{Z}^{*}\right)}{\sqrt{1 - \left({}^{B}\mathcal{Z} \cdot {}^{B}\mathcal{Z}^{*}\right)^{2}}} {}^{B}\mathcal{Z} \times {}^{B}\mathcal{Z}^{*}, \tag{5.67} $$

where ${}^B\mathcal{Z}^*$ is populated from (5.60). Note that the angular error will always be perpendicular to ${}^B\mathcal{Z} = [0\ 0\ 1]^T$, which implies that the yaw channel will never be used to correct angular errors. The yaw-free body-frame angular error in (5.67) is used as the tracking error for the attitude controller, $\mathbf{G}_{\boldsymbol{\eta}}(\mathbf{s})$, as shown in Figure 5.1. An apparent singularity exists in (5.67) when ${}^B\mathcal{Z} \cdot {}^B\mathcal{Z}^* = 1$, which corresponds to the two vectors lying on top of each other (which is the intended configuration). However, the numerator in (5.67) additionally tends to zero under this condition. In practice, this division by zero operation can be protected in code by first checking whether $\cos^{-1}\left({}^B\mathcal{Z} \cdot {}^B\mathcal{Z}^*\right)$ is approaching zero before applying the division operation. Because the third component of (5.67) is always equal to zero, one can extract the first two components using

$$
{}^B\boldsymbol{\Theta}_{e_{xy}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} {}^B\boldsymbol{\Theta}_e, \tag{5.68}
$$

which is used to drive the roll and pitch feedback controllers.

The yaw-free thrust vectoring Simulink implementation is shown in Figure 5.26. With reference to (5.60) and Appendix A.2.2, the desired body-frame z-axis, ${}^B\mathcal{Z}^*$, inherently contains feedback from the estimated quaternion, ${}^W\mathbf{q}_B$, which is used to rotate ${}^B\mathcal{Z}$ onto ${}^B\mathcal{Z}^*$.

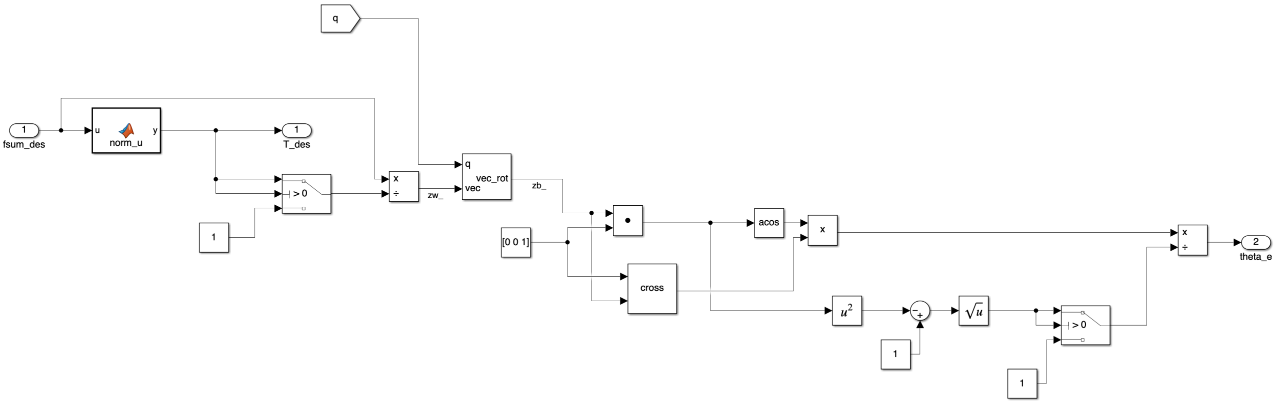


Figure 5.26: Simulink implementation of yaw-free thrust vectoring approach.

5.7 Determining the translation plants

The inertial-frame force equation from (3.47) is repeated below (neglecting the unmodelled disturbance component)

$$
\begin{aligned}
m{}^W\ddot{\mathbf{p}} &= {}^W\mathbf{R}_B{}^B\mathbf{f}_\Sigma + m{}^W\mathbf{g}, \\
&= {}^W\mathbf{f}_\Sigma + m{}^W\mathbf{g}
\end{aligned} \tag{5.69}
$$

which describes the relationship between the generated body-frame thrust, ${}^{B}\mathbf{f}_{\Sigma}$, (encoded) quadcopter orientation, ${}^{W}\mathbf{R}_{B}$, and the resulting rectilinear acceleration ${}^{W}\ddot{\mathbf{p}}$ in frame $\{W\}$. The gravitational force, $m{}^{W}\mathbf{g}$, can be predominantly accounted for using feedforward, with any residual components removed using integrators in the forward transfer function. Following this, under the assumption that the gravitational force has been accounted for by integral action, (5.69) can be simplified to

$$\begin{aligned} m{}^{W}\ddot{\mathbf{p}} &= {}^{W}\mathbf{R}_{B}{}^{B}\mathbf{f}_{\Sigma} \\ &= {}^{W}\mathbf{f}_{\Sigma}. \end{aligned} \tag{5.70}$$

The inertial-frame thrust vector, ${}^{W}\mathbf{f}_{\Sigma} = {}^{W}\mathbf{R}_{B}{}^{B}\mathbf{f}_{\Sigma}$, therefore serves as the control action of the inertial-frame position system. Referring to (3.45), the magnitude of this thrust vector, $||{}^{W}\mathbf{f}_{\Sigma}|| = ||{}^{B}\mathbf{f}_{\Sigma}||$, is dictated by the collective thrust, namely

$$||{}^{W}\mathbf{f}_{\Sigma}|| = \sum_{k=1}^{4} f_k. \tag{5.71}$$

The direction of ${}^{W}\mathbf{f}_{\Sigma} = {}^{W}\mathbf{R}_{B}$ is achieved using the quadcopter orientation. By extension, the bandwidth related to reorienting ${}^{W}\mathbf{f}_{\Sigma}$ will be equivalent to the bandwidth of the closed-loop attitude subsystem. The attitude controller relies on the rotor thrust controllers as the innermost feedback loop mechanism. Referring to Section 5.5.2, the rotor thrusts serve as coupled inputs to the roll and pitch channels. Notably, two integrators relate the relevant rotor thrusts to the corresponding incremental body-frame roll and pitch angular changes. As such, the rotor thrust bandwidth is expected to be significantly higher than that of the attitude plant. As the sum of the rotor thrusts describe $||{}^{W}\mathbf{f}_{\Sigma}||$ based on (5.71), a suitable assumption is to model the dynamics of ${}^{W}\mathbf{f}_{\Sigma}$ using the dominant attitude plant behaviour. This effectively makes use of timescale separation to treat $\sum_{k=1}^{4} f_k$ as being in quasi steady-state with respect to the attitude of the quadcopter. Following this, if the closed-loop roll/pitch attitude system plant instance i is denoted as $T^{i}_{\eta_{xy}}(s)$ (where the symmetrical roll and pitch channels are designed to have identical transfer behaviour), (5.70) can be modelled in transfer function form as

$$\begin{aligned} {}^{w}\mathbf{p}(s) &= \frac{1}{ms^2}\mathbf{T}^{abc}_{\eta_{xy}}(s)\mathbf{u}_p, \\ &= \mathbf{P}^{abc}_{p}(s)\mathbf{u}_p. \end{aligned} \tag{5.72}$$

where

$$\mathbf{T}^{abc}_{\eta_{xy}}(s) = \begin{bmatrix} T^{a}_{\eta_{xy}}(s) & 0 & 0 \\ 0 & T^{b}_{\eta_{xy}}(s) & 0 \\ 0 & 0 & T^{c}_{\eta_{xy}}(s) \end{bmatrix}. \tag{5.73}$$

Note that ${}^{W}\mathbf{f}^{*}_{\Sigma} = \mathbf{u}_p$ is the 3 × 1 control action, measured in Newtons, which describes the desired inertial-frame thrust vector, whereas $\mathbf{T}^{abc}_{\eta_{xy}}(s)\mathbf{u}_p \approx {}^{W}\mathbf{f}^{*}_{\Sigma}$ approximates the actual inertial-frame thrust vector (effectively filtering the desired thrust vector based on the bandwidth of the thrust vectoring system). Note that feedforward information of the desired inertial-frame translational acceleration and gravitational weight can also be appended to ${}^{W}\mathbf{f}^{*}_{\Sigma}$, as shown in (5.55), but is left out of the modelling stage.

5.8 Proposed selection of the attitude and position controllers

5.8.1 Selection of $\mathbf{G}_{\eta}(\mathbf{s})$

With reference to Section 5.6, the attitude controllers make use of a body-frame angular error, ${}^{B}\boldsymbol{\Theta}_{e}$, that does not require control action in the yaw channel. Attitude adjustments are achieved by allocating thrust across the four rotors, which is facilitated by the rotor thrust control scheme in Section 5.2.3. Notably, the rotor thrust scheme is only able to regulate the rotor thrust generated by the quadcopter rotors. Additional thrust from external sources, such as aerodynamic drag and thrust wake interactions, will not be detected by the rotor thrust controllers.

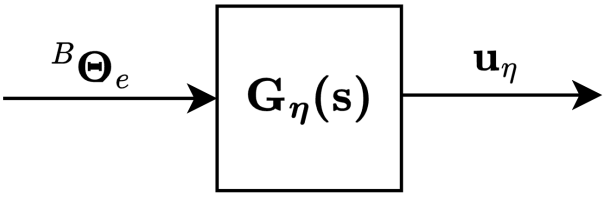


Figure 5.27: Attitude feedback controller subsection from Figure 5.1, which acts on the body-frame angular error, ${}^{B}\boldsymbol{\Theta}_{e}$, and generates the control action, $\mathbf{u}_{\eta}$.

Roll and pitch controllers

Referring to Section 5.5.2, a double integrator is present in the roll and pitch channels when considering angular displacement as the output. This "natural" integration is beneficial in terms of accounting for any low-frequency thrust disturbances that the rotor thrust controllers cannot discern. The roll and pitch channels are treated as identical, as they share common mass moment of inertias, thrust bandwidth, and control allocation. By extension, the feedback controllers will likely take the same form.

The yaw-free body-frame angular error, ${}^{B}\boldsymbol{\Theta}_{e}$, is populated using the inertial-frame quaternion, ${}^{W}\mathbf{q}_{B}$, which is drawn from the explicit complementary filter (detailed in Section 4.3). Refined body-frame angular rate information is also available from the explicit complementary filter, which can be used for high-bandwidth feedback.

Design of the roll and pitch elements of $\mathbf{G}_{\eta}(\mathbf{s})$ can follow classical SISO QFT design, using the plant templates developed in (5.42). Performance specifications can be prescribed based on how quickly the quadcopter is intended to be thrust vectored, which suggests that a model-error tracking QFT formulation is most appropriate for the roll and pitch control channels. However, the rate-limited servomotors must also be factored into the design process.

Yaw-rate controller

With reference to Section 5.4, the design constraint of $\Omega_z \approx 0$ necessitates a body-frame yaw-rate controller, with the objective of regulating the yaw rate at zero. Unlike the roll and pitch controllers, the yaw-rate controller will then accept the negative body-frame yaw rate as the regulation error, with the measurement obtained from the explicit complementary filter (detailed in Section 4.3). Notably, the body-frame angular error is not used as part of the yaw-rate controller, as explained in Section 5.6.1.

The task of regulating a null yaw-rate means that the yaw channel feedback controller should contain at least one integrator. This additional integrator will also assist with effective disturbances from the imperfect thrust to torque mapping in (5.29), as well as other unbalanced torques acting about ${}^{B}\mathcal{Z}$.

The yaw-rate plant set in (5.51) can be used to facilitate the feedback control design of the yaw-rate regulator. As the reference yaw-rate is intended to stay constant at zero, a sensitivity design will be required to effectively regulate the yaw-rate loop.

5.8.2 Selection of $\mathbf{G_p}(s)$

The position feedback controller, $\mathbf{G_p}(s)$, acts on the difference between the desired inertial-frame position, ${}^{W}\mathbf{p}^{*}(s)$, and the measured inertial-frame position, ${}^{W}\mathbf{p}(s)$. Note that ${}^{W}\mathbf{p}^{*}(s)$ can originate from a trajectory planner, such as one of the methods detailed in Appendix B. A pre-known trajectory will result in the smoothest flight behaviour, as the desired translational acceleration, ${}^{W}\ddot{\mathbf{p}}^{*}$, can be incorporated in the thrust vectoring approach, as shown in (5.55), which takes the form of a feedforward signal that exploits differential flatness. One can alternatively use a QFT-based design methodology to design a frequency-dependant feedforward filter (in place of feeding forward the desired acceleration). In the absence of a known desired translational acceleration or feedforward filter, (5.55) can still be used, but the control action from $\mathbf{G_p}(s)$, $\mathbf{u}_p$, will be required to account for the absence of the acceleration term during transients, thereby requiring larger feedback controller gain in the mid- to high-frequency band.

After successful loop closure of the attitude plant, $\mathbf{G_p}(s)$ can be designed using information from the other closed-loop subsystems. Time-scale separation (quasi steady-state interpretation of the attitude controller with respect to the position controller) can be exploited to simplify the relationship between the closed-loop attitude plant bandwidth and thrust vectoring bandwidth.

5.9 Simulation of position control system

The proposed control scheme captured in Figure 5.1 exploits the fact that the fundamental dynamics are sufficiently well known. Additionally, the decoupling scheme, which is made up of the rotor thrust control, control authority allocation, and thrust vectoring, makes the system so strongly diagonally dominant that single loop controllers are sufficient to complete the control design. In other words, the MIMO design problem is solved using engineering insights to remove channel coupling, circumventing the need for multivariate control design. As such, the MIMO bound generation methods developed in Section 6.5-6.6 is not required in the overall design of the quadcopter.

This Section presents a simulation of the position control system, which is predicated on appropriate design of the attitude control system. The objective is to demonstrate multi-channel operation of the variable-pitch quadcopter, using thrust vectoring, to move from a starting inertial-frame location to another.

5.9.1 Design of attitude controller, $\mathbf{G}_\eta(s)$

Referring to Figure 5.28, which is repeated from Figure 5.1 for sake of convenience, $\mathbf{G}_\eta(s)$ can be designed using information from Section 5.5.2-5.5.3. With reference to Section 5.8, the roll and pitch channels are used to vector the quadcopter thrust, whereas the yaw channel is responsible for maintaining zero rotational rate about ${}^{B}\mathcal{Z}$. The roll-pitch angular error vector, ${}^{B}\mathbf{\Theta}_{e_{xy}}$ was derived in Section 5.6.1 and is used by $\mathbf{G}_\eta(s)$ in order to adjust the thrust vector. The yaw-rate error is equal to the negative body-frame angular rate about ${}^{B}\mathcal{Z}$, $-{}^{B}\Omega_z$, as the reference yaw-rate is zero. Following this, the attitude control action, $\mathbf{u}_\eta$, is expressed as

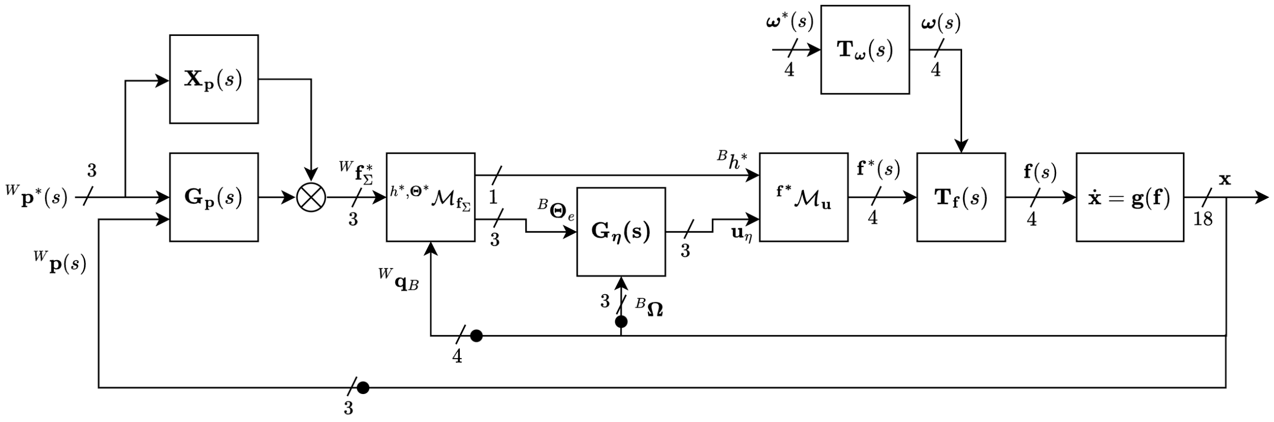


Figure 5.28: Abstracted block diagram of quadcopter control scheme, which contains the various control signal mappings, feedback controllers, and closed-loop subsystems. The signals and systems are detailed in Table 5.1.

$$\mathbf{u}_\eta = \mathbf{G}_\eta(s) \begin{bmatrix} {}^{B}\mathbf{\Theta}_{e_{xy}} \\ -{}^{B}\Omega_z \end{bmatrix}. \tag{5.74}$$

$\mathbf{G}_{\eta}(s)$ is designed as a diagonal matrix, as the combination of the low-level rotor thrust, rotor speed control, and control authority allocation, results in channel decoupling when ${}^{B}\Omega_z \approx 0$.

Yaw-rate control design

Following the insights from Section 5.4, the yaw-rate controller is responsible for regulating the yaw-rate with a null reference (${}^{B}\Omega_z^*(s) = 0$). Disturbances and uncertainty effects acting on the yaw channel necessitate integral control, which can enforce ${}^{B}\Omega_z(s) = 0$ at steady-state. Unlike the roll/pitch channels (that are used to vector the quadcopter thrust), the yaw channel is not encompassed by any subsequent feedback loops, which means that the yaw-rate feedback controller is solely responsible for regulating ${}^{B}\Omega_z(s) = 0$. Based on Section 5.5.3, yaw-rate open-loop plant instance $\{a, b, c, d\}$ was found to be

$$\frac{\Omega_z(s)}{u_\psi} = P_{\Omega_z}^{abcd}(s) = \frac{1}{J_{zz}s}\left[T_f^a(s)\beta_1\gamma_\psi - T_f^b(s)\left(1-\beta_1\right)\gamma_\psi + T_f^c(s)\beta_2\left(1-\gamma_\psi\right) - T_f^d(s)\left(1-\beta_2\right)\left(\gamma_\psi - 1\right)\right]. \tag{5.75}$$

The corresponding plant set is determined by sweeping through every plant quartet, which defines the set $P_{\Omega_z}^{abcd}(s) \in \mathcal{P}_{\Omega_z}(s)$. The yaw-rate feedback controller, $G_{\Omega_z}(s)$, is designed with the objective of meeting the sensitivity specifications

$$\begin{aligned}
&\left|\frac{1}{1+P_{\Omega_z}^{abcd}(j\omega)G_{\Omega_z}(j\omega)}\right| \le 10^{6/20}, \forall a,b,c,d,\ \forall\omega,\\
&\left|\frac{1}{1+P_{\Omega_z}^{abcd}(j\omega)G_{\Omega_z}(j\omega)}\right| \le 10^{-20/20}, \forall a,b,c,d,\ \omega \le 0.1,
\end{aligned} \tag{5.76}$$

using $\omega_d \in \{0.1, 1, 5, 10, 20, 30\}$ rad/s as the design frequencies of interest. The sensitivity design regions are shown in Figure 5.29, as well as the nominal open-loop plant frequency response, with

$$G_{\Omega_z}(s) = 0.062\frac{1+s/3.2}{s(1+s/169)(1+s/400)}. \tag{5.77}$$

The strictly proper structure of $G_{\Omega_z}(s)$ provides high-frequency roll of the gain, which helps to suppress noise from the angular rate measurement.

Roll/pitch control design

The roll and pitch channels, and by extension, roll and pitch controllers, are treated as identical. This is a suitable assumption, as both channels have matching inertia and control authority. Designing the two channels with parity means that both channels can work together to vector the quadcopter thrust, instead of having to unevenly rely on one channel when translating in the inertial frame. Following this, the roll/pitch plant for plant

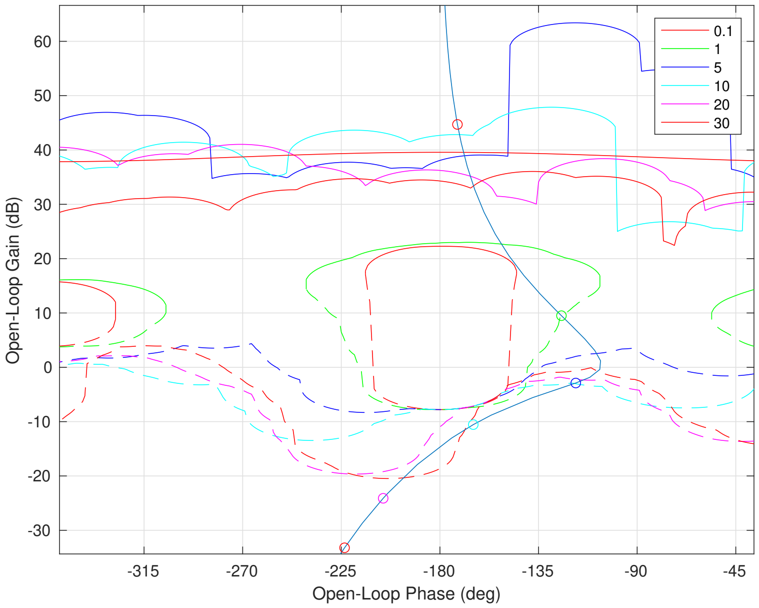


Figure 5.29: Log-polar plane, showing *stay-out* sensitivity design regions from (5.76). The nominal open-loop plant frequency response is also shown.

pair $\{i, k\}$, repeated from (5.43), is

$$P^{ik}_{\Delta\phi}(s) = P^{ik}_{\Delta\theta}(s) = \frac{T^i_f(s)\gamma_\phi - T^k_f(s)\left[\gamma_\phi - 1\right]}{J_{xx}s^2}, \ \forall i, k. \tag{5.78}$$

Sweeping through every pair of $\{T^i_f(s), T^k_f(s)\}$ results in the roll/plant set of $P^{ik}_{\Delta\phi}(s) = P^{ik}_{\Delta\theta}(s) = P^{ik}_{\eta_{xy}}(s) \in \mathcal{P}_{\eta_{xy}}(s)$. Similar to the yaw-rate control loop, matching sensitivity design constraints are prescribed

$$\begin{aligned} \left|\frac{1}{1 + P^{ik}_{\Delta\theta}(j\omega)G_{\eta_{xy}}(j\omega)}\right| &\leq 10^{6/20}, \forall i, k, \ \forall\omega, \\ \left|\frac{1}{1 + P^{ik}_{\Delta\theta}(j\omega)G_{\eta_{xy}}(j\omega)}\right| &\leq 10^{-20/20}, \forall i, k, \ \omega \leq 1, \end{aligned} \tag{5.79}$$

using $\omega_d \in \{0.1, 1, 5, 10, 20, 30, 50\}$ rad/s as the design frequencies of interest. The sensitivity design regions relating to the roll/pitch channel are shown in Figure 5.30, as well as the nominal open-loop plant frequency response, with

$$G_{\eta_{xy}}(s) = 0.019\frac{1 + s/0.62}{1 + s/303}. \tag{5.80}$$

$G_{\eta_{xy}}(s)$ is designed as a PD controller with the knowledge that the outer position loop can account for any orientation errors as part of the thrust vectoring. The high-frequency gain of $G_{\eta_{xy}}(s)$ is equal to 9.29, which

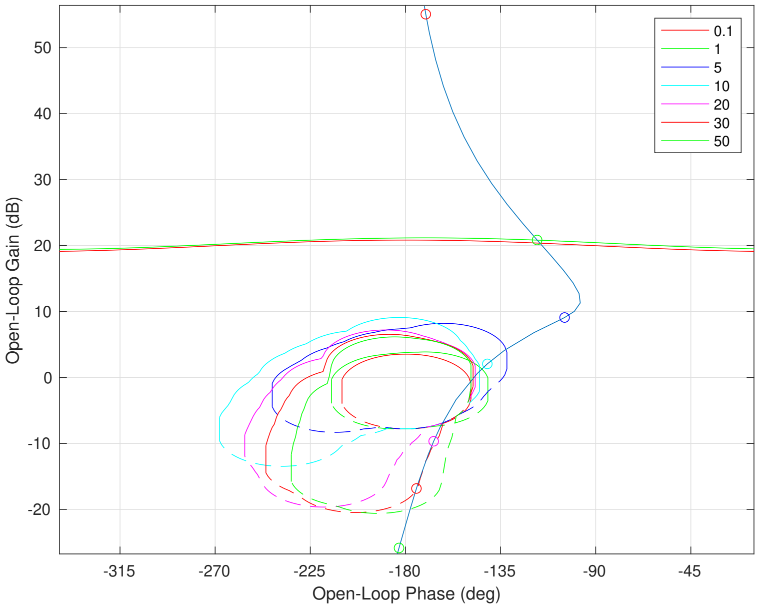


Figure 5.30: Log-polar plane, showing *stay-out* sensitivity design regions from (5.79). The nominal open-loop plant frequency response is also shown.

is significantly higher than that of the DC gain of 0.019. However, the roll/pitch attitude controller is expected to act on body-frame errors that have already been filtered by the position control loop, which also has a lower control bandwidth. Following the design of the yaw-rate and roll/pitch controllers, (5.74) can be written as

$$\begin{aligned}\mathbf{u}_\eta &= \mathbf{G}_\eta(s)\begin{bmatrix}{}^B\boldsymbol{\Theta}_{e_{xy}} \\ -{}^B\Omega_z\end{bmatrix}, \\ &= \begin{bmatrix}G_{\eta_{xy}}(s) & 0 & 0 \\ 0 & G_{\eta_{xy}}(s) & 0 \\ 0 & 0 & G_{\Omega_z}(s)\end{bmatrix}\begin{bmatrix}{}^B\boldsymbol{\Theta}_{e_{xy}} \\ -{}^B\Omega_z\end{bmatrix},\end{aligned} \tag{5.81}$$

as shown in Figure 5.28.

5.9.2 Design of position controller, $\mathbf{G_p}(s)$

Following the design of $\mathbf{G}_\eta(s)$, the translational plant set can be modelled based on the insights in Section 5.7. Specifically, ignoring disturbance effects, the inertial-frame position can be modelled as

$$\begin{aligned}{}^w\mathbf{p}(s) &= \frac{1}{ms^2}\mathbf{T}^{abc}_{\eta_{xy}}(s)\mathbf{u}_p, \\ &= \mathbf{P}^{abc}_p(s)\mathbf{u}_p,\end{aligned} \tag{5.82}$$

where

$$\mathbf{T}^{abc}_{\eta_{xy}}(s) = \begin{bmatrix} T^{a}_{\eta_{xy}}(s) & 0 & 0 \\ 0 & T^{b}_{\eta_{xy}}(s) & 0 \\ 0 & 0 & T^{c}_{\eta_{xy}}(s) \end{bmatrix} \tag{5.83}$$

represents the bandwidth of the quadcopter thrust vectoring. Based on the composition of (5.83), SISO feedback control design can proceed on one of the position channels without loss of generality. For sake of simplicity, sensitivity constraints are first applied to the position control loop of

$$\begin{aligned} \left|\frac{1}{1+P^{abc}_{p}(j\omega)G_{p}(j\omega)}\right| &\leq 10^{6/20}, \forall a,b,c,\ \forall\omega, \\ \left|\frac{1}{1+P^{abc}_{p}(j\omega)G_{p}(j\omega)}\right| &\leq 10^{-20/20}, \forall a,b,c,\ \omega \leq 1, \end{aligned} \tag{5.84}$$

using $\omega_d \in \{0.05, 0.5, 2.5, 5, 10, 15, 25\}$ rad/s as the design frequencies of interest. While the sensitivity bounds in (5.84) are equivalent in magnitude to those of (5.79), the frequencies are chosen to be half of the corresponding attitude controller design frequencies, which sets the position loop bandwidth at approximately half of the attitude loop bandwidth.

The sensitivity design regions resulting from (5.84) are shown in Figure 5.31, as well as the nominal open-loop plant frequency response, with

$$G_p(s) = 0.64\frac{(1+s/0.5)(1+s/0.78)}{s(1+s/72)}. \tag{5.85}$$

$G_p(s)$ represents a PID controller, which benefits from large gain at low frequencies and phase lead at the mid frequencies (which is needed to overcome the phase lag from the two integrators in the forward transfer function). The inertial-frame position vector controller follows as

$$\mathbf{G_p}(s) = \begin{bmatrix} G_p(s) & 0 & 0 \\ 0 & G_p(s) & 0 \\ 0 & 0 & G_p(s) \end{bmatrix}, \tag{5.86}$$

as shown in Figure 5.28. Note that while the inertial-frame lateral and longitudinal behaviour of the quadcopter will, in general, perform differently depending on the direction of the thrust vector, the dominant double integrator term of $\frac{1}{ms^2}$, relating thrust and position, is common across the position channels. The step response of the resulting closed-loop plant set is shown in Figure 5.32. The overshoot of approximately 40% requires reduction, which can be achieved using a feedforward filter structure as shown in Figure 5.28. Following the design approach in Section 5.2.3, a feedforward filter, $X_p(s)$ is selected such that the closed-loop position system

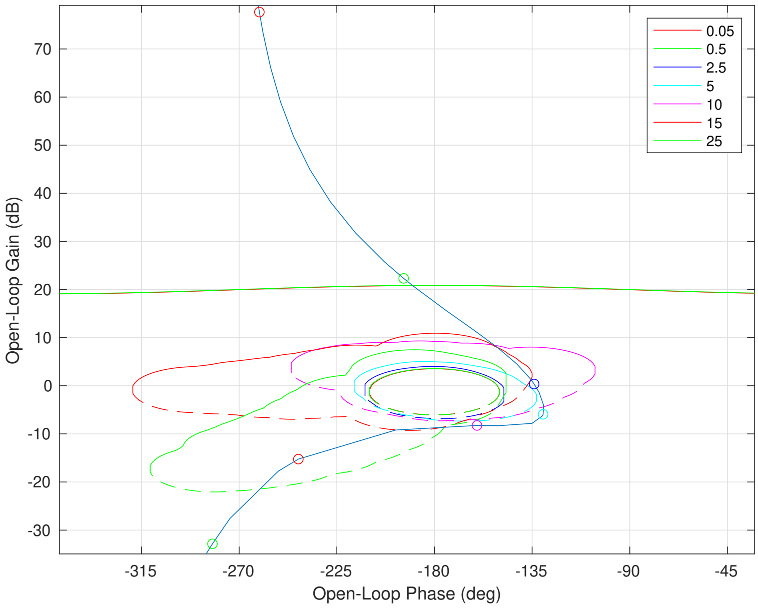


Figure 5.31: Log-polar plane, showing *stay-out* sensitivity design regions from (5.79). The nominal open-loop plant frequency response is also shown.

approaches the designated reference model

$$M_p(s) = \frac{3.33^2}{s^2 + 2(0.6)3.33s + 3.33^2}, \tag{5.87}$$

for which the unit step response is also shown in Figure 5.32. Using a model-error tolerance set of $\boldsymbol{\beta} = \{0.01, 0.1, 0.31, 0.3, 0.32, 0.15, 0.08\}$, the model-error design boundaries arise as shown in Figure 5.33. A valid feedforward filter is

$$X_p(s) = -0.34\frac{(1 + s/0.5)(1 + s/11.93)}{(1 + s/21.7)(1 + s/25.57)}, \tag{5.88}$$

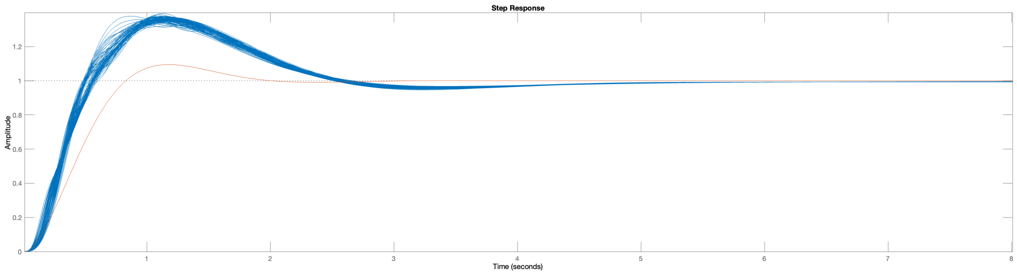


Figure 5.32: Unit step response of closed-loop position control plant set (blue). The designated reference model step response is also shown (red)

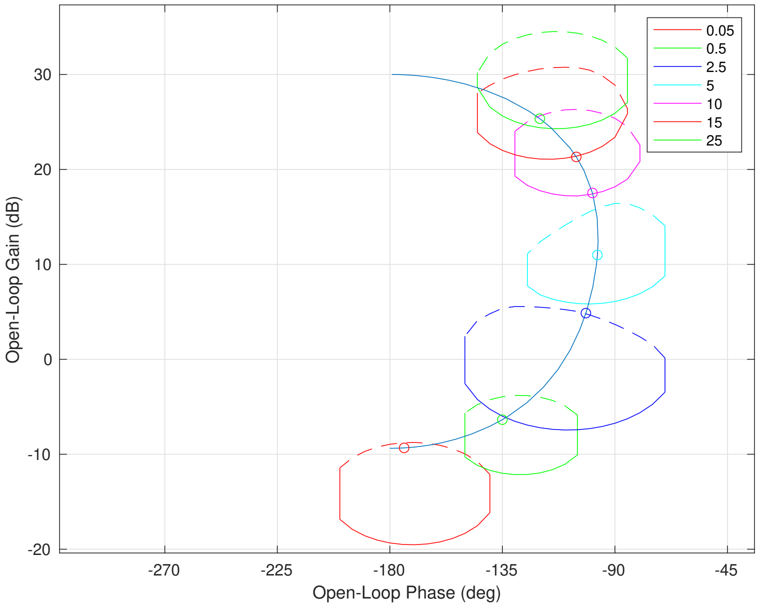


Figure 5.33: Log-polar plane, showing *stay-in* design regions of $X_p(j\omega)$. The frequency response of the chosen feedforward filter is also shown.

which is also shown in Figure 5.33. The step response of the closed-loop position system after inclusion of $X_p(s)$ is shown in Figure 5.34, which exhibits less overshoot without sacrificing settling time. The matrix feedforward filter follows as

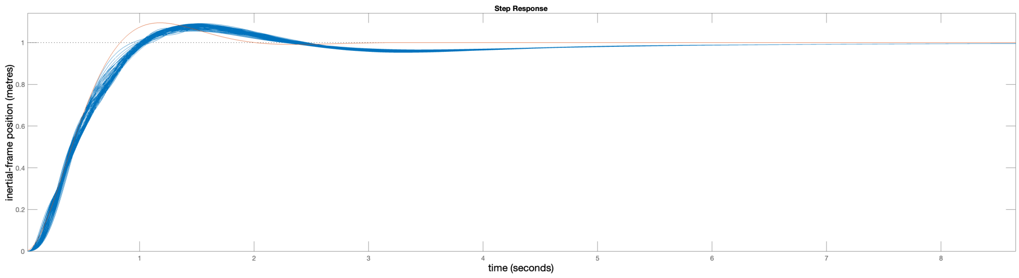


Figure 5.34: Unit step response of closed-loop position control plant set (blue). The designated reference model step response is also shown (red)

$$\mathbf{X_p}(s) = \begin{bmatrix} X_p(s) & 0 & 0 \\ 0 & X_p(s) & 0 \\ 0 & 0 & X_p(s) \end{bmatrix}, \tag{5.89}$$

which acts on all position channels as shown in Figure 5.28.

5.9.3 Simulation

The full control system is implemented in Simulink using the simulation environment detailed in Section 2.5, as shown in Figure 5.35. Note that the structure in Figure 5.35 mirrors that of Figure 5.28.

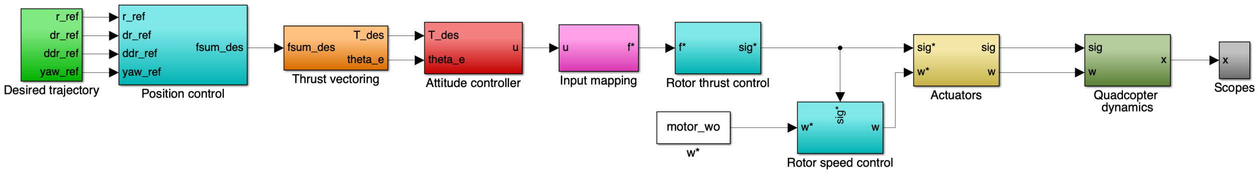


Figure 5.35: Top-level representation of simulation environment using Simulink. The simulation model captures the six-degree-of freedom motion of the quadcopter system, and contains real-world effects, such as measurement noise, digitisation, nonlinearities, and time delay effects. Feedback paths are not explicitly shown in the model but are facilitated using "Goto" and "From" simulink blocks within the various subsystems.

The rotor thrust and torque behaviour is modelled based on the developed and verified equations in Section 3.3, and captures the nonlinear blade angle to thrust/torque mappings. The rotor speed and thrust dynamics are simulated based on the experimental results in Section 3.7-3.8. The blade angle has a saturation limit of ±2500 (in 16-bit count), which corresponds to an angle of 18°. Backlash, quantisation, and rate limiting are also captured as part of the servomotor-rotor model, which makes use of the identified parameters from Chapter 3.

Band-limited white noise is added to the relevant state measurements to emulate that of the physical system, as detailed in Table 5.9, and all controllers and filters are implemented in digital discrete form with a sampling rate of 200 Hz.

N_f	N_Ω	N_q
8×10^{-6}	4×10^{-4}	1×10^{-8}

Table 5.9: Noise power of band-limited white noise that is applied to the relevant state measurements. N_f, N_Ω, and N_q refer to the noise power of the rotor thrust, body-frame angular rate, and quaternion, respectively.

The quadcopter is expected to take off from the ground at $t = 0$ seconds and rise to a height of 1 metre off the ground in the ${}^{W}\mathcal{Z}$ axis. At 10 seconds, the quadcopter is commanded to move to the inertial-frame position ${}^{W}\mathbf{p}^{*} = [1 \ -1 \ 1]^{T}$, which requires simultaneous pitching and rolling of the quadcopter body. With reference to Figure 5.28 and (5.55), the approximate gravitational weight is also fed forward to the desired inertial-frame thrust vector, in order to account for the gravitational disturbance (at steady-state). The inertial-frame position reference and response of the simulated variable-pitch quadcopter is shown in Figure 5.36. With reference to Figure 5.36, after settling at the initial height of 1 metre, the altitude control remains constant while the quadcopter translates, which suggests that the thrust vectoring approach is correctly allocating control action between the heave command and attitude controller. Snapshots of the six-degree-of-freedom motion of

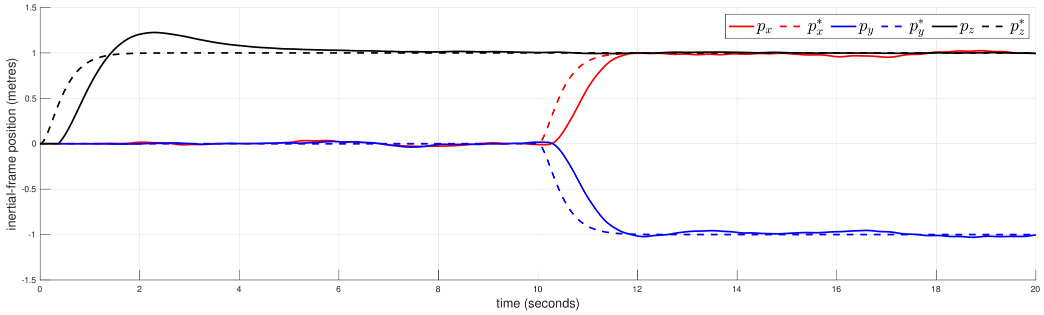
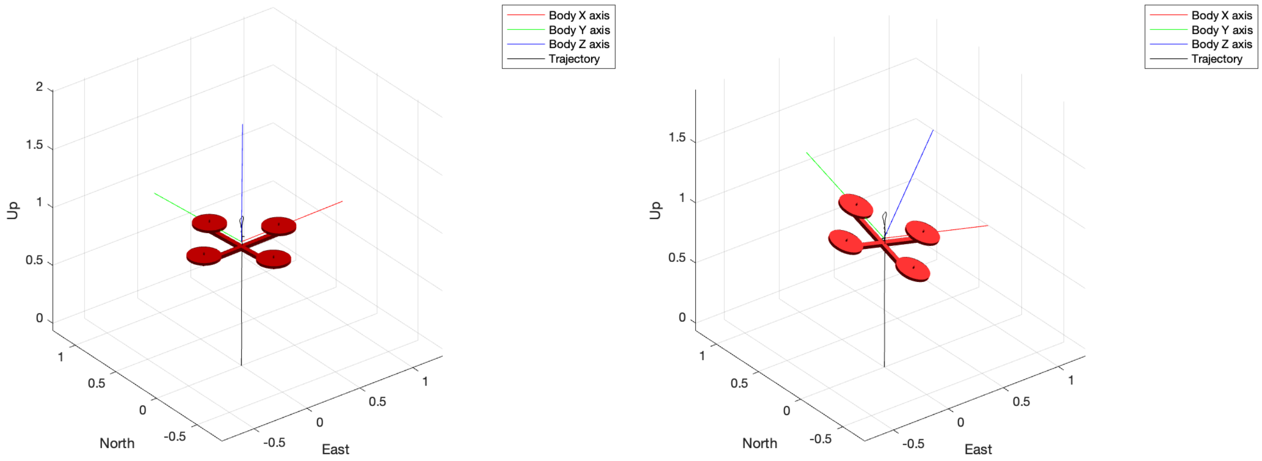


Figure 5.36: Time-domain plot, showing system response of inertial-frame position vector. Dotted lines represent reference positions, whereas solid lines represent the actual positions.

the quadcopter is shown in Figure 5.37. The intrinsic *ZYX* Euler angles are used to represent the quadcopter orientation in Figure 5.38 (for ease of reading). Notably, the yaw angle, ψ, is not constant in Figure 5.38, and drifts over time (based on only regulating the body-frame yaw rate). However, the yaw-free thrust vectoring approach detailed in Section 5.6 is able to allocate control action to the roll and pitch channels appropriately, regardless of the yaw angle. Importantly, the yaw angle is drifting slowly, which suggests that the yaw-rate constaint of ${}^{B}\Omega_z \approx 0$ is being adhered to (thereby avoiding roll-pitch channel coupling). The rotor thrust control serves as the inner loop of the attitude control system, and its response is shown in Figure 5.39. The noise power is noticeably high in Figure 5.39 (based on matching the simulation model with that of the thrust estimator in Section 4.4), but the rotor thrust controllers are still able to track the corresponding desired thrust vectors. Finally, the rotor blade angle behaviour is shown in Figure 5.40. The blade pitch actuation is safely within the limits of ±18° and do not exhibit any limit cycling.

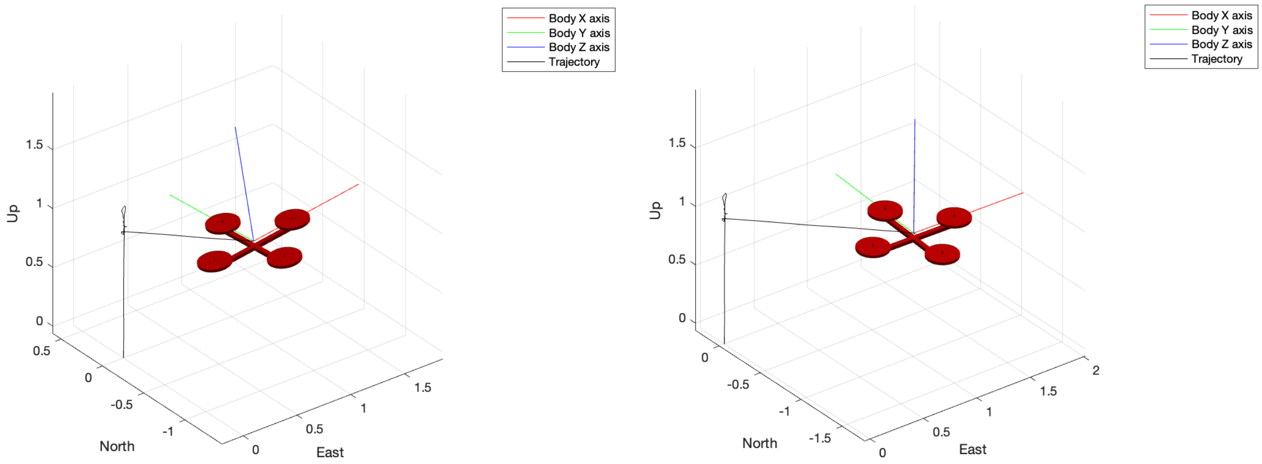
During this project, multiple flight tests were conducted. This included attitude-only control, and attitude-plus-altitude-hold control. However, the flight tests were preliminary experiments and not founded on robust control design as captured in this Chapter. As such, these flights are not documented here. The quadcopter was unfortunately subject to a number of mechanical malfunctions, which led to crashes and often long periods of repair/replacement of parts. For this reason, final flight tests were never implemented on the variable-pitch quadcopter.

The closed-loop simulation of the variable-pitch quadcopter is available for viewing in the simulationEnvironment GitHub repository.



(a) $t = 7$ seconds.

(b) $t = 10.2$ seconds.



(c) $t = 11$ seconds.

(d) $t = 12.4$ seconds.

Figure 5.37: Snapshots of the six-degree-of-freedom variable quadcopter during the simulated manoeuvre.

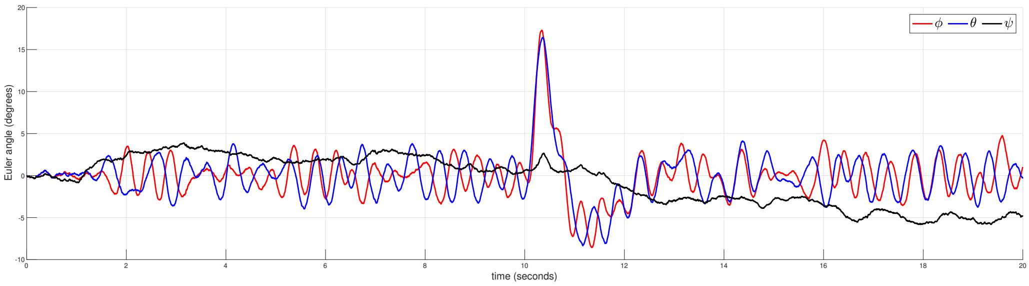


Figure 5.38: Time-domain plot, showing inertial-frame Euler angles that describe quadcopter orientation.

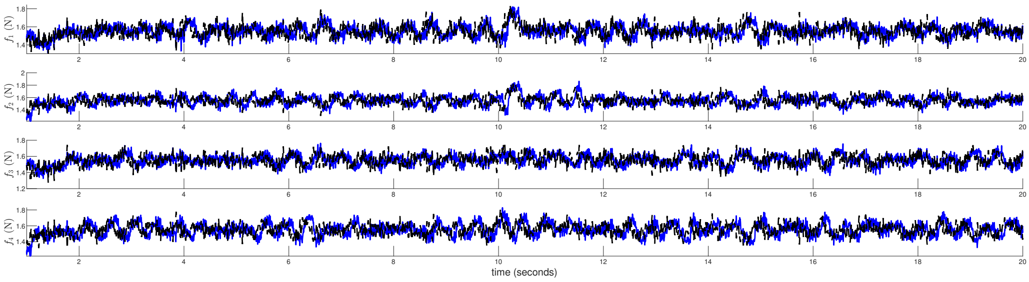


Figure 5.39: Time-domain plot, showing response of desired rotor thrusts (black) and measured rotor thrusts (blue).

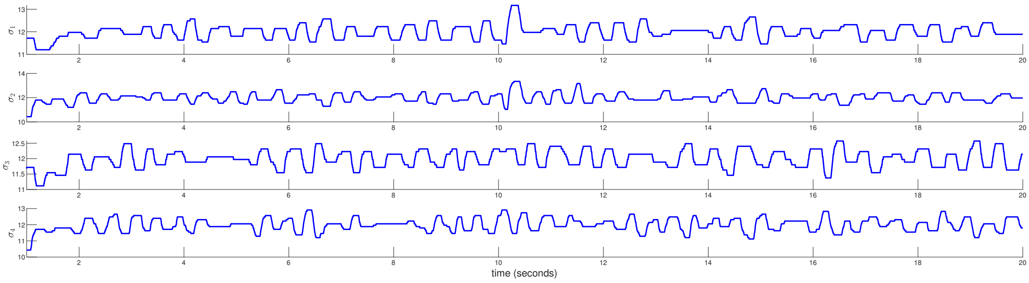


Figure 5.40: Time-domain plot, showing quadcopter blade angle behaviour (measured in degrees) during simulated flight.

Chapter 6

Contributions to QFT

This chapter is based, in part, on the following first-authored and published papers:

- Robust plant by plant control design using model-error tracking sets [73]
- A complementary QFT solution to the 2×2 tracking error problem [20]
- A refinement approach to the multivariable tracking error problem [74]

Section 6.1 provides a summary of the formative work related to quantitative feedback theory contributions in this thesis. A short historical note on the development of related multivariable frequency-domain methods are provided in Section 6.2. Section 6.3 details the minor notational changes used in the verbatim journal papers. Section 6.4 presents a multi-model approach to the model-error tracking problem, where each plant instance has a corresponding reference model that is synthesized using optimisation. Section 6.5 develops a complementary QFT design solution that is applicable to 2×2 plant sets. This approach makes use of a non plant-inverting constraint set that is able to reduce feedback control design conservatism at the gain-phase crossover region. Finally, Section 6.6 provides a refinement approach to multivariable tracking designs, whereby iterative refinement is able to reduce design conservatism, and by extension, the feedback controller gain required to meet the prescribed performance specifications.

6.1 Developments in quantitative feedback theory

This Section contains a summarised account of the dominant preceding contributions in quantitative feedback theory to date.

6.1.1 The two-degree-of-freedom control paradigm

Quantitative feedback theory (QFT) was developed by Isaac Horowitz who was one of the first to recognise the importance of feedback in dealing with uncertainty in a systematic way. Although feedback has been used

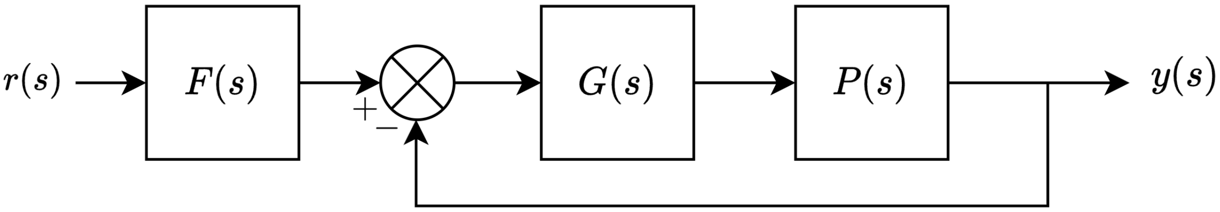


Figure 6.1: Block diagram of traditional two-degree-of-freedom control scheme, where prefilter $F(s)$ acts on the reference signal, and feedback controller $G(s)$ acts on the filtered reference tracking error.

throughout history to deal with uncertainty, it was not approached as a deliberate, quantitative design [75]. The underlying design methodology is used to convert time-domain specifications into approximate frequency-domain counterparts, where uncertainty is easily quantified and accounted for in the robust design process [76].

For tracking problems, QFT commonly makes use of a two-degree-of-freedom (2DOF) control structure, as exemplified in Figure 6.1, in the case that both the plant output, $y(t)$, and reference signal, $r(t)$, are measurable [77]. $G(s)$ and $F(s)$ are the feedback controller and pre-filter, respectively, and $P^i(s)$ indicates plant instance i within the uncertain plant set: $P^i(s) \in \mathcal{P}$.

In the case of reference tracking designs, the control scheme, comprising $F(s)$ and $G(s)$, is designed in order to force the closed-loop output signal, $y(t)$, to follow the reference tracking signal, $r(t)$, within prescribed performance tolerances. While these specifications often originate in the time-domain, the QFT methodology relies on frequency-domain specifications, which requires approximate performance mappings from the time domain to the frequency domain. As a result, iterative controller refinement stages are usually required to achieve the desired closed-loop performance. A one-degree-of-freedom (1DOF) design ($F(s) = 1$) is necessitated when the only measurable signal in the closed-loop system is the tracking error, $e(t) = r(t) - y(t)$, or if the reference signal remains constant, $\dot{r}(t) = 0$. For measurable, arbitrary reference signals, pre-filter could help to improve the steady-state tracking if the complementary sensitivity is not identity. Control design practitioners may also elect to forgo the second degree of freedom in order to simplify the design problem.

Single-input-single-output systems

The overarching benefit of using a 2DOF control scheme can be seen by examining the interaction between the the closed-loop sensitivity function

$$S(s) = \frac{1}{1 + P(s)G(s)}, \tag{6.1}$$

and the corresponding closed-loop transfer behaviour from reference to output

$$T(s) = \frac{P(s)G(s)F(s)}{1 + P(s)G(s)}. \tag{6.2}$$

In the case of a 1DOF control system ($F(s) = 1$), the relationship between $S(s)$ and $T(s)$ is simply

$$S(s) = 1 - T(s) = \frac{1}{1 + P(s)G(s)}, \tag{6.3}$$

which infers that there is complementary interaction between the systems reference tracking ability and sensitivity. However, the 2DOF equivalent structure has the relationship

$$S(s) = 1 - \frac{T(s)}{F(s)}, \tag{6.4}$$

where $F(s)$ is dictated by the control designer and is used to decouple the tracking and sensitivity designs. Sensible design aims to minimise the noise amplification to the plant input,

$$U(s) = \frac{-G(s)}{1 + P(s)G(s)}, \tag{6.5}$$

which implies that $G(s)$ should be selected with the lowest possible gain in the frequency band where measurement noise is most prevalent. As such, $G(s)$ is commonly designed to simultaneously satisfy the disturbance rejection and closed-loop reference to output uncertainty reduction requirements with the lowest possible gain (subject to controller complexity). Design of $F(s)$ regularly ensues after the disturbance rejection and uncertainty requirements have been met, with the sole purpose of shaping the reference signal prior to its injection into the feedback loop, thereby adjusting the reference to output bandwidth as required, and suppressing unwanted resonant peaks in the complementary sensitivity magnitude response. In all SISO feedback designs, the closed loop transfer functions are linear fractional mappings of the controller transfer function. This means that SISO designs are translated into satisfying quadratic inequalities.

Multi-input-multi-output systems

The multi-input-multi-output (MIMO) control paradigm shares many analogues with that of the original SISO 2DOF formulation. In addition, the MIMO attribute of channel coupling and interaction poses additional challenges when selecting the controller elements to can satisfy prescribed performance specifications. Broadly speaking, the feedback controller matrix, $\mathbf{G}(s)$, often designated as diagonal to simplify the design, acts to reduce plant uncertainty and disturbance effects, and the pre-filter matrix, $\mathbf{F}(s)$, attempts to adjust reference to output vector bandwidth. $\mathbf{G}(s)$ is also inherently responsible for reducing channel coupling, which is traded off with increased feedback control gain. However, incorporating a fully populated pre-filter matrix has the substantial benefit of assisting with the reference to output channel coupling and uncertainty reduction. This is arguably the main benefit of incorporating a systematic 2DOF MIMO control scheme, as nondiagonal matrix $\mathbf{F}(s)$ can be designed in such a way as to ease the decoupling burden on $\mathbf{G}(s)$, thereby reducing sensor noise amplification at the plant input. This type of design commonly makes use of existence conditions, whereby the resulting design regions on $\mathbf{G}(s)$ (and subsequently $\mathbf{F}(s)$) inherently aim to reduce the closed-loop reference to output uncertainty and channel coupling using both $\mathbf{G}(s)$ and $\mathbf{F}(s)$. This important point is best illustrated in the subsequent MIMO sections.

Unlike the SISO counterpart, multivariate QFT relies on application of the triangle inequality to account for unknown gain and phase behaviour prior to completion of the control design. This methodology inherently results in design conservatism that depends on the severity of the over-bounding induced when assuming the worst-case gain and phase bounds.

6.1.2 Above/below tracking bounds

SISO design formulation

The method of magnitude-based tracking bounds is common among QFT practitioners [11, 78, 79] and was first applied to SISO systems. In this SISO design configuration, one can specify per-frequency above and below magnitude bounds on the frequency response of each closed-loop plant instance, $T^i(s),\ \forall i$. Application of the magnitude bounds to all closed-loop plant instances results in a set of design constraints of the form,

$$B(\omega) \le \left|T^i(j\omega)\right| \le A(\omega),\ \forall i, \tag{6.6}$$

where $B(\omega)$ and $A(\omega)$ are the *stay-above* and *stay-below* magnitude bounds, respectively, as shown in Figure 6.2a. $B(\omega)$ therefore serves as the low-bandwidth specification, with $A(\omega)$ representing the high-bandwidth constraint. Specifying the magnitude of a SISO, minimum phase transfer function over all frequencies specifies the phase via a Bode integral. Non-minimum phase zeros in the open loop system appear in the closed-loop system. Given that the closed-loop transfer behaviour for plant i from reference to output is

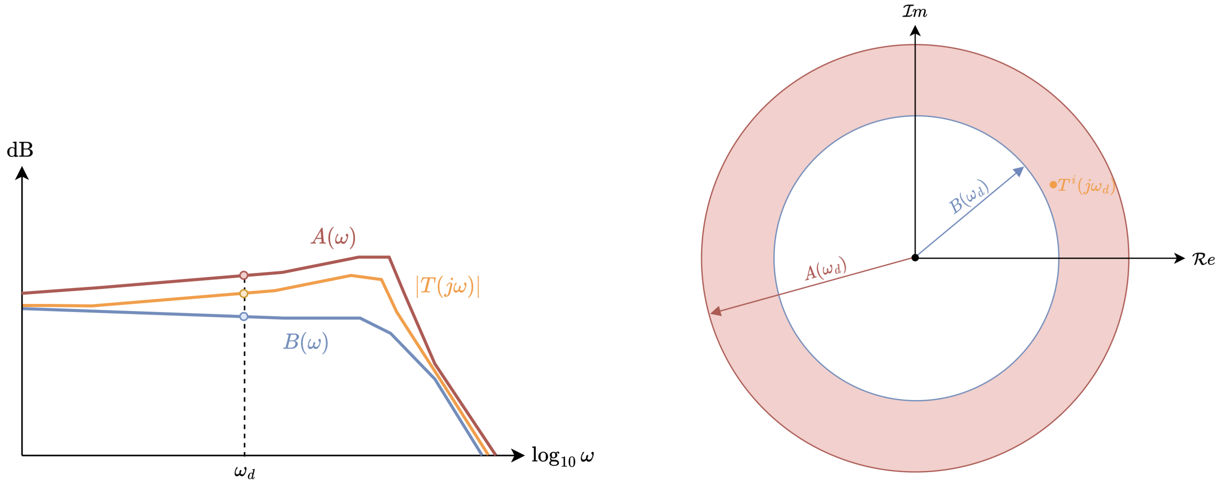
$$T^i(s) = \frac{P^i(s)G(s)F(s)}{1+P^i(s)G(s)} = \frac{L^i(s)F(s)}{1+L^i(s)}, \tag{6.7}$$

where $L^i(s) = P^i(s)G(s)$, the design inequality in (6.6) can be represented explicitly as

$$B(\omega) \le \left|\frac{L^i(j\omega)F(j\omega)}{1+L^i(j\omega)}\right| \le A(\omega). \tag{6.8}$$

The design constraint of (6.6) can be represented visually on a Bode magnitude plot, as shown in Figure 6.2a, or on a per-frequency basis in the arithmetic-complex plane, as shown in Figure 6.2b.

Satisfying the constraint in (6.8) is dependant on the gain and phase response of $G(s)$, as well as the magnitude response of $F(s)$. The 2DOF design is decoupled by defining necessary conditions for the existence of an $F(j\omega)$ that can satisfy (6.8). Dividing through by the inverse of the complementary-sensitivity magnitude response



(a) (b)

Figure 6.2: Visual representations of magnitude-based tracking bounds, displayed on (a) a Bode magnitude plot and (b) arithmetic-complex plane with the design frequency point ω_d.

yields

$$B(\omega)\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right| \leq |F(j\omega)| \leq A(\omega)\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|, \tag{6.9}$$

which describes the solution space of $F(j\omega)$ for plant instance i. Geometrically, the solution space of $F(j\omega)$ from (6.9) is an annulus in the arithmetic complex plane, with an outer radius of $A(\omega)\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|$, and an inner radius of $B(\omega)\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|$, as seen in Figure 6.3. By extension, the solution space of $F(j\omega)$ across all plant cases follows as

$$B(\omega)\max\{\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|, \forall i\} \leq |F(j\omega)| \leq A(\omega)\min\{\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|, \forall i\}, \tag{6.10}$$

which in simple terms requires that the largest inner radius (across all plant instances) be smaller than that of the smallest outer radius (also across all plant instances). This requirement is expressed mathematically as

$$B(\omega)\max\{\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|, \forall i\} \leq A(\omega)\min\{\left|\frac{1+L^i(j\omega)}{L^i(j\omega)}\right|, \forall i\}, \tag{6.11}$$

or

$$B(\omega)\max\{\left|\frac{1}{P^i(j\omega)}+G(j\omega)\right|, \forall i\} \leq A(\omega)\min\{\left|\frac{1}{P^i(j\omega)}+G(j\omega)\right|, \forall i\}. \tag{6.12}$$

An equivalent form that can be converted into a mathematical bound generation algorithm is

$$B(\omega)\left|\frac{1}{P^i(j\omega)}+G(j\omega)\right| \leq A(\omega)\left|\frac{1}{P^k(j\omega)}+G(j\omega)\right|, \forall i,k, \tag{6.13}$$

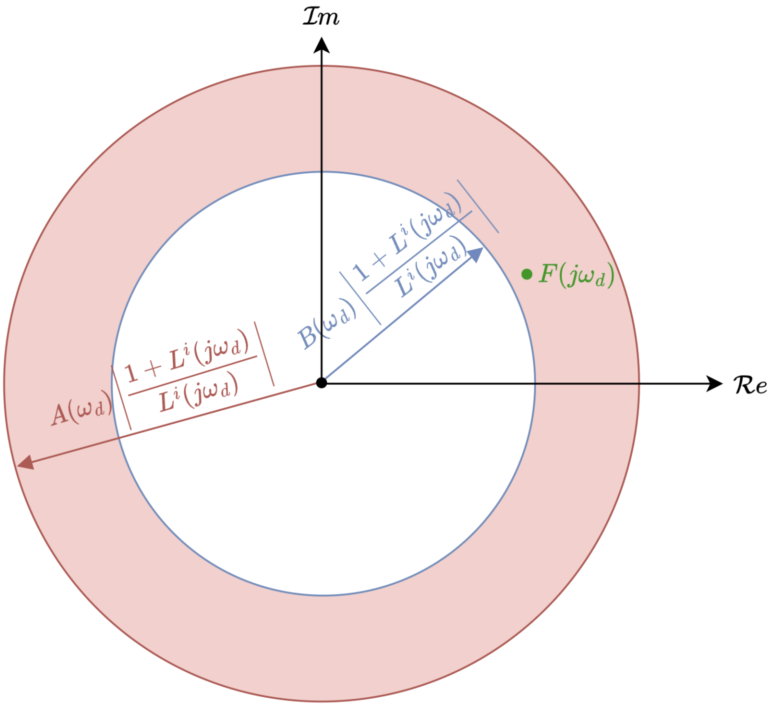


Figure 6.3: Arithmetic-complex plane, showing the solution space of pre-filter $F(j\omega_d)$ for plant instance i and design frequency ω_d.

where i and k indicate two distinct plant instances. With some reordering, (6.13), takes the form of a linear fractional mapping,

$$\frac{\left|1/P^i(j\omega)+G(j\omega)\right|}{\left|1/P^k(j\omega)+G(j\omega)\right|} \le \frac{A(\omega)}{B(\omega)}, \forall i, k, \tag{6.14}$$

which is used to generate and solve quadratic inequalities that describe the admissible solution space of $G(j\omega)$ for a particular design frequency ($\omega \to \omega_d$) and feedback control phase angle ($\angle\{G(j\omega)\}$). QFT practitioners commonly make use of a log-polar plot to display the design regions of $G(j\omega)$, which is traditionally scaled and shifted by an arbitrarily selected nominal plant to represent a Nichols chart. Following a valid selection of $G(s)$ (if the specifications are compatible with the non-minimum phase plant constraints), the design of $F(s)$ ensues using (6.9).

MIMO design formulation

A multi-input-multi-output equivalent exists for magnitude-based tracking bounds, which makes use of the same underlying design methodology [80]. The multivariate equivalent of (6.7) is written as

$$\mathbf{T}^i(s) = \left[\mathbf{I}+\mathbf{P}^i(s)\mathbf{G}(s)\right]^{-1}\mathbf{P}^i(s)\mathbf{G}(s)\mathbf{F}(s). \tag{6.15}$$

Plant inversion is then used to rearrange (6.15) as

$$\left[\hat{\mathbf{P}}^i(s) + \mathbf{G}(s)\right] \mathbf{T}^i(s) = \mathbf{G}(s)\mathbf{F}(s), \tag{6.16}$$

where $\hat{\mathbf{P}}^i(s)$ represents the inverse of $\mathbf{P}^i(s)$. Assuming $\mathbf{G}(s)$ is diagonal and a diagonal splitting is applied to (6.16), $\mathbf{T}^i(s)$ can be implicitly determined as

$$t^i_{rc}(s) = \frac{1}{p^i_{rr}(s) + g_{rr}(s)} \left[g_{rr}(s) f_{rc}(s) - \sum_{k \neq r} \hat{p}^i_{rk}(s) t^i_{kc}(s) \right], \tag{6.17}$$

where $t^i_{rc}(s)$ denotes the (r, c) element of $\mathbf{T}^i(s)$. Note that the diagonal splitting results in a non-sequential design. Analogous to (6.6), an above and below magnitude is placed on each element of $\mathbf{T}^i(j\omega)$

$$\beta_{rc}(\omega) \leq \left| t^i_{rc}(j\omega) \right| \leq \alpha_{rc}(\omega), \ \forall i. \tag{6.18}$$

Substituting the frequency response of (6.17) into (6.18) into yields

$$\beta_{rc}(\omega) \leq \left| \frac{g_{rr}(j\omega) f_{rc}(j\omega) - \sum_{k \neq r} \hat{p}^i_{rk}(j\omega) t^i_{kc}(j\omega)}{p^i_{rr}(j\omega) + g_{rr}(j\omega)} \right| \leq \alpha_{rc}(\omega), \ \forall i. \tag{6.19}$$

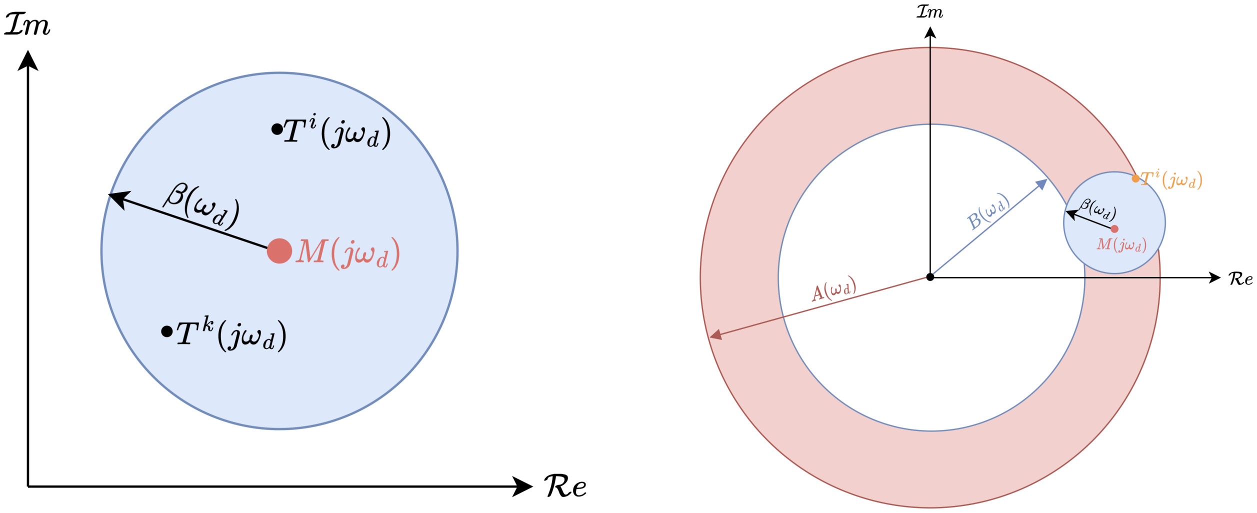
The $t_{kc}(j\omega)$ elements in (6.19) are unknown prior to design of $\mathbf{G}(s)$ and $\mathbf{F}(s)$. Following the MIMO QFT design methodology, $\left| t^i_{rc}(j\omega) \right|$ is overbounded using the triangle inequality

$$\beta_{rc}(\omega) \leq \left| \frac{g_{rr}(j\omega) f_{rc}(j\omega)}{p^i_{rr}(j\omega) + g_{rr}(j\omega)} \right| \pm \left| \frac{\sum_{k \neq r} \hat{p}^i_{rk}(j\omega) t^i_{kc}(j\omega)}{p^i_{rr}(j\omega) + g_{rr}(j\omega)} \right| \leq \alpha_{rc}(\omega), \ \forall i. \tag{6.20}$$

Note that the overbounding assumes the worst-case gain and phase combination for both the above and below bound. The procedures follows the SISO design methodology that leads to (6.9), namely manipulating (6.20) until one obtains magnitude constraints on $f_{rc}(j\omega)$, where the unknown $t_{kc}(j\omega)$ elements are overbounded by $\beta_{rc}(\omega)$ or $\alpha_{rc}(\omega)$ depending on the worst-case instance. A sequential design procedure also exists, where information from previous loop closures is carried into subsequent feedback controller element designs. The details are spared here as this method is not formative to the QFT contributions presented in this thesis.

6.1.3 Model-based tracking formulation

Bounding the magnitude response of the reference-to-output transfer behaviour has the limitation in that the corresponding phase response is not specified, as shown Figure 6.2b. Given that the performance specifications, $\{A(\omega, B(\omega)\}$, commonly originate from time-domain requirements, the unconstrained phase response can lead to closed-loop behaviour that may not be well aligned with that of the desired tracking behaviour. A more recent contribution to QFT-based reference tracking design methodologies involves bounding both the closed-loop gain



(a) (b)

Figure 6.4: Arithmetic-complex plane, showing (a) the geometric representation of the tracking error design constraint in (6.21) for design frequency point ω_d. Satisfying (6.21) requires that each closed-loop plant instance reside within the (blue) *stay-in* disc. A comparison of the design regions of $T^i(j\omega_d)$ for the above/below tracking bound methodology in (6.6) and model-error tracking design in (6.21) are shown in (b).

and phase behaviour, in order to enforce frequency-domain constraints that better match that of the time-domain counterparts [15].

The most prevalent of QFT-based tracking designs makes use of a single reference model, $M(j\omega)$, often derived from a transfer representation, $M(s)$, which represents the idealised gain and phase behaviour of $T^i(j\omega)$ in the frequency domain. The design objective is then to configure the controllable elements of $T^i(j\omega)$ such that $T^i(j\omega) \to M(j\omega)$ to within a predefined tolerance,

$$\left|M(j\omega) - T^i(j\omega)\right| \leq \beta(\omega), \ \forall i, \tag{6.21}$$

where $\beta(\omega)$ is the reference model tracking tolerance that dictates how closely the set of $T^i(j\omega) \in \mathcal{T}$ must approach $M(j\omega)$ at the particular design frequency. The geometric representation of (6.21) is shown in the arithmetic-complex plane in Figure 6.4a, where $T^i(j\omega)$, $\forall i$ is required to reside in the disk of centre $M(j\omega)$ and radius $\beta(\omega)$ in order to adhere to the design constraint across all plant cases. A visual comparison between the model-error tracking formulation in (6.21) and magnitude bounding approach from (6.6) is shown in Figure 6.4b.

6.1.4 Tracking error bounds using pre-filter synthesis

SISO design formulation

The model-based approach in Section 6.1.3 requires separation of the pre-filter and feedback control elements, in order to generate univariate design boundaries on the control elements $\{G(j\omega), F(j\omega)\}$. One approach, introduced in [15], is to synthesize the pre-filter based on the selection of an appropriate nominal open-loop plant, $P^0(s)$. Given that $M(s)$ describes the idealised transfer behaviour from reference to output, setting the nominal closed-loop transfer behaviour as

$$T^0(s) = \frac{P^0(s)G(s)F(s)}{1 + P^0(s)G(s)} = M(s) \tag{6.22}$$

implies that $G(s)$ and $F(s)$ are constrained such that the nominal closed-loop system is identical to that of the idealised reference to output behaviour. Using this relationship, the pre-filter is synthesized as

$$F(s) = M(s)\frac{1 + P^0(s)G(s)}{P^0(s)G(s)}. \tag{6.23}$$

Note that this selection of $F(s)$ is made prior to the design of $G(s)$ and can only be calculated after completing the feedback control design. Following this, (6.21) can be written explicitly as

$$\begin{aligned} \left|M(j\omega) - T^i(j\omega)\right| &\leq \beta(\omega),\ \forall i, \\ \left|M(j\omega) - \frac{P^i(j\omega)G(j\omega)F(j\omega)}{1 + P^i(j\omega)G(j\omega)}\right| &\leq \beta(\omega),\ \forall i, \\ |M(j\omega)|\left|1 - \frac{1/P^0(j\omega) + G(j\omega)}{1/P^i(j\omega) + G(j\omega)}\right| &\leq \beta(\omega),\ \forall i. \end{aligned} \tag{6.24}$$

After some simplification, the constraint in (6.24) simplifies to

$$\begin{aligned} \frac{\left|M(j\omega)/P^0(j\omega) - M(j\omega)/P^i(j\omega)\right|}{|1/P^i(j\omega) + G(j\omega)|} &\leq \beta(\omega) \\ |M(j\omega)|\left|1/P^0(j\omega) - 1/P^i(j\omega)\right| &\leq \left|1/P^i(j\omega) + G(j\omega)\right|\beta(\omega) \\ \frac{|M(j\omega)|}{\beta(\omega)}\left|1/P^0(j\omega) - 1/P^i(j\omega)\right| &\leq \left|G(j\omega) - \left(-1/P^i(j\omega)\right)\right|, \end{aligned} \tag{6.25}$$

which describes a discoidal *stay-out* design region of $G(j\omega)$, with a centre of $-1/P^i(j\omega)$, and a corresponding radius of $\frac{|M(j\omega)|}{\beta(\omega)}\left|1/P^0(j\omega) - 1/P^i(j\omega)\right|$, as shown in Figure 6.5. The set of design constraints in (6.25) takes the form of a linear fractional mapping and can be used to generate the admissible design regions of $G(j\omega)$ for each discrete design frequency of interest. Selection of $G(s)$ then dictates the complexity of $F(s)$ based on (6.23). Notably, the designer has no direct control over the order and structure of $F(s)$. As such, depending on the designated nominal plant and selected feedback controller, the pre-filter may be unecesarily high-order.

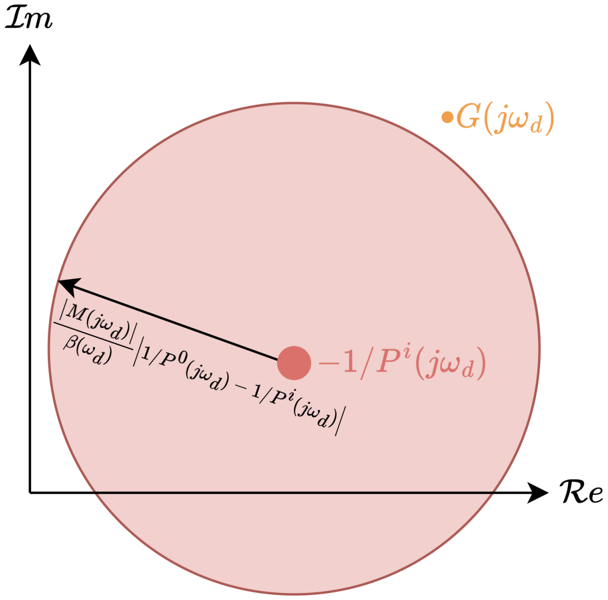


Figure 6.5: Arithmetic-complex plane, showing the geometric representation of the tracking error design constraint in (6.25) for design frequency point ω_d. Satisfying (6.25) requires that $G(j\omega_d)$ be located outside of the (red) *stay-out* disc.

Selection of the nominal plant, $P^0(s)$, has a direct impact on the radius of the *stay-out* in (6.25), and by extension, the solution space of $G(j\omega)$ in Figure 6.5.

Pre-filter refinement

An improved design methodology that builds on the model-based tracking error paradigm from Section 6.1.3 was presented in [18], whereby the pre-filter is refined after $G(s)$ has been selected. With a known feedback controller element, (6.24) describes a constraint set on the pre-filter element, namely

$$\left| M(j\omega) - \frac{P^i(j\omega)G(j\omega)F(j\omega)}{1 + P^i(j\omega)G(j\omega)} \right| \leq \beta(\omega),\ \forall i, \tag{6.26}$$

where $F(s)$ is the only unknown. After some simplification, (6.26) can be represented as

$$\left| \frac{M(j\omega)\left(1 + P^i(j\omega)G(j\omega)\right)}{P^i(j\omega)G(j\omega)} - F(j\omega) \right| \leq \beta(\omega)\frac{\left|1 + P^i(j\omega)G(j\omega)\right|}{\left|P^i(j\omega)G(j\omega)\right|},\ \forall i, \tag{6.27}$$

which describes the discoidal *stay-in* design region of $F(j\omega)$, with a centre of $M(j\omega)\frac{1+P^i(j\omega)G(j\omega)}{P^i(j\omega)G(j\omega)}$, and radius of $\beta(\omega)\frac{\left|1+P^i(j\omega)G(j\omega)\right|}{\left|P^i(j\omega)G(j\omega)\right|}$, as shown in Figure 6.6. In this way, the control practitioner has some agency over the structure and order of $F(s)$, subject to the generated design regions. Notably, the design boundaries that originate from (6.27) are biased based on the selection of $G(s)$, and by extension, the initial selection of $P^0(s)$. Also worth highlighting is the fact that phase angle constraints exist for $F(j\omega)$ using the tracking error paradigm

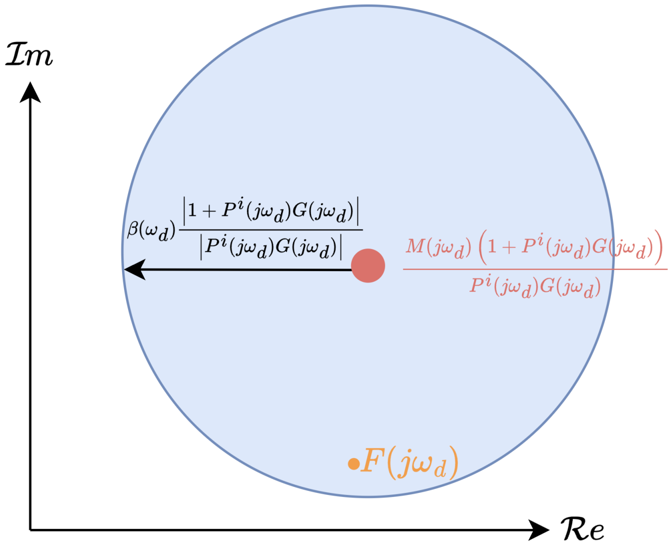


Figure 6.6: Arithmetic-complex plane, showing the geometric representation of the tracking error design constraint in (6.27) for design frequency point ω_d. Satisfying (6.27) requires that $F(j\omega_d)$ be located inside of the (blue) *stay-in* disc.

in (6.27), whereas satisfying the above/below constraints in (6.10) is not dependant on the phase behaviour of $F(j\omega)$.

6.1.5 Model-error tracking using existence conditions

The bound generation approach in Section 6.1.4 necessitates selection of a nominal plant, which can contribute to design conservatism — up to 6 dB at worst [15]. The work in [4] provides a model-based tracking error design methodology predicated on existence conditions, which draw certain parallels to the approach detailed in Section 6.1.2. With reference to Figure 6.7, the two-degree-of-freedom control structure comprises the feedback controller, $G(s)$, and feedforward filter, $X(s)$. Additionally, the reference model, $M(s)$, is applied directly to the reference signal $r(s)$.

Expanding (6.21) for the configuration in Figure 6.7 results in

$$\left|\frac{M(j\omega)-P^i(j\omega)X(j\omega)}{1+P^i(j\omega)G(j\omega)}\right| = \left|S^i(j\omega)\right|\left|M(j\omega)-P^i(j\omega)X(j\omega)\right| \leq \beta(\omega). \tag{6.28}$$

Analogous to the pre-filter synthesis in Section 6.1.4, $X(j\omega)$ can be synthesized to make the numerator in (6.28) small over the plant set, thereby reducing the burden on the feedback loop to make the sensitivity function small. With some manipulation, (6.28) can be reposed as

$$\left|M(j\omega)/P^i(j\omega)-X(j\omega)\right| \leq \beta(\omega)\left|1/P^i(j\omega)+G(j\omega)\right|, \tag{6.29}$$

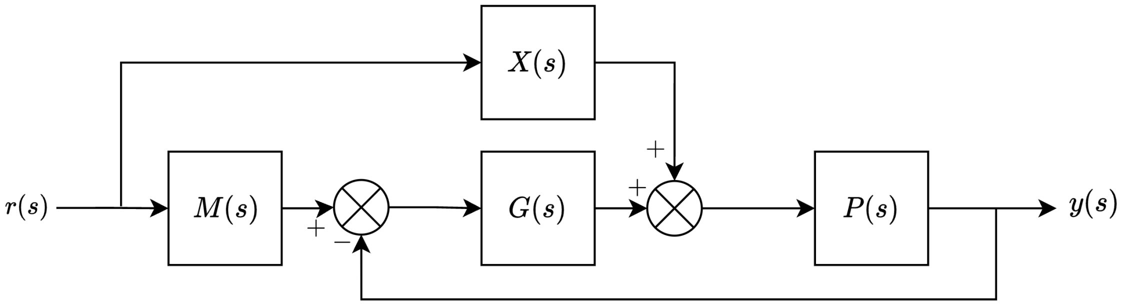


Figure 6.7: Block diagram of traditional two-degree-of-freedom control scheme, where feedforward filter $X(s)$ maps the reference signal to the plant input, and feedback controller $G(s)$ acts on the filtered reference tracking error.

which describes the solution space of $X(j\omega)$ for plant instance i. The geometric representation of the solution space is a *stay-in* disc, with centre $M(j\omega)/P^i(j\omega)$ and radius $\beta(\omega)\left|1/P^i(j\omega)+G(j\omega)\right|$, as shown in Figure 6.8. Given any two arbitrary plant cases, $\{i,k\}$, the existence of at least one valid $X(j\omega)$ that can satisfy both

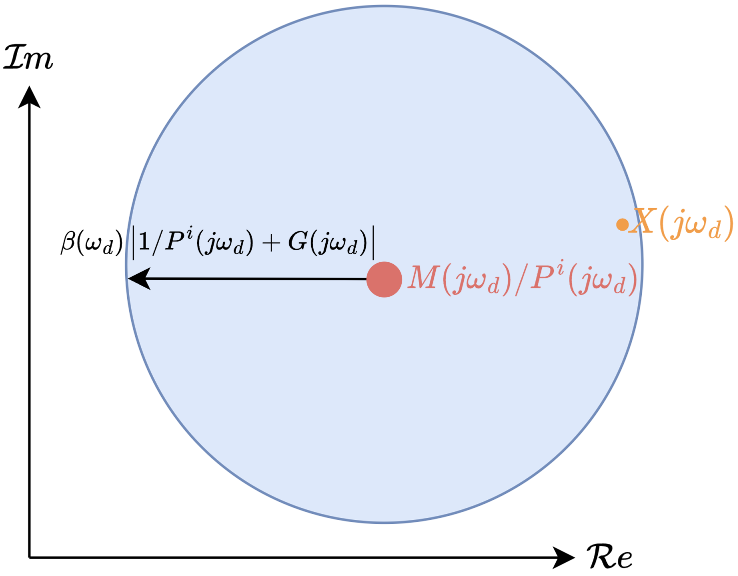


Figure 6.8: Arithmetic-complex plane, showing the geometric representation of the tracking error design constraint in (6.28) for design frequency point ω_d. Satisfying (6.28) requires that $X(j\omega_d)$ be located inside of the (blue) *stay-in* disc.

constraints necessitates an overlap of the two *stay-in* discs, as shown in Figure 6.9. Specifically, this requires that the distance between the two disc centres is smaller than the sum of the corresponding radii, namely

$$\left|M(j\omega)/P^i(j\omega)-M(j\omega)/P^k(j\omega)\right| \leq \beta(\omega)\left|1/P^i(j\omega)+G(j\omega)\right|+\beta(\omega)\left|1/P^k(j\omega)+G(j\omega)\right|. \tag{6.30}$$

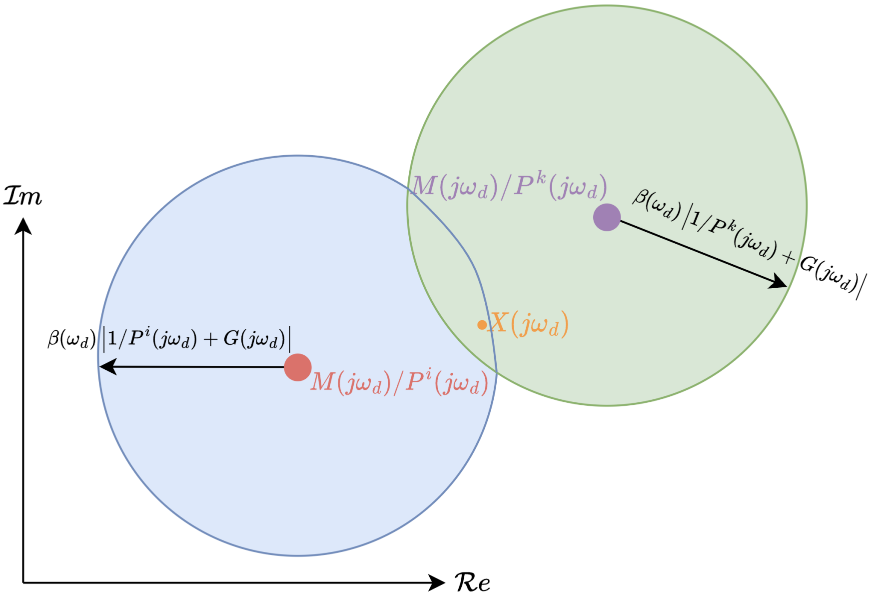


Figure 6.9: Arithmetic-complex plane, showing two *stay-in* discs with a common intersection for design frequency point ω_d.

Equation (6.30) describes the necessary and sufficient existence condition of $X(j\omega)$ for plant pair i-k. By extension, the existence of a feedforward element that can satisfy (6.28) for every possible plant pair is

$$\left|M(j\omega)/P^i(j\omega) - M(j\omega)/P^k(j\omega)\right| \leq \beta(\omega)\left|1/P^i(j\omega) + G(j\omega)\right| + \beta(\omega)\left|1/P^k(j\omega) + G(j\omega)\right|, \ \forall i, k. \quad (6.31)$$

The set of design constraints in (6.31) can be converted into quadratic inequalities that describe the admissible solution space of $G(j\omega)$ for a particular design frequency and feedback controller phase angle. Notably, the constraint set in (6.31) is not biased by the selection of a nominal plant, thereby offering a nonconservative design approach for the 2DOF tracking error problem.

When visualised in the arithmetic-complex plane, the per-plant solution space of $G(j\omega)$ from (6.31) is a *stay-out* ellipse, with foci at $-1/P^i(j\omega)$ and $-1/P^k(j\omega)$, and a major axis of $\left|M(j\omega)/P^i(j\omega) - M(j\omega)/P^k(j\omega)\right|/\beta(\omega)$, as shown in Figure 6.10. Successful design of $G(s)$ gives rise to the non-empty solution space of $X(j\omega)$, based on (6.30). Analogous to the design procedure in Section 6.1.2, the solution space of $X(j\omega)$ will be dependant on the selection of $G(s)$. That is, as $G(j\omega)$ tends towards the boundary limits, the corresponding solution space of $X(j\omega)$ will collapse to a point. In this regard, the designer is required to allocate headroom when designing $G(s)$, in order to enjoy some design freedom during the selection of $X(s)$. Notably, a non-empty solution space for every design frequency spanning $X(j\omega)$ does not, in general, guarantee that a causal, rational $X(s)$ can be selected that meets all the prescribed frequency-domain design constraints.

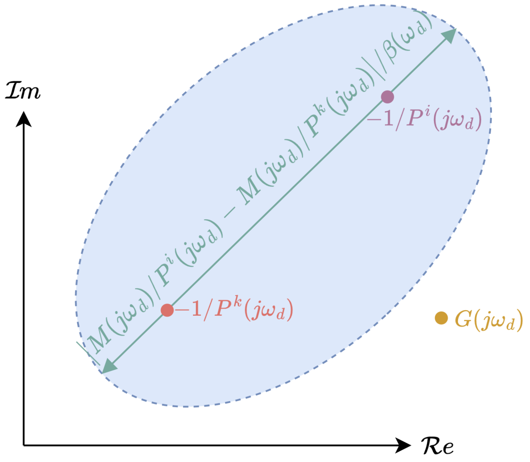


Figure 6.10: Arithmetic-complex plane, showing ellipsoidal *stay-out* design region of $G(j\omega_d)$ from (6.31) for design frequency point ω_d.

As noted in [4], adopting the classical 2DOF control structure from Figure 6.1, and following the same procedure of defining existence conditions for $F(j\omega)$, one will still arrive at the same design constraints in (6.31). The two control architectures in Figure 6.1 and Figure 6.7 are therefore mathematically equivalent and related by

$$F(s) = M(s) + G^{-1}(s)X(s). \tag{6.32}$$

Non-sequential approach to MIMO systems

The bound generation approach outlined in Section 6.1.5 was extended to multi-input-multi-output systems in [5] and [81], with the latter also encapsulating bound generation when plant disturbances are measurable. Using the feedforward structure in Figure 6.7, the multivariable representation of (6.7) is expressed as

$$\mathbf{T}^i(s) = \left[\mathbf{I} + \mathbf{P}^i(s)\mathbf{G}(s)\right]^{-1} \mathbf{P}^i(s) \left[\mathbf{G}(s)\mathbf{M}(s) + \mathbf{X}(s)\right], \tag{6.33}$$

where the boldface variables are $n \times n$ matrix counterparts of the SISO elements. By extension, the transfer matrix from the reference vector to the error vector is

$$\mathbf{E}^i(s) = \left[\mathbf{I} + \mathbf{P}^i(s)\mathbf{G}(s)\right]^{-1} \left[\mathbf{M}(s) - \mathbf{P}^i(s)\mathbf{X}(s)\right]. \tag{6.34}$$

Provided that $\mathbf{P}^i(s)$ is nonsingular for all plant instances and design frequencies of interest, (6.34) can be represented as

$$\left[\hat{\mathbf{P}}^i(s) + \mathbf{G}(s)\right] \mathbf{E}^i(s) = \hat{\mathbf{P}}^i(s)\mathbf{M}(s) - \mathbf{X}(s), \tag{6.35}$$

where $\hat{\mathbf{P}}^i(s)$ represents the inverse of $\mathbf{P}^i(s)$. Assuming $\mathbf{G}(s)$ is designed to be diagonal, a diagonal splitting can be applied to the matrix coefficient of $\mathbf{E}^i(s)$ in (6.35), which results in

$$\hat{\mathbf{P}}^i_n(s)\mathbf{E}^i(s) + \left[\hat{\mathbf{P}}^i_d(s) + \mathbf{G}(s)\right]\mathbf{E}^i(s) = \hat{\mathbf{P}}^i(s)\mathbf{M}(s) - \mathbf{X}(s), \tag{6.36}$$

where $\hat{\mathbf{P}}^i_d(s)$ and $\hat{\mathbf{P}}^i_n(s)$ indicate the diagonalised and off-diagonal matrix of $\hat{\mathbf{P}}^i(s)$, respectively. The implicit matrix transfer behaviour from reference to error follows as

$$\mathbf{E}^i(s) = \left[\hat{\mathbf{P}}^i_d(s) + \mathbf{G}(s)\right]^{-1}\left[\hat{\mathbf{P}}^i(s)\mathbf{M}(s) - \mathbf{X}(s) - \hat{\mathbf{P}}^i_n(s)\mathbf{E}^i(s)\right]. \tag{6.37}$$

Using $\hat{p}^i_{rc}(s)$ to represent the (r,c)-element of $\hat{\mathbf{P}}^i(s)$, (6.37) can be expressed in element-wise form as

$$e^i_{rc}(s) = \frac{1}{\hat{p}^i_{rr}(s) + g_{rr}(s)}\left[\sum_{a=1}^{n}\hat{p}^i_{ra}(s)m_{ac}(s) - x_{rc}(s) - \sum_{a\neq 1}\hat{p}^i_{ra}(s)e^i_{ac}(s)\right]. \tag{6.38}$$

Analogous to (6.21), the multivariable tracking design method is formulated based on overbounding each element in magnitude

$$\left|e^i_{rc}(j\omega)\right| \leq \beta_{rc}(\omega), \tag{6.39}$$

which is written explicitly as

$$\left|\frac{\left[\sum_{a=1}^{n}\hat{p}^i_{ra}(j\omega)m_{ac}(j\omega) - x_{rc}(j\omega) - \sum_{a\neq 1}\hat{p}^i_{ra}(j\omega)e^i_{ac}(j\omega)\right]}{\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)}\right| \leq \beta_{rc}(\omega). \tag{6.40}$$

Equation (6.40) cannot be evaluated directly because of the unknown $e^i_{ac}(j\omega)$ terms. Using Horowitz' original design methodology [82], (6.40) can be overbounded using the triangle inequality as

$$\left|e^i_{rc}(j\omega)\right| \leq \left|\frac{\sum_{a=1}^{n}\hat{p}^i_{ra}(j\omega)m_{ac}(j\omega) - x_{rc}(j\omega)}{\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)}\right| + \frac{\sum_{a\neq 1}\left|\hat{p}^i_{ra}(j\omega)\right|\beta_{ac}(\omega)}{\left|\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)\right|} \leq \beta_{rc}(\omega). \tag{6.41}$$

Satisfying the right-hand side inequality from (6.41),

$$\left|\frac{\sum_{a=1}^{n}\hat{p}^i_{ra}(j\omega)m_{ac}(j\omega) - x_{rc}(j\omega)}{\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)}\right| + \frac{\sum_{a\neq 1}\left|\hat{p}^i_{ra}(j\omega)\right|\beta_{ac}(\omega)}{\left|\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)\right|} \leq \beta_{rc}(\omega) \tag{6.42}$$

therefore implies that (6.39) is also satisfied. The set of design constraints in (6.41) are then apportioned into the two constraint sets of

$$\left|\frac{\sum_{a=1}^{n}\hat{p}^i_{ra}(j\omega)m_{ac}(j\omega) - x_{rc}(j\omega)}{\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)}\right| \leq \beta_{rc}(\omega)\left[1 - \gamma_{rc}(\omega)\right], \tag{6.43}$$

$$\frac{\sum_{a\neq 1}\left|\hat{p}^{i}_{ra}(j\omega)\right|\beta_{ac}(\omega)}{\left|\hat{p}^{i}_{rr}(j\omega)+g_{rr}(j\omega)\right|}\leq \beta_{rc}(\omega)\gamma_{rc}(\omega), \tag{6.44}$$

where $\gamma_{rc}(\omega) \in (0, 1)$ determines the relative weightings of the two constraint sets. $\gamma_{rc}(\omega)$ is selected manually by the user on a per-frequency basis, with the aim of apportioning the division of tolerances such that the bounds generated from (6.43) and (6.44) are similar in size. Assuming one can specify $\gamma_{rc}(\omega)$ reasonably, (6.44) represents a SISO equivalent sensitivity design, albeit with overdesign due to the application of the triangle inequality, and can be used to generate valid design regions on $g_{rr}(j\omega)$ using pre-existing methods. However, the constraint set in (6.43) is dependant on $g_{rr}(j\omega)$ and $x_{rc}(j\omega)$. Notably, one option would be to synthesize $x_{rc}(j\omega)$, as in the pre-filter synthesis approach in Section 6.1.4. To circumvent having to solve (6.43) for both $x_{rc}(j\omega)$ and $g_{rr}(j\omega)$, or rely on synthesis, the procedure of using existence conditions from Section 6.1.5 is applied. Specifically, (6.43) can be reposed as

$$\left|\sum_{a=1}^{n}\hat{p}^{i}_{ra}(j\omega)m_{ac}(j\omega)-x_{rc}(j\omega)\right|\leq \beta_{rc}(\omega)\left(1-\gamma_{rc}(\omega)\right)\left|\hat{p}^{i}_{rr}(j\omega)+g_{rr}(j\omega)\right|, \tag{6.45}$$

which describes the discoidal solution space of $x_{rc}(j\omega)$, with centre $\sum_{a=1}^{n}\hat{p}^{i}_{ra}(j\omega)m_{ac}(j\omega)$ and radius $\beta_{rc}(\omega)\left(1-\gamma_{rc}\right)\left|\hat{p}^{i}_{rr}(j\omega)+g_{rr}(j\omega)\right|$. The existence of an $x_{rc}(j\omega)$ element that satisfies (6.45) for two plant instances, (i, k), necessitates an intersection of the two discs, which requires that the distance between the two disc centres be smaller than or equal to the sum of the corresponding radii:

$$\begin{aligned}&\left|\sum_{a=1}^{n}\hat{p}^{i}_{ra}(j\omega)m_{ac}(j\omega)-\sum_{a=1}^{n}\hat{p}^{k}_{ra}(j\omega)m_{ac}(j\omega)\right|\\&\qquad\leq \beta_{rc}(\omega)\left(1-\gamma_{rc}(\omega)\right)\left[\left|\hat{p}^{i}_{rr}(j\omega)+g_{rr}(j\omega)\right|+\left|\hat{p}^{k}_{rr}(j\omega)+g_{rr}(j\omega)\right|\right].\end{aligned} \tag{6.46}$$

The resulting design constraint in (6.46) can then be extended across all plant pairs, which collectively describes the admissible solution space of $g_{rr}(j\omega)$. As in the SISO counterpart of (6.31), the constraint set in (6.46) represents ellipsoidal *stay-out* design regions on $g_{rr}(j\omega)$. Conversion from design constraints to visualisable log-polar bounds is facilitated by representing (6.46) as a quadratic inequality, followed by determining the values of $g_{rr}(j\omega)$ for which the quadratic polynomial remains positive (see [5] for more information). Note that $g_{rr}(s)$ must simultaneously satisfy the design constraints in (6.44) and (6.46). Assuming the feedback control design is successful, the selected $g_{rr}(s)$ elements are then used to populate the set of constraints in (6.45), which gives rise to the discoidal *stay-in* design regions on $x_{rc}(j\omega)$.

Sequential approach to MIMO systems

The authors in [5] and [81] additional provide a sequential design routine that reduces the design conservatism for subsequent control loop closures. Instead of applying a diagonal splitting as in (6.36), a triangular splitting is adopted, which replaces (6.38) with

$$e^i_{rc}(s) = \frac{1}{{}^r\hat{p}^i_{rr}(s) + g_{rr}(s)}\left[\sum_{a=1}^{n} {}^r\hat{p}^i_{ra}(s)m_{ac}(s) + \sum_{a<r}\frac{{}^a\hat{p}_{ra}{}^a f_{ac}}{{}^a\hat{p}^i_{aa} + g_{aa}} - x_{rc}(s) - \sum_{a>r} {}^r\hat{p}^i{}_{ra}(s)e^i_{ac}(s)\right], \qquad (6.47)$$

where ${}^r\hat{p}^i_{rr}(s)$ and ${}^a f_{ac}(s)$ are the equivalent plant and feedforward element, respectively. These terms incorporate the previously designed elements of $\mathbf{G}(s)$ and $\mathbf{F}(s)$, based on

$${}^{v+1}\hat{p}^i_{rc}(s) = {}^v\hat{p}^i_{rc}(s) - \frac{{}^v\hat{p}^i_{ra}(s){}^v\hat{p}^i_{ac}(s)}{{}^v\hat{p}^i_{vv}(s) + g_{rr}(s)}, \qquad (6.48)$$

$${}^{v+1}\hat{f}_{rc}(s) = {}^v\hat{f}_{rc}(s) - \frac{{}^v\hat{p}^i_{ra}(s){}^v\hat{f}_{ac}(s)}{{}^v\hat{p}^i_{vv}(s) + g_{rr}(s)}, \qquad (6.49)$$

where $r \geq 1$, ${}^1\hat{p}^i_{rc}(s) = \hat{p}^i_{rc}(s)$, and ${}^1\hat{f}_{rc}(s) = \hat{f}_{rc}(s)$. Following this, (6.43) and (6.44) are updated as

$$\left|\frac{{}^r\hat{p}^i_{rr}(j\omega)m_{rc}(j\omega) + \sum_{a\neq r}^{n} {}^r\hat{p}^i_{ra}(j\omega)m_{ac}(j\omega) - x_{rc}(j\omega) + \sum_{a<r}\frac{{}^a\hat{p}_ra(j\omega){}^a f_{ac}(j\omega)}{{}^a\hat{p}_{aa}(j\omega)+g_{aa}(j\omega)}}{\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)}\right| \leq \beta_{rc}(\omega)\left(1 - \gamma_{rc}(\omega)\right), \qquad (6.50)$$

$$\frac{\sum_{a>r}\left|{}^r\hat{p}^i_{ra}(j\omega)\right|\beta_{ac}(\omega)}{\left|{}^r\hat{p}^i_{rr}(j\omega) + g_{rr}(j\omega)\right|} \leq \beta_{rc}(\omega)\gamma_{rc}(\omega). \qquad (6.51)$$

As with (6.43) and (6.44), the design equations in (6.50) and (6.51) can be converted into log-polar boundaries for use in selecting a valid $g_{rr}(s)$, followed by selection of $x_{rc}(s)$.

The non-sequential and sequential approaches outlined in this Section has the notable limitation in that the relative bound weighting, $\gamma(\omega)$, is selected on a per-frequency basis. Consulting the design constraints on $g_{rr}(j\omega)$ ((6.44) and (6.46) for the non-sequential approach), the two corresponding arithmetic-complex design regions represent different geometric shapes — the former being a disc, and the latter an ellipse. As such, if a common $\gamma(\omega)$ is used to balance the two boundaries at each design frequency, the two boundaries will, in general, never be coincident at all controller phase angles. Instead, one boundary will tend to dominate the other boundary over a range of controller phase angles, resulting in additional design conservatism. One could instead select the relative bound weighting on a per-frequency and per-phase basis, in order to force the two boundaries to converge on each other at all phase angles, but this concept is absent from [5] and [81].

6.1.6 Extension to QFT contributions

This section provides a summary of the formative work from other authors that was used to develop the published QFT contributions contained in Section 6.4-6.6.

Robust plant by plant control design using model-error tracking sets (Section 6.4)

The bound generation routine developed in Section 6.4 is based on the SISO design methodology in Section 6.1.5 [4]. Specifically, the same concept of using existence conditions across discoidal solution spaces is applied to reduce the design problem to a set of univariate mathematical constraints on $G(j\omega)$, as in (6.31). With reference to Figure 6.11, the contribution in Section 6.4 extends the model-based tracking error concept such that each plant case has a predefined reference model, instead of using a single reference model across all plant instances. This approach results in different mathematical constraints that can be converted into geometric design regions using quartic inequalities. The resulting design regions on $G(j\omega)$ are either *stay-in* or *stay-out*, depending on the relative locations of the per-plant reference models, and corresponding error tolerances.

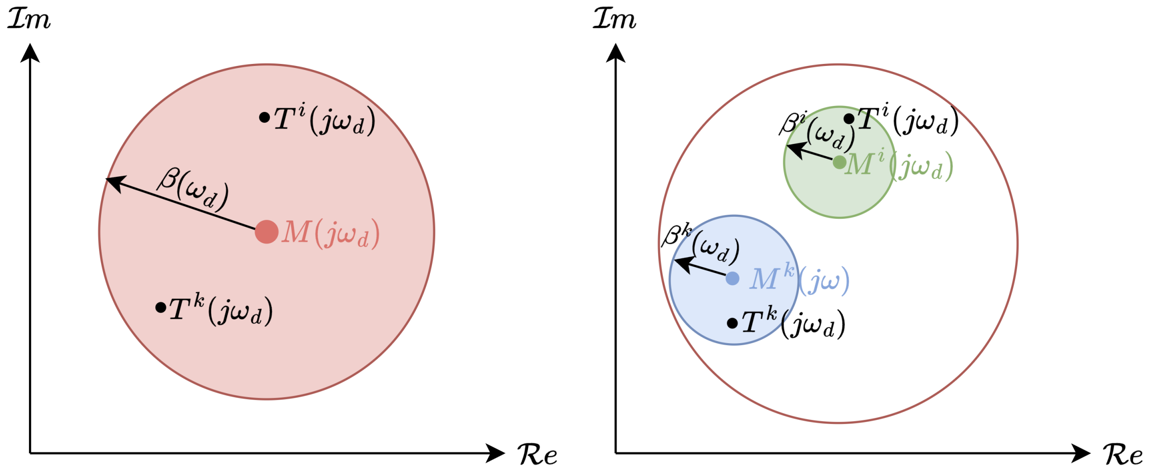


Figure 6.11: Arithmetic-complex plane, showing the *stay-in* design regions of the reference to to output transfer behaviour at design frequency point ω_d, for (left) a single reference model specification, used in [4], and (right) a per-plant reference model specification, used in Section 6.4.

While the bound generation procedure in Section 6.4 is conceptually similar to that of Section 6.1.5, the notable differences include using simple optimisation to generate feasible, per-plant reference models, as well as incorporating a level set of tracking error tolerances — the combination of these two innovations are aimed at alleviating the induced design strictures that arise when incompatible design specifications are prescribed. For more information, see Section 6.4.

A complementary QFT solution to the 2x2 tracking error problem (Section 6.5)

The prior work from Section 6.1.5 [5] on the MIMO tracking error bound generation is formative to the contribution presented in Section 6.5. Specifically, the same nonsequential methodology is followed until (6.40) for a 2×2 system. However, the contributed work then differs by defining existence conditions on the feedforward filter directly, instead of using a division of tolerances as in (6.43) and (6.44). The result is that the offering in

Section 6.5 does not require visual, manual adjustment of the $\gamma_{rc}(\omega)$ weightings, and presents a more elegant solution that, in general, is less conservative than the nonsequential approach from [5]. For more information, see Section 6.5.

The contribution in Section 6.5 significantly diverges from the prior work in Section 6.1.5, by deriving a set of mathematical constraints on the diagonal elements of $\mathbf{G}(s)$ that are born from a non-inverting plant methodology. This novel approach generates two sets of design constraints on the diagonal elements of $\mathbf{G}(s)$, with the union of the two corresponding design regions describing the globally admissible solution space. As a result, this approach can significantly reduce conservatism at the mid- to high-frequency band, including around the gain-phase crossover region. Another notable difference between the formative work in Section 6.1.5 and contribution in Section 6.5 is that the design constraints used to form the pre-filter solution space are different. The methodology outlined in Section 6.1.5 uses the conservative constraints from (6.45) to design $x_{rc}(s)$ after $g_{rr}(s)$ has been selected, whereas the contribution in Section 6.5 uses an entirely new bound generation routine to select $x_{rc}(s)$ that does not contain any induced conservatism, aside from the (potentially) sub-optimal selection of $g_{rr}(s)$. As a result, the solution space of $x_{rc}(s)$ is, in general, larger for the proposed method.

A refinement approach to the multivariable tracking error problem (Section 6.6)

Section 6.6 also makes use of the preliminary MIMO development in Section 6.1.5 [5]. As with Section 6.5, Section 6.6 follows the same design procedure as detailed in Section 6.1.5, up until (6.40). A generalised multivariable adaptation of Section 6.5 is then developed using the same concept of discoidal existence conditions.

In contrast to the work in Section 6.1.5 or contribution in Section 6.5, the offering in Section 6.6 details a refinement approach that can exploit information from previous feedback control designs in order to reduce the design conservatism that is induced when using the triangle inequality. Additionally, the 2×2 feedforward filter bound generation methodology in Section 6.5 is abstracted to a generalised $n \times n$ system without the need for overbounding gain-phase behaviour. For more information, see Section 6.6.

6.2 Developments in related multivariable frequency-domain methods

This Section summarises related design and synthesis techniques that make use of the multivariable frequency-domain framework. The intention is to draw common threads between these approaches and modern QFT, whilst also highlighting their relative strengths and weaknesses.

6.2.1 Inverse Nyquist array

Design of multivariable control systems using the inverse Nyquist array (INA) was first presented in [83]. Given a multivariable plant under consideration, which is described by an $m \times m$ rational transfer function $\mathbf{P}(s)$, and an accompanying feedback controller $\mathbf{G}(s)$ with matching size, the objective is design $\mathbf{G}(s)$ such that the corresponding closed-loop system is stable and adheres to the prescribed transient and steady-state performance requirements. The INA methodology is best conceptualised by placing a constant, diagonal matrix, **F**, in the feedback path, as shown in Figure 6.12, where $\mathbf{r}(s)$ and $\mathbf{y}(s)$ represent the reference and output vectors, respectively. Based on Figure 6.12, the closed-loop transfer behaviour from $\mathbf{r}(s)$ to $\mathbf{y}(s)$ is easily determined as

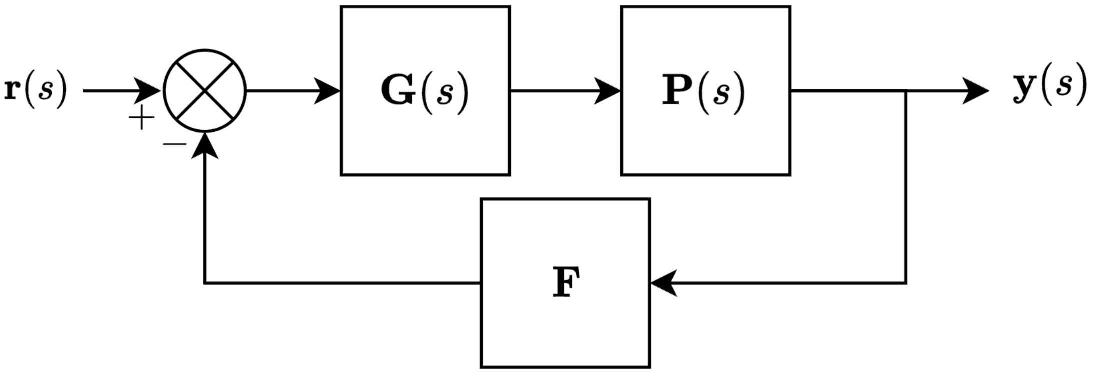


Figure 6.12: Multivariable feedback configuration. All signals are m-dimensional vectors.

$$\mathbf{H}(s) = (\mathbf{I} + \mathbf{Q}(s)\mathbf{F})^{-1}\,\mathbf{Q}(s), \tag{6.52}$$

where $\mathbf{Q}(s) = \mathbf{P}(s)\mathbf{G}(s)$ represents the forward transfer matrix from $\mathbf{r}(s)$ to $\mathbf{y}(s)$. Reflecting on (6.52), the relationship between the open-loop system $\mathbf{Q}(s)$ and the closed-loop system $\mathbf{H}(s)$ is nontrivial. However, inversion of (6.52) results in

$$\mathbf{H}^{-1}(s) = \mathbf{F} + \mathbf{Q}^{-1}(s), \tag{6.53}$$

in which the open-loop and closed-loop plant instances are directly related to each other. It is common to write $\mathbf{H}^{-1}(s) = \hat{\mathbf{H}}(s)$ and $\mathbf{Q}^{-1}(s) = \hat{\mathbf{Q}}(s)$, which then implies that $\hat{h}_{ik}(s)$ is the (i,k) element of $\mathbf{H}^{-1}(s)$. The form in (6.53) implies that when $\mathbf{F} = 0$ the system is in open-loop and $\mathbf{H}^{-1}(s) = \mathbf{Q}^{-1}(s)$. Note also that diagonal **F** means that $\hat{h}_{ik}(s) = \hat{q}_{ik}(s)$ when $i \neq k$. Matrix **F** can be further specified as possessing unit entries along its diagonal, corresponding to those loops that are closed. This allows determination of the diagonal elements of $\mathbf{H}^{-1}(s)$ depending on whether the particular feedback loop is open or closed and is expressed as

$$\hat{h}_{ii}(s) = \begin{cases} \hat{q}_{ii}(s), & f_{ii} = 0, \\ 1 + \hat{q}_{ii}(s), & f_{ii} = 1. \end{cases} \tag{6.54}$$

The inverse Nyquist array comprises m^2 diagrams of $\hat{q}_{ik}(j\omega)$, $\forall i,k$ in the arithmetic-complex plane. An example is shown in Figure 6.55 for the system

$$\mathbf{Q}(s) = \begin{bmatrix} \frac{1}{(1+s)^2} & \frac{0.5}{(1+s)^2} \\ \frac{0.2}{(1+s)^2} & \frac{2}{(1+s)^2} \end{bmatrix}. \tag{6.55}$$

If $\mathbf{Q}(s) = \mathbf{P}(s)\mathbf{G}(s)$ is a diagonal matrix, then this results in m strictly non-interacting loops that can be designed individually. It can be shown that if the gains in all the closed loops are sufficiently high, and only the i^{th} loop is left open, then $\hat{q}_{ii}(s)$ provides a good indication of the inverse response between $r_i(s)$ and $y_i(s)$, namely

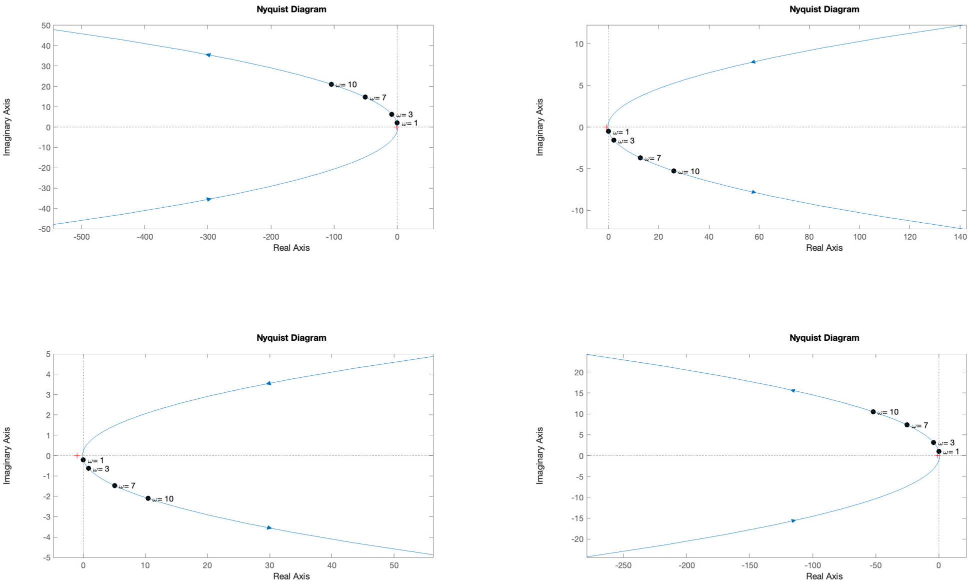


Figure 6.13: Inverse Nyquist diagram of matrix transfer function of $\hat{\mathbf{Q}}(s)$ from (6.55).

$$\hat{q}_{ii}(s) \approx \frac{1}{h_{ii}(s)}. \tag{6.56}$$

The (i, i) feedback controller element can then be designed to shape $\hat{q}_{ii}(s)$ based on interpretation of the inverse Nyquist loci. In practice, arbitrarily large loop gain is not possible at all frequencies and are in fact constrained to be low at key frequencies, such as in the vicinity of the gain-phase crossover region.

Instead of attempting to diagonalise the loop transfer function matrix, the INA method endeavours to enforce the looser criterion of diagonal dominance. Matrix (row) diagonal dominance can be assessed using Gershgorin's theorem [84], which states that $m \times m$ matrix $\hat{\mathbf{Q}}(s)$ is row diagonal dominant if the sum of the moduli of the off-diagonal elements in each row of $\hat{\mathbf{Q}}(s)$ is smaller than the corresponding diagonal element modulus. This is represented mathematically as

$$|\hat{q}_{ii}(s)| > \sum_{k=1, k\neq i}^{m} |\hat{q}_{ik}(s)|\,, \tag{6.57}$$

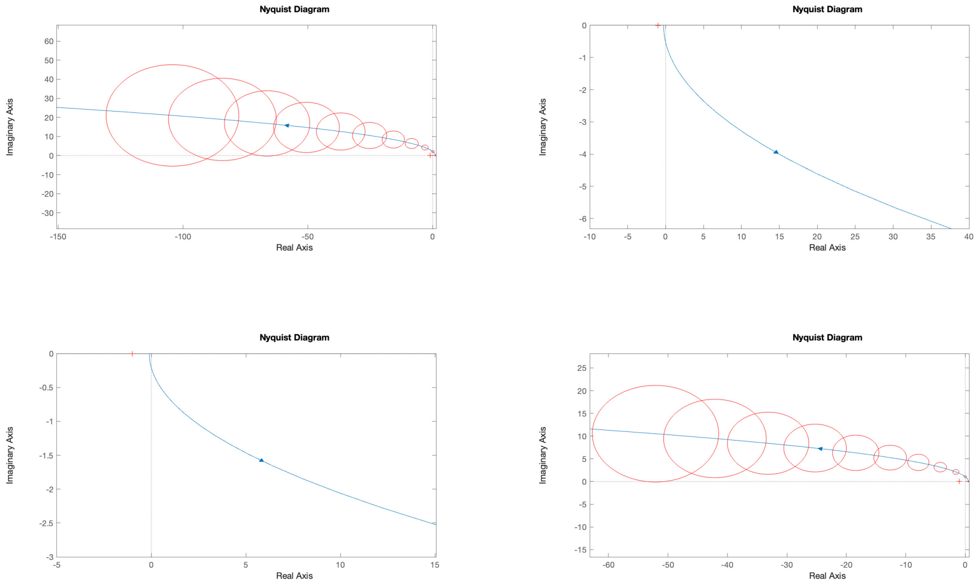


Figure 6.14: Inverse Nyquist diagram of matrix transfer function of $\hat{\mathbf{Q}}(s)$ from (6.55). Gershgorin bands are overlaid on the diagonal loci.

for $i = 1, 2, ...m$ and for all s. Column diagonal dominance is similarly stated as

$$|\hat{q}_{ii}(s)| > \sum_{k=1, k\neq i}^{m} |\hat{q}_{ki}(s)| \,. \tag{6.58}$$

The (row) dominance of a matrix can be visualised in the arithmetic-complex plane by plotting a circle, centred at $\hat{q}_{ii}(j\omega)$, with radius

$$d_i(\omega) = \sum_{k=1, k\neq i}^{m} |\hat{q}_{ik}(j\omega)| \,, \tag{6.59}$$

for $i = 1, 2, ...m$ and for all $\omega \in [0, \infty)$. If the union of the resulting disks excludes the origin for all inverse Nyquist arrays of $\hat{q}_{ii}(j\omega)$, then (6.57) is satisfied and $\hat{\mathbf{Q}}(s)$ is row dominant. Column dominance can be tested in the same manner, but with

$$d_i(\omega) = \sum_{k=1, k\neq i}^{m} |\hat{q}_{ki}(j\omega)| \,. \tag{6.60}$$

The collection of disks for a particular (inverse) Nyquist diagram is commonly referred to as Gershgorin bands, with an example shown in Figure 6.14. The narrower the Gershgorin bands, the more closely $\hat{\mathbf{Q}}(s)$ resembles a diagonal $m \times m$ matrix. Aside from assessing matrix dominance, the union of the characteristic loci is known to be contained within the union of this band, based on Gershgorin's theorem. By extension, Gershgorin bands can also be used to assess closed-loop stability. Namely, if all the Gershgorin bands exclude the critical point, then counting the number of encirclements of the -1 point corresponds with the number of encirclements made by the characteristic loci. However this assessment makes use of sufficient, but not necessary, conditions for stability, meaning that closed-loop stability may still be maintained if the Gershgorin band encompasses the critical point.

A less conservative measure of the error incurred by using the approximation in (6.56) is facilitated using Ostrowski's theorem [85], which can be used to show that

$$\left| \hat{q}_{ii}(j\omega) - \frac{1}{\hat{h}_{ii}(j\omega)} \right| \leq d_i(\omega)\phi_i(\omega), \tag{6.61}$$

where $d_i(\omega)$ is the radius of the i^{th} Gershgorin circle from (6.59) or (6.60), and

$$\phi_i(\omega) = \max_{k \neq i} \frac{\sum_{a \neq k} \left| \hat{h}_{ka}(j\omega) \right|}{\left| \hat{h}_{kk}(j\omega) \right|}. \tag{6.62}$$

Since $\hat{\mathbf{H}}(j\omega)$ is assumed to be diagonal dominant, it follows that $\phi_i(\omega) < 1$. Based on (6.61), we can deduce that $\frac{1}{\hat{h}_{ii}(j\omega)}$ is contained in a circle of centre $\hat{q}_{ii}(j\omega)$ and radius $d_i(\omega)\phi_i(\omega)$. Note that this circle is equivalent to the i^{th} Gershgorin circle after the radius has been scaled by $\phi_i(\omega)$. As $\phi_i(\omega) < 1$, this implies that $\frac{1}{\hat{h}_{ii}(j\omega)}$ will reside in the i^{th} Gershgorin circle after it has been shrunk by a factor of $\phi_i(\omega)$. This shrunken circle is commonly referred to as an Ostrowski circle, and the union of such circles is known as the Ostrowski band. This band is a less conservative measure of the interaction between the closed loops of the system, relative to Gershgorin bands, and more accurately locates $\frac{1}{\hat{h}_{ii}(j\omega)}$ in the inverse Nyquist diagram.

The feedback controller structure proposed in [86] is that of

$$\mathbf{G}(\mathbf{s}) = \mathbf{G}_a \mathbf{G}_b(s) \mathbf{G}_c(s), \tag{6.63}$$

where $\mathbf{G}_a$ is a permutation matrix, $\mathbf{G}_b(s)$ is a product of elementary matrices, and $\mathbf{G}_c(s)$ is a diagonal matrix. $\mathbf{G}_a$ is used to reorder the input-output pairings (if necessary), and $\mathbf{G}_b(s)$ is used to redistribute the "difficulty of control" across all loops [87]. Together, $\mathbf{G}_a\mathbf{G}_b(s)$ attempt to make $\mathbf{Q}(s)$ diagonally dominant using information from the inverse Nyquist diagrams and Gershgorin/Ostrowski bands. If diagonal dominance is achieved, then the system has been decoupled to some extent, and the diagonal elements of $\mathbf{G}_c(s)$ can be designed individually using the inverse Nyquist diagram. As the design takes place on $\hat{\mathbf{Q}}(s) = \hat{\mathbf{G}}(s)\hat{\mathbf{P}}(s)$, the feedback controller inverse is designed directly, with the structure

$$\hat{\mathbf{G}}(s) = \hat{\mathbf{G}}_c(s) \hat{\mathbf{G}}_b(s) \hat{\mathbf{G}}_a, \tag{6.64}$$

which requires that $\hat{\mathbf{G}}_c(s)$, $\hat{\mathbf{G}}_b(s)$, and $\hat{\mathbf{G}}_a$, are stable, minimum-phase, and have a relative degrees of zero or less (in order to allow for matrix inversion).

An alternative approach of pre-compensation is to choose $\hat{\mathbf{G}}_c(s)\hat{\mathbf{G}}_b(s) = \mathbf{P}(j0)$, if $|\mathbf{P}(j0)|$ is non-singular. This is equivalent to diagonalising the loop transfer matrix at zero frequency. However, there is the risk that diagonalising at a particular design frequency will result in stronger, and possibly more problematic coupling at other frequencies. A more systematic approach involves selecting the pre-compensation based on attempting to minimise (6.59) in a least-mean-squares sense for a particular band of frequencies. Work in [88] uses Perron-Frobenius theory to obtain optimal diagonal pre-compensators at the selected frequency, which results in the reduction of the off-diagonal element magnitudes. Other methods include the *cut and try* approach, and pseudo-diagonalisation [89].

The INA approach and multivariable QFT share some common methodological threads. In both cases, elements of the plant matrix inverse form part of the system analysis. The two methods enable the control practitioner to work with in the real-coordinate frame of the problem, which allows the understanding of any special features of the plant to be exploited in the design [84]. Both techniques rely on loopshaping of the open-loop behaviour in order to achieve closed-loop performance. The non-sequential MIMO QFT design approach also makes implicit assumptions about diagonal dominance, albeit through overbounding using the triangle inequality, in order to design all feedback control loops individually. The Ostrowski bands used in INA are also comparable to the equivalent disturbance technique used in MIMO QFT, which can be used to trade design conservatism in order to arrive at m single-input-single-output design equations.

6.2.2 The characteristic-locus method

The multivariable characteristic locus (CL) method was first introduced in [90]. The concept of this method, which was extended in [91], is to manipulate the characteristic loci of each loop, made up of the frequency-dependent eigenvalues of the corresponding open-loop transfer function channel, as if they are standard Nyquist loci.

Assuming the multivariable plant under consideration can be represented by a square, $m \times m$ transfer function matrix, $\mathbf{P}(s)$, a spectral decomposition of the plant is described by

$$\mathbf{P}(s) = \mathbf{W}(s)\mathbf{\Lambda}(s)\mathbf{W}^{-1}(s), \tag{6.65}$$

where $\mathbf{W}(s)$ is a matrix whose columns are the eigenvectors (or characteristic directions) of $\mathbf{P}(s)$, and

$$\mathbf{\Lambda}(s) = \text{diag}\{\lambda_1(s), \lambda_2(s), ...\lambda_m(s)\}, \tag{6.66}$$

where $\lambda_i(s)$ is the i^{th} eigenvalue (or characteristic function) of $\mathbf{P}(s)$. If we select the feedback controller structure as

$$\mathbf{G}(s) = \mathbf{W}(s)\mathbf{M}(s)\mathbf{W}^{-1}(s), \tag{6.67}$$

where

$$\mathbf{M}(s) = \text{diag}\{\mu_1(s), \mu_2(s), ...\mu_m(s)\}, \tag{6.68}$$

then the open-loop transfer function matrix follows as

$$\mathbf{Q}(s) = \mathbf{P}(s)\mathbf{G}(s) = \mathbf{W}(s)\mathbf{N}(s)\mathbf{W}^{-1}(s). \tag{6.69}$$

Matrix $\mathbf{N}(s)$ represents a diagonal matrix whereby the diagonal elements comprise the product of eigenvalues from $\mathbf{P}(s)$ and $\mathbf{G}(s)$, namely

$$\mathbf{N}(s) = \text{diag}\{v_1(s), v_2(s), ...v_m(s)\}, \tag{6.70}$$

where $v_i(s) = \lambda_i(s)\mu_i(s)$. The structure in (6.65) and (6.67) suggest that $\mathbf{P}(s)$ and $\mathbf{G}(s)$ commute, namely

$$\mathbf{P}(s)\mathbf{G}(s) = \mathbf{G}(s)\mathbf{P}(s), \tag{6.71}$$

which gives rise to the term ’commutative’ compensation.

Using the framework above, one can then use well-established single-loop frequency-domain techniques to design "compensator" $\mu_i(s)$ such that $v_i(s)$ has sufficient performance and stability qualities. Once $\mu_i(s)$ has been selected for all $i = 1, ...m$, $\mathbf{G}(s)$ can then be determined using (6.67), where $\mathbf{W}(s)$ is obtained from the eigendecomposition in (6.65).

It can be shown [84] that if the open-loop transfer function has the spectral decomposition as in (6.69), then matrix sensitivity and complementary sensitivity functions can be decomposed as

$$\mathbf{S}(s) = \mathbf{W}(s)\text{diag}\{\frac{1}{1 + v_i(s)}\}\mathbf{W}^{-1}(s), \tag{6.72}$$

$$\mathbf{T}(s) = \mathbf{W}(s)\text{diag}\{\frac{v_i}{1 + v_i(s)}\}\mathbf{W}^{-1}(s), \tag{6.73}$$

respectively. Note that (6.69), (6.72), and (6.73) all share the same eigenvectors. A reasonable assumption would be to to shape $v_i(s)$ based on classical graphical frequency response techniques. However, the eigenvalues of matrix $\mathbf{Q}(s)$ will be bounded in magnitude based on

$$\underline{\sigma}(\mathbf{Q}(s)) \leq |v_i(s)| \leq \overline{\sigma}(\mathbf{Q}(s)), \tag{6.74}$$

where $\underline{\sigma}(\mathbf{Q}(s))$ and $\overline{\sigma}(\mathbf{Q}(s))$ represent the smallest and largest principal gains, respectively. This implies that shaping the performance of $v_i(s)$ is only indicative of the corresponding closed-loop transfer behaviour when the condition number of $\mathbf{Q}(s)$ is close to unity. It can also be shown that if $\mathbf{W}(s)$ is close to orthogonal, then the smallest and largest eigenvalues are close to the smallest and largest singular values. Shaping $v_i(s)$ will therefore result in a transparent design when $\mathbf{Q}(s)$ has low skewness [84].

For realisable purposes, $\mathbf{G}(s)$ in (6.67) must be a stable, causal, rational transfer function matrix. The reliance of obtaining $\mathbf{W}(s)$ from (6.65) is therefore problematic in general, as there is no guarantee that both $\mathbf{W}(s)$ and $\mathbf{W}^{-1}(s)$ will be rational. A practical alternative is to structure the feedback controller matrix as

$$\mathbf{G}(s) = \mathbf{A}(s)\mathbf{M}(s)\mathbf{B}(s), \tag{6.75}$$

where $\mathbf{A}(s)$ and $\mathbf{B}(s)$ are realisable transfer function matrices such that

$$\mathbf{A}(s) \approx \mathbf{W}(s), \tag{6.76}$$

and

$$\mathbf{B}(s) \approx \mathbf{W}^{-1}(s). \tag{6.77}$$

The structure in (6.75) is referred to as approximate commutative compensation. The simplest means of selecting $\mathbf{A}(s)$ and $\mathbf{B}(s)$ is as approximations to $\mathbf{W}(j\omega_0)$ and $\mathbf{W}^{-1}(j\omega_0)$, respectively, at a particular design frequency ω_0. The resulting $\mathbf{A}(j\omega_0)$ and $\mathbf{B}(j\omega_0)$ will be a complex-valued matrix in general, which can then be approximated by real matrices for sake of implementation. One such approach of approximating complex matrices as real-valued matrices is the ALIGN algorithm [92]. This simple approach tends to work well when all the characteristic loci require shaping over approximately the same frequency band. However, in general, the characteristic loci will require shaping at different frequencies, which can be resolved using cascaded approximate commutative compensators of the form

$$\mathbf{G}(s) = \prod_{i=1}^{m} \mathbf{G}_i(s) \tag{6.78}$$

where

$$\mathbf{G}_i(s) = \mathbf{A}_i\mathbf{M}_i(s)\mathbf{B}_i. \tag{6.79}$$

$\mathbf{G}_i(s)$ would then be designed to provide compensation at design frequency ω_i. This naturally results in a high-order compensator, depending on the number of design frequencies, and also may suffer from the fact that each new design stage may interfere with the previous design(s), thereby undoing some of the loopshaping.

The characteristic-locus method is based on the cascaded commutative structure in (6.78) and prescribes a sequential, three-stage design procedure of the form

$$\mathbf{G}(s) = \mathbf{G}_h\mathbf{G}_m(s)\mathbf{G}_l(s), \tag{6.80}$$

where $\mathbf{G}_h$, $\mathbf{G}_m(s)$, and $\mathbf{G}_l(s)$ correspond to compensation at the high, medium, and low frequencies, respectively. The high-frequency gain matrix, $\mathbf{G}_h$, is synthesized first, with the intention of decoupling open-loop transfer function matrix at (or near) the desired closed-loop bandwidth, ω_b. This is achieved by setting

$$\mathbf{G}_h \approx -\mathbf{P}^{-1}(j\omega_b), \tag{6.81}$$

which can be approximated using the ALIGN algorithm. The benefit of this first step is that the resulting characteristic loci are forced to have similar values near the frequency ω_b, and this makes the subsequent design of $\mathbf{G}_m(s)$ easier. The high-frequency compensated plant, $\mathbf{P}(s)\mathbf{G}_h$, is then compensated for within the medium frequencies of $\omega_m < \omega_b$. This stage of the design is concerned with stability margins and often requires lead or lag compensation, which necessitates the approximate commutative structure from (6.79) in order to shape the characteristic loci that are too close to the critical point. This is achieved using the structure

$$\mathbf{G}_m(s) = \mathbf{A}_m\mathbf{M}_m(s)\mathbf{B}_m, \tag{6.82}$$

where

$$\mathbf{A}_m \approx \mathbf{W}(j\omega_m), \tag{6.83}$$

$$\mathbf{B}_m \approx \mathbf{W}^{-1}(j\omega_m), \tag{6.84}$$

and

$$\mathbf{P}(s)\mathbf{G}_h = \mathbf{W}(s)\mathbf{\Lambda}(s)\mathbf{W}^{-1}(s). \tag{6.85}$$

$\mathbf{A}_m$ and $\mathbf{B}_m$ are implemented as real-valued matrices using the ALIGN algorithm. $\mathbf{G}_m(s)$ is ideally selected such that

$$\lim_{\omega\to\infty} \mathbf{G}_m(j\omega) \to \mathbf{I}, \tag{6.86}$$

as this will reduce the interference of $\mathbf{G}_m(s)$ with the high-frequency decoupling of $\mathbf{G}_h$. A more rigorous requirement would be

$$\lim_{\omega\to\omega_b} \mathbf{G}_m(j\omega) \to \mathbf{I}, \tag{6.87}$$

but this is not realistic in general. Once satisfactory stability margins have been achieved, the low-frequency compensation can proceed, with the primary intention of ensuring the required steady-state error tolerances. This is conducted by designing approximate commutative controller $G_l(s)$ at some frequency $\omega_l < \omega_m$, for the compensated plant $\mathbf{P}(s)\mathbf{G}_h\mathbf{G}_m(s)$. $G_l(s)$ will commonly employ integral control, whilst ensuring that

$$\lim_{\omega\to\infty} \mathbf{G}_l(j\omega) \to \mathbf{I}, \tag{6.88}$$

in order to reduce the interference with $\mathbf{G}_m(s)$. This commonly takes the form of a proportional-plus-integral controller matrix. The resulting feedback controller matrix in (6.80) is often of high order, owing to the fact that the final controller solution is made up of up to seven matrix products. One such approach of implementing the compensator is by converting the matrix structure into an equivalent state space realisation [84]. While this does not solve the problem of a high-order controller, the rigour required to implement the control scheme is reduced.

The design order detailed above is but one approach to the characteristic-locus design method. For example, one could instead design from low frequencies to high frequencies without incurring much difference to the result. Permutation matrices, if motivated, are also useful tools of pre-compensating the open-loop matrix transfer function prior to engaging in a detailed design. The work in [90] provides some detail on this, as well as a few other archetypal approaches that will inevitably be dependant on the particular problem at hand.

The characteristic-locus design method is most effective when approximately equivalent bandwidths can be expected of the plant in closed-loop form. However, if there is a disparity between the response time of multiple loops, then the design procedure may need to be reframed such that the loops are closed sequentially, starting from the highest bandwidth loop and then working down in a manner similar to the approach detailed above.

As with INA and QFT, the CL design methodology relies on loopshaping, albeit of characteristic loci and not an open-loop frequency response. While the Nyquist diagram is the common means to visualise and design the commutative compensators, one could also make use of the Nichols chart, as is common in QFT circles, when shaping the loci. However, a common issue, whether using a Nichols chart or Nyquist diagram with closed loop loci is that the visualization presents the behaviour of the nominal, diagonal loop, not the achieved performance with coupling and uncertainty. One weakness of the CL method is the lack of transparency of loop interaction when shaping the loci in the Nyquist diagram, as a result of applying the eigendecomposition (although this information can still be assessed in some sense by considering the orthogonality of the eigenvector matrix). As previously stated, when the skew of the eigenvector matrices increase, the under- and overbounding from 6.74 becomes more conservative, and the discrepancy between perceived and actual performance increases. There is, however, no means of visualising the extent of this conservatism on a Nyquist diagram when using the CL approach.

A secondary weakness that is shared by both INA and CL design is the assumption that diagonalisation, via approximate inverse of the plant matrix at some frequency, is effective under plant uncertainty. This is comparable to some QFT methods that rely on nominal plant selection as part of the design procedure. In all cases, selection of a nominal plant that effectively decouples the design at a particular frequency under uncertainty is difficult to generalise. It is for this reason that controller solutions that rely on some nominal plant are, at best-case, sup-optimal, and worst-case unable to resolve the design problem. By extension, the assumption that the SISO equivalent designs will lead to the resulting multivariable system meeting the original performance criteria is not guaranteed [93]. Open-loop diagonalisation via pre-compensation is a key feature of both INA and the CL method, but these approaches also extend to other frequency-domain design techniques, such as multivariable QFT, with [94] as one example. While QFT does not explicitly rely on diagonal dominance (for example via pre-compensation), enforcing this to some extent naturally reduces channel interaction, thereby alleviating the design conservatism that is induced when applying the triangle inequality. That being said, with sufficiently high channel interaction, sensitivity and performance specifications arising from the QFT design paradigm will become incompatible, and no feedback control solution will appear to exist.

6.2.3 H-Infinity synthesis

The field of H-infinity (H_∞) is concerned with formulating optimisation problems that can be solved iteratively to synthesize a suitable stabilising control solution that provides a quantified level of robustness. The roots of H_∞ can be traced back to the seminal work in [95], which considered the minimisation of the $\infty-$norm of the sensitivity function of a linear single-input-single-output system. This novel approach quickly received a large amount of interest from the control theory community and these concepts were later generalised to multi-input-multi-output mixed sensitivity problems [96]. H_∞ is now a mainstream control technique that can be used to solve robust control design problems using computer-based toolboxes [97].

Consider the standard MIMO, single-degree-of-freedom control system in Figure 6.15, where $\mathbf{P}(s)$ and $\mathbf{G}(s)$ are matrix transfer functions that represent the plant and feedback controller respectively. The exogenous signals comprise of the reference signal, $\mathbf{r}(s)$, disturbance (reflected at the plant output for sake of convenience), $\mathbf{d}(s)$, and the measurement noise, $\mathbf{n}(s)$. The controlled variable is $\mathbf{y}(s)$, with the objective of designing $\mathbf{G}(s)$ such that $\mathbf{y}(s)$ approaches $\mathbf{r}(s)$ with the prescribed performance. This often involves consideration of the sensitivity function,

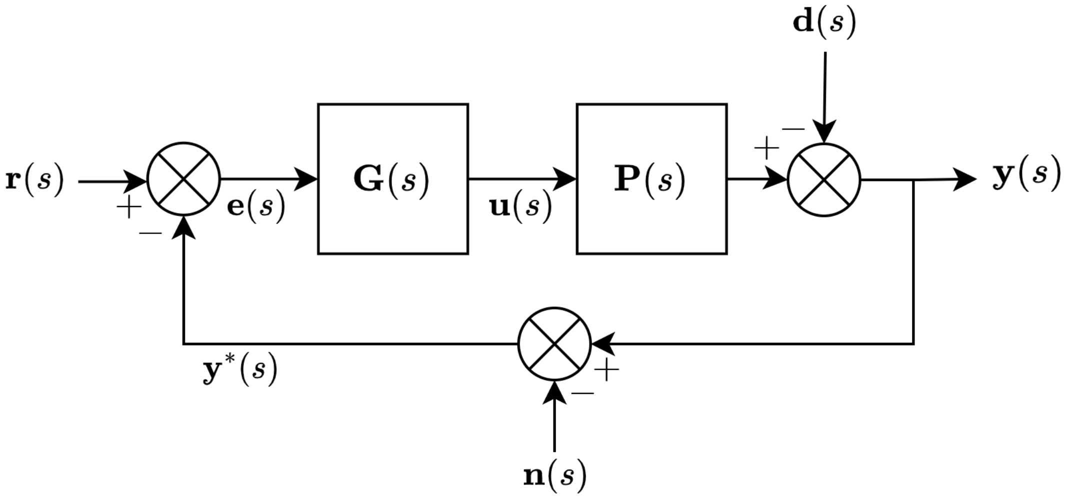


Figure 6.15: Standard MIMO, single-degree-of-freedom control system.

$$\mathbf{S}(s) = (\mathbf{I} + \mathbf{P}(s)\mathbf{G}(s))^{-1}, \tag{6.89}$$

and input sensitivity function

$$\mathbf{GS}(s) = \mathbf{G}(s)\left(\mathbf{I} + \mathbf{P}(s)\mathbf{G}(s)\right)^{-1}. \tag{6.90}$$

Note that $\mathbf{d}(s)$, $\mathbf{n}(s)$, and $\mathbf{r}(s)$ are related to $\mathbf{e}(s)$ and $\mathbf{u}(s)$ by $\mathbf{S}(s)$ and $\mathbf{GS}(s)$, respectively. However, the external signals generally have signal power at different frequency bands. For example, $\mathbf{d}(s)$ and $\mathbf{r}(s)$ are usually low-frequency signals within the loop bandwidth, whereas $\mathbf{n}(s)$ commonly represents high-frequency sensor noise.

Practical control design generally requires that the reference tracking error, $\mathbf{e}(s)$, remain suitably small over a defined frequency range of interest, whilst also minimising the control effort, $\mathbf{u}(s)$. In other words, $|\mathbf{S}(j\omega)|$ and $|\mathbf{GS}(j\omega)|$ should be bounded in some sensible way. The H_∞ framework leverages this information to repose the control problem as a standard H_∞ configuration, seen in Figure 6.16, where

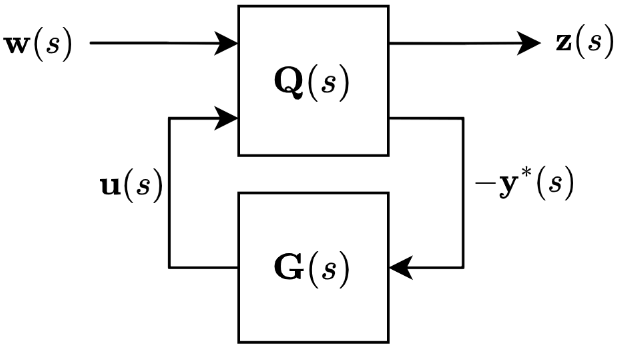


Figure 6.16: Standard H_∞ configuration.

$$\mathbf{Q}(s) = \begin{bmatrix} \mathbf{Q}_{11}(s) & \mathbf{Q}_{12}(s) \\ \mathbf{Q}_{21}(s) & \mathbf{Q}_{22}(s) \end{bmatrix} \tag{6.91}$$

is referred to as the generalised plant,

$$\mathbf{z}(s) = \begin{bmatrix} \mathbf{z}_1(s) \\ \mathbf{z}_2(s) \end{bmatrix} = \begin{bmatrix} \mathbf{e}(s) \\ \mathbf{u}(s) \end{bmatrix} \tag{6.92}$$

represents the output vector of $\mathbf{Q}(s)$, and

$$\mathbf{w}(s) = \begin{bmatrix} \mathbf{r}(s) \\ \mathbf{d}(s) \\ \mathbf{n}(s) \end{bmatrix} \tag{6.93}$$

comprise all external signals acting on the generalised plant. Details of the two-degree-of-freedom reference tracking formulation can be found in [98]. For the standard H_∞-optimal regulator problem, $\mathbf{r}(s) = 0$, and the vector of exogenous signals reduces to

$$\mathbf{w}(s) = \begin{bmatrix} \mathbf{d}(s) \\ \mathbf{n}(s) \end{bmatrix}. \tag{6.94}$$

With reference to Figure 6.15-6.16, without external reference, the open-loop behaviour of $\mathbf{Q}(s)$ is described by

$$\begin{bmatrix} \mathbf{z}(s) \\ -\mathbf{y}^*(s) \end{bmatrix} = \mathbf{Q}(s) \begin{bmatrix} \mathbf{w}(s) \\ \mathbf{u}(s) \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{I} & -\mathbf{P}(s) \\ \mathbf{O} & \mathbf{I} \\ \hline \mathbf{I} & \mathbf{P}(s) \end{array}\right] \begin{bmatrix} \mathbf{w}(s) \\ \mathbf{u}(s) \end{bmatrix}, \tag{6.95}$$

which implies that

$$\mathbf{Q}(s) = \begin{bmatrix} \mathbf{Q}_{11}(s) & \mathbf{Q}_{12}(s) \\ \mathbf{Q}_{21}(s) & \mathbf{Q}_{22}(s) \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{I} & -\mathbf{P}(s) \\ \mathbf{O} & \mathbf{I} \\ \hline \mathbf{I} & \mathbf{P}(s) \end{array}\right]. \tag{6.96}$$

The objective is to find a stabilising controller, $\mathbf{G}(s)$, to minimise $\mathbf{z}(s)$ in an energy sense. More specifically, the worst-case gain from $\mathbf{w}$ to $\mathbf{z}$ is to be minimised, with the matrix transfer function

$$\mathbf{z}(s) = \mathbf{H}(\mathbf{s})\mathbf{w}(s) \tag{6.97}$$

representing this closed-loop relationship. Quantifying this worst-case gain is performed using induced norms. If $\mathbf{w}$ and $\mathbf{z}$ are norm-bounded using the Euclidean norm

$$\begin{aligned} ||\mathbf{w}||_2 &= \sqrt{\int_{-\infty}^{\infty} \mathbf{w}^{\mathrm{H}}(t)\mathbf{w}(t)dt}, \\ ||\mathbf{z}||_2 &= \sqrt{\int_{-\infty}^{\infty} \mathbf{z}^{\mathrm{H}}(t)\mathbf{z}(t)dt}, \end{aligned} \tag{6.98}$$

then it can be shown that the norm of $\mathbf{H}$ induced by the signal norms of $\mathbf{w}$ and $\mathbf{z}$ is given by

$$||\mathbf{H}||_\infty = \sup_{\omega \in \mathbb{R}} ||\mathbf{H}(j\omega)||_2. \tag{6.99}$$

For constant, complex-valued matrix $\mathbf{H}(j\omega)$, $||\mathbf{H}(j\omega)||_2$ corresponds to the spectral norm, namely,

$$||\mathbf{H}(j\omega)||_2 = \max_{\forall i} \sigma_i \left(\mathbf{H}(j\omega)\right), \tag{6.100}$$

where σ_i is the i^{th} singular value of $\mathbf{H}(j\omega)$. $||\mathbf{H}||_\infty$ is therefore found by first calculating the largest singular value of $\mathbf{H}(j\omega)$ for each frequency, and then determining the largest result across the entire frequency range.

Using the developments above, the H_∞ optimisation problem is framed as

$$\min_{\mathbf{G}(s)} ||\mathbf{H}||_\infty, \tag{6.101}$$

where $\mathbf{G}(s)$ is required to be stable based on the Youla parameterisation [84]. Based on the input-output relationship of $\mathbf{H}(s)$, (6.101) can be written as

$$\min_{\mathbf{G}(s)} \left\| \begin{matrix} (\mathbf{I}+\mathbf{PG})^{-1} \\ \mathbf{G}\,(\mathbf{I}+\mathbf{PG})^{-1} \end{matrix} \right\|_{\infty}, \tag{6.102}$$

which is commonly referred to as mixed sensitivity optimisation (owing to the fact the sensitivity and load sensitivity are both considered in the cost function). Using (6.95) and (6.96), (6.97) can be represented by

$$\begin{aligned} \mathbf{z}(s) &= \left[\mathbf{Q}_{11}(s) + \mathbf{Q}_{12}(s)\mathbf{G}(s)\,(\mathbf{I} - \mathbf{Q}_{22}(s)\mathbf{G}(s))^{-1}\,\mathbf{Q}_{21}(s)\right] \mathbf{w}(s), \\ &= \mathcal{F}_l(\mathbf{Q},\mathbf{G})\mathbf{w}(s), \end{aligned} \tag{6.103}$$

where $\mathcal{F}_l(\mathbf{Q},\mathbf{G})$ is a lower linear fractional transform of $\mathbf{Q}(\mathbf{s})$ and $\mathbf{G}(\mathbf{s})$, and can be used as the argument to the optimisation formulation in (6.102), namely

$$\min_{\mathbf{G}(s)} \|\mathcal{F}_l(\mathbf{Q},\mathbf{G})\|_{\infty}, \tag{6.104}$$

The cost function posed in (6.104) considers the largest singular values over the sensitivity and load sensitivity functions, but the nature of these functions is that we are concerned with limiting their gain at particular frequencies, specifically where the exogenous signals are expected to act. For example, the sensitivity function is known to approach identity at high frequencies, but we are interested in minimising the sensitivity in the low- to mid-frequency band where disturbances are expected to act (and be most problematic). For this reason, the H_{∞} methodology makes use of weighting functions that prioritise the frequencies of concern, which redefines the output equation from (6.92) as

$$\mathbf{z}(s) = \begin{bmatrix} \mathbf{z}_1(s) \\ \mathbf{z}_2(s) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1(s)\mathbf{e}(s) \\ \mathbf{W}_2(s)\mathbf{u}(s), \end{bmatrix} \tag{6.105}$$

as shown in Figure 6.17. $\mathbf{W}_1(s)$ and $\mathbf{W}_2(s)$ are selected to prioritise particular frequencies for both the sensitivity and load sensitivity behaviour of the closed-loop system. The optimisation formulation from (6.102) follows as

$$\min_{\mathbf{G}(s)} \left\| \begin{matrix} \mathbf{W}_1\,(\mathbf{I}+\mathbf{PG})^{-1} \\ \mathbf{W}_2\mathbf{G}\,(\mathbf{I}+\mathbf{PG})^{-1} \end{matrix} \right\|_{\infty}, \tag{6.106}$$

with $\mathbf{Q}(s)$ from (6.96) similarly updated as

$$\mathbf{Q}(s) = \begin{bmatrix} \mathbf{Q}_{11}(s) & \mathbf{Q}_{12}(s) \\ \mathbf{Q}_{21}(s) & \mathbf{Q}_{22}(s) \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{W}_1(s) & -\mathbf{W}_1(s)\mathbf{P}(s) \\ \mathbf{O} & \mathbf{W}_2(s) \\ \hline \mathbf{I} & \mathbf{P}(s) \end{array}\right]. \tag{6.107}$$

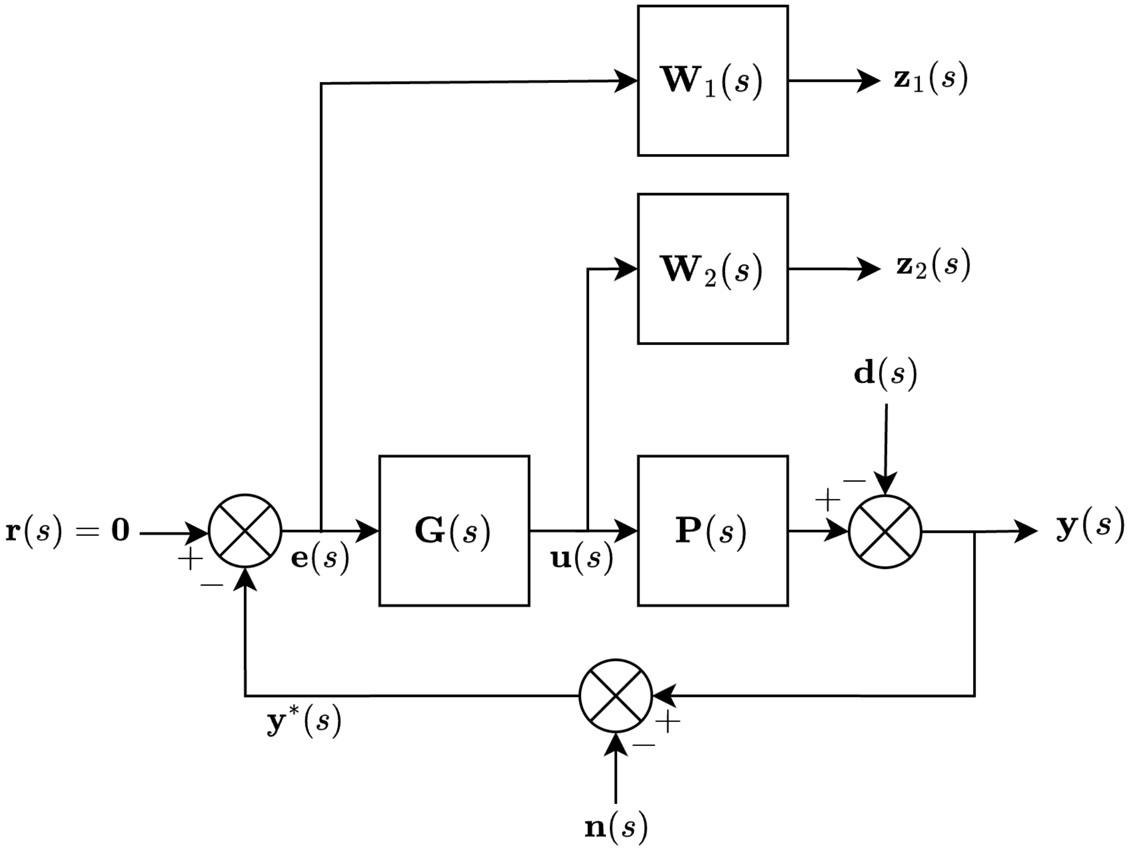


Figure 6.17: Standard MIMO, single-degree-of-freedom regulator control system. Output signals from $\mathbf{H}(s)$ with weighting functions are also included.

In general, the solution to (6.104) is not unique. In practice, the optimisation problem is recast such that the H_∞-norm of the closed-loop transfer function is less than a sufficiently large positive number,

$$\|\mathcal{F}_l(\mathbf{Q}, \mathbf{G})\|_\infty < \gamma. \tag{6.108}$$

Under the correct conditions, a stable $\mathbf{G}(s)$ can be determined from (6.108) using multiple techniques, such as J-spectral factorisation [99], Riccati-based approaches using certain assumptions, and reformulating Riccati equations using linear matrix inequalities [100]. If a stable $\mathbf{G}(s)$ exists that satisfies (6.108), one can then successively reduce γ and generate a corresponding feedback control solution that results in a smaller closed-loop H_∞-norm. The so-called optimal solution is determined when γ cannot be made any smaller without resulting in an infeasible $\mathbf{G}(s)$. The optimal solution that results from (6.108) is fundamentally limited by the stability of the closed-loop system.

The H$_\infty$ methodology quantifies plant parameter variations in the form of unstructured uncertainty. There are multiple ways in which to capture this uncertainty, but a common approach is that of output multiplicative uncertainty of the form

$$\mathbf{P}^*(s) = [\mathbf{I} + \mathbf{\Delta}(s)]\,\mathbf{P}(s), \tag{6.109}$$

as shown in Figure 6.18, where $\mathbf{\Delta}(s)$ is a stable matrix transfer function that represents the relative uncertainty of the plant. By the small-gain theorem [101], a sufficient condition for the closed-loop system in Figure 6.18 to be stable is

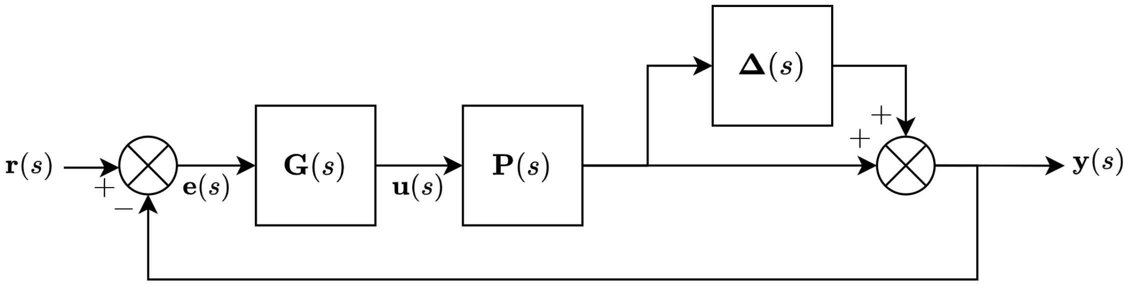


Figure 6.18: Standard MIMO, single-degree-of-freedom control system. Output multiplicate uncertainty is also shown. Disturbance and noise signals are omitted for sake of clarity.

$$||\mathbf{H}\mathbf{\Delta}||_\infty < 1. \tag{6.110}$$

This is guaranteed if

$$||\mathbf{H}\mathbf{\Delta}||_\infty \leq ||\mathbf{H}||_\infty ||\mathbf{\Delta}||_\infty < 1, \tag{6.111}$$

which implies that the solution in (6.108) is closed-loop stabilising if

$$||\mathbf{\Delta}||_\infty \leq \frac{1}{\gamma}. \tag{6.112}$$

Note that this is a sufficient but not necessary condition.

The H$_\infty$ methodology offers a powerful mathematical framework that quantifies the level of robustness using induced norms. Synthesis of an H$_\infty$ controller has no reliance on the open-loop transfer function being diagonal dominant or sufficiently decoupled, which sets it apart from the likes of the inverse Nyquist array and characteristic-locus methods. H$_\infty$ control solutions also implicitly make use of all elements of feedback controller $\mathbf{G}(s)$, which assists with the plant decoupling and control allocation — something that can be very complicated and arduous to design when using loop-by-loop design methods such as QFT.

The use of induced norms entails accepting some design conservatism as part of the inherent overbounding of the actual, unstructured plant uncertainty by one of the structured representations that are available (input, output, multiplicative, etc). H_∞ also requires selection of a nominal plant as part of the design procedure. This nominal should ideally be located in the centre of each frequency-dependent uncertainty template, based on the method's use of discoidal uncertainty bounds. Additionally, the plant templates for particular frequencies will not necessarily have disc-like shapes, which means that additional conservatism will be imposed when growing the uncertainty boundaries appropriately to encompass the known uncertainty. Moreover, the methodology necessitates unstructured uncertainty, which means that phase information from (potentially known) structured uncertainty must be abandoned when formulating the H_∞ routine [102].

6.3 Notation used in journal publications

There are minor differences between the notation used in Section 6.4-6.6. The differences are summarised below.

- Section 6.4 makes use of subscripts to predominantly indicate plant enumeration. This is common in QFT literature when considering SISO problem formulations. Section 6.5-6.6 are concerned with MIMO systems and use subscripts to indicate matrix element rows and columns, with superscripts used to show plant enumeration.
- Section 6.5-6.6 use capitalised variables to indicate matrices and the corresponding lower case variables to show specific elements of the matrix. Section 6.4 only considers SISO systems and instead uses capitalised variables to indicate transfer functions and frequency response quantities.

Section 6.4, 6.5, and 6.6 each contain a Problem Statement section that provides an implicit explanation of the notation used.

6.4 Robust plant by plant control design using model-error tracking sets

6.4.1 Abstract

This Section presents a method for designing a robust two-degree-of-freedom (2DOF) control scheme, capable of satisfying multiple model-error specifications on a plant by plant basis. Traditional quantitative feedback theory (QFT) methods generally use a single model-error or above-below magnitude tracking specification, which can result in over-design for plants located away from the bounding conditions. The performance specifications are also generally hand-tuned, or iteratively adjusted to keep the underlying time-domain signals within permissible levels. Our method aims to perform a model-error design on a per-plant basis, such that each plant's corresponding model tracking has equal weighting given the plant's inherent feedback requirements and capability. The QFT method allows this per-plant approach to be undertaken with ease. Additionally, sufficiently low-order model specifications are designed using simple optimisation, which take into account performance limiting effects, such as non-minimum-phase (NMP) behaviour and signal constraints. A worked example is presented, showing the viability and transparency of the proposed method.

6.4.2 Introduction

Linear robust feedback control problems are generally cast in the frequency-domain, and take the form of 2DOF controllers [12]. The bandwidth trade-off between the two controller elements, conventionally being the pre-filter and feedback controller, as shown in Figure 6.19, is not however obvious. Most commonly employed among QFT practitioners, is the method of magnitude-based tracking bounds [11, 78, 79]. In this form, one can specify per-frequency above and below magnitude bounds on the closed-loop system. This method inherently facilitates the trade-off between the pre-filter and feedback controller. There is however no explicit specification on the closed-loop phase requirement, which can lead to over/under-design of the controller as well as degradation of the tracking precision [13]. This can be remedied by adding above and below phase constraints to the closed-loop system [14]. This will generate a second set of per-frequency bounds, whereby the global solution space is the intersection of the magnitude and phase-derived bounds. Model-error reference signal tracking specifications, also known as model matching, use the magnitude of the closed-loop tracking error, thereby including phase information in the solution space [15]. In this context, one method of resolving the 2DOF trade-off is to fix the pre-filter based on the model and closed-loop nominal plant inverse, prior to the feedback design. Depending on the extent and variety of the plant uncertainty, the choice of nominal plant may introduce conservatism into the bound generation of the feedback controller [103]. Additionally, the order of this pre-filter can be impractically high. The author in [18] solves this by re-designing the pre-filter, using closed-loop tracking error specifications, after the feedback controller has been fixed. More recent work done in [103] and [4] outlines two methods of designing the pre-filter and feedback controller in a model-error form, with no dependence of the nominal plant choice. The authors of [4] develop an inequality solely dependent on the feedback control element which, if satisfied, guarantees the existence of a non-empty solution space to the pre-filter, based on the standard model-error tracking inequality. This effectively decouples the design of the feedback controller and pre-filter, allowing them to be designed sequentially, without any conservatism imposed on the generated bounds. The aforementioned methods only use a single tracking specification for all plant conditions, meaning that a valid solution unavoidably leads to over/under-design of certain plants which do not sit close to the boundary conditions.

In practice, proper control design requires realistic performance specifications, which reflects the plants limitations, be it non-minimum-phase (NMP) behaviour or input constraints. These specifications are generally treated as "client given", or "tuned" until the specifications are achievable. As the specifications do not directly take time-domain signal limits into account, finding a valid solution may yield unacceptable plant signals (for example input-rate saturations), which can degrade performance, and potentially lead to instability. Work done in [104] develops gain-phase constraints on the open-loop system, to ensure that the closed-loop plant input of a 1DOF system does not saturate in amplitude, when being excited by a step input of constrained magnitude. The authors of [105] and [16] avoid signal saturations by embedding a controlled artificial saturation block in front of the actual saturation block. A cascaded internal feedback structure is then designed such that when the artificial element saturates, the system is still closed-loop stable. This has the effect of non-linearly reducing the closed-loop bandwidth as the signal magnitude increases. Work done in [17] chooses above-below tracking bounds to ensure that the worst-case plant (in terms of gain and phase) cannot saturate within the valid design space. Although this does constrain the plant input signal, it imposes conservatism over the entire plant set, especially high gain cases. Other research on the input saturation problem exist, with respect to the QFT design framework [106].

The work presented in this Section resolves the aforementioned design shortfalls. First, a sufficiently low-order model set, which realistically reflects the associated plant set capabilities, is found via simple optimisation. A per-plant model-error inequality is then developed, based on [4], which is used to find the local solution space of the feedback controller. A global controller solution space (across all plants) is then generated by finding the intersection across all local solution spaces. An appropriate feedback controller can then be designed to lie in each discrete design frequency-based solution space. Once the feedback controller has been fixed, the pre-filter is designed by applying the original model-error specifications to the closed-loop transfer behaviour. Both elements are designed using a level-set of error radii, which gives transparency in the discrete frequency to frequency performance trade-offs. The effectiveness of this method is demonstrated with an example in Section 6.4.4.

6.4.3 Proposed method

Problem statement

Given a known, ordered plant set $P_i \in \{\mathbf{P}\}$, $i = \{1, 2, ..., m\}$, a corresponding ordered set of closed-loop tracking models $M_i \in \{\mathbf{M}\}$ can be specified/designed based on the plant capabilities and signal levels. Plant

P_i may be unstable and/or NMP, and may have time-domain input constraints, such as input and input-rate limits, $|u| \leq \overline{u}$ and $|\dot{u}| \leq \overline{\dot{u}}$. After synthesis of a feasible and achievable model set, the design task is to find control elements G and F, shown in Figure 6.19, such that the reference to output transfer behaviour $Y_i/R = T_i F$ suitably approaches M_i for all plant cases, where

$$T_i = \frac{P_i G}{1 + P_i G} \tag{6.113}$$

is the complementary-sensitivity function for closed-loop plant i, where $T_i \in \{\mathbf{T}\}$, $i = \{1, 2, ..., m\}$. The control elements should be sufficiently low order, for sake of implementation. Because time- and frequency-domain signals are only related via the integral over all frequencies (Fourier transform), the resulting closed-loop system may still violate the prescribed time-domain input constraints [107].

Reference model design

The reference model design is not the focus of this work and will only be touched upon briefly here. For sake of simplicity, we use an all-pole reference model structure, with plant dependent dead-time. The order of reference model M_i is chosen to be two greater than that of the corresponding P_i. This is to ensure that the operation $sU_i/R = sM_i/P_i$ is strictly proper (for plant input-rate consideration). Any NMP elements in the specific plant are included in the corresponding reference model, in order to ensure that $U_i/R = M_i/P_i$ is a stable transfer function. For example, if P_1 is an all-pole plant of order m, and has a dead-time of τ seconds, the corresponding reference model structure will be

$$M_1 = \frac{1}{1 + d_1 s + d_2 s^2 + ... + d_{m+2} s^{m+2}} e^{-s\tau}, \tag{6.114}$$

where the coefficients, d_p, in (6.114) are the free parameters. An expected (potentially worst case) reference signal $R(s) = \mathcal{L}\{r(t)\}$, is used to generate the model response $Y_{m_i} = M_i R$, which is then converted into the corresponding time-domain signal y_{m_i}, using the inverse Laplace transform. Additionally, the time-domain reference model input, u_{m_i}, and input-rate, $\dot{u}_{m_i}$, can be found by taking the inverse Laplace transform of $(M_i/P_i)R$ and $(sM_i/P_i)R$ respectively. A simple cost function is then posed which is a function of the time-domain reference tracking error, $e_i = r - y_{m_i}$. The time-domain constraints used are that of the plant input's limitations, namely $|u_i| \leq \overline{u}$ and $|\dot{u}_i| \leq \overline{\dot{u}}$. The aforementioned information is placed in a constrained optimisation routine (*fmincon*[108]). This process is repeated for each plant in the plant set, and this generates a set of corresponding reference models. The resulting reference model set allows high performance tracking whilst also being achievable (in the context of the plant input capabilities).

Controller synthesis

Consider the standard 2DOF control structure shown in Figure 6.19, where F is the pre-filter, G is the feedback controller, $\mathbf{P}$ is the plant set, and H is the transfer function describing the sensor dynamics. For sake of

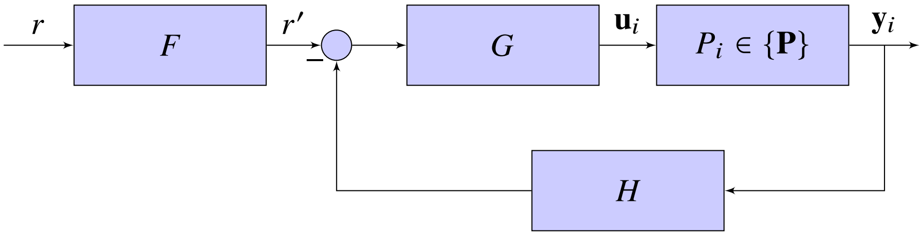


Figure 6.19: Block diagram of standard control scheme with two degrees of freedom.

simplicity, we will assume there is no significant measurement dynamics within the frequency range of interest ($H \approx 1$). If there were just two plants in the plant set, the control elements G and F could be solved for at equality ($F = M_1/T_1 = M_2/T_2$) to satisfy the model specifications exactly. For example, given $P_i \in \{P_1, P_2\}$ and $M_i \in \{M_1, M_2\}$, the control elements can be solved simultaneously as

$$F = \frac{M_1 M_2 (P_1 - P_2)}{M_1 P_2 - M_2 P_1}, \tag{6.115}$$

$$G = \frac{M_2 P_1 - M_1 P_2}{P_1 P_2 (M_1 - M_2)}. \tag{6.116}$$

Note that if $M_1 = M_2$ and $P_1 \neq P_2$, G requires infinite gain to achieve the specifications with zero error, which is an obvious result for single model-error tracking problems. Also, if $M_1 \neq M_2$ and $P_1 = P_2$, the specifications cannot be met for a single reference signal, as a single plant cannot have two different closed-loop transfer behaviours for a fixed control scheme.

When there are more than two plant-model specifications, each specification cannot be exactly met in general, owing to the fact that there are only two degrees of freedom in the control scheme. Instead, one would have to make a compromise across all plant-model combinations. This trade-off can be facilitated by using the method of model-error tracking specifications on a plant by plant basis. Similar to the work in [15], the model-error tracking inequality for plant i is

$$|M_i - T_i F|\Big|_{\omega_d} = \left|M_i - \frac{P_i G}{1 + P_i G} F\right|\Bigg|_{\omega_d} \leq E\Big|_{\omega_d} = E(\omega_d), \tag{6.117}$$

where $E(\omega_d)$ is a user-defined model-tracking error radius (centred at $M_i(j\omega_d)$) and $\omega_d \in \{\mathbf{\Omega}\}$, $d \in \mathbb{Z}^+$. A valid solution at ω_d will therefore place $T_i(j\omega_d)F(j\omega_d)$ inside the corresponding error circle, as shown in Figure 6.20. With a sufficiently rich and appropriate set of discrete design frequency points, a continuous controller can then be designed, which satisfies the inequality across the discrete design space. The underlying assumption here is that the set of design frequency points, $\{\mathbf{\Omega}\}$, sufficiently covers the bandwidth of interest, in terms of the reference model set to be tracked. Note, the fundamental departure from the work in [4] is the existence of a bespoke reference model specification for each plant case, as opposed to a single reference model

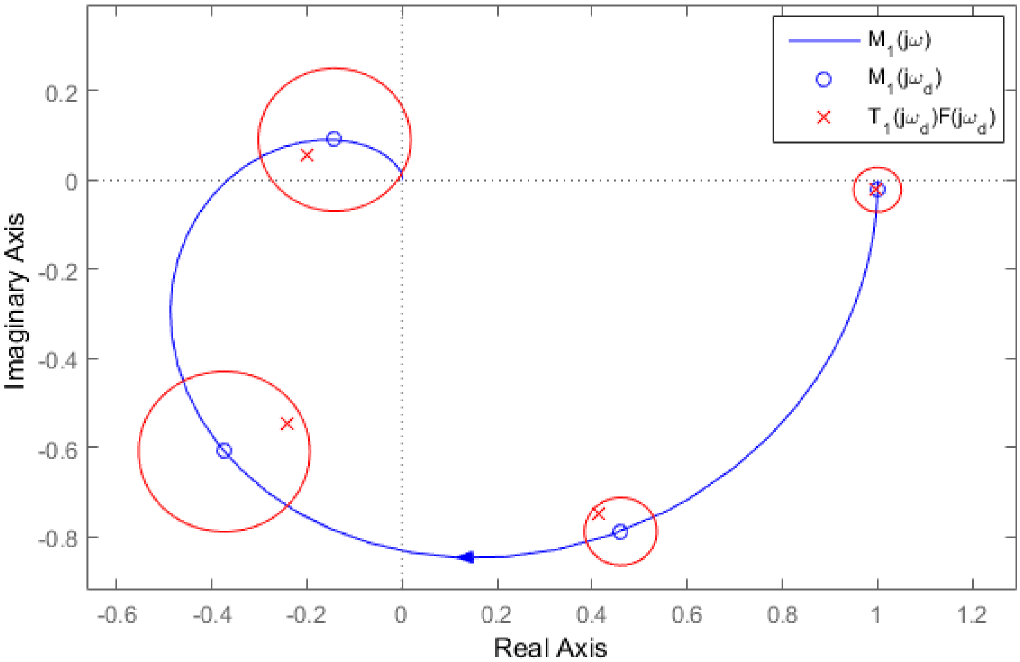


Figure 6.20: Arithmetic complex plane showing $M_1(j\omega)$, from (6.131) and Table 6.1, as well as the discrete model-error tracking requirements at $\omega_d \in \{0.1, 5, 10, 20\}$ rad/s. A valid solution of $T_1(j\omega_d)F(j\omega_d)$ will lie within the error circle, at the corresponding design frequency point ω_d.

covering the entire plant set. Equation (6.117) is a function of both G and F, which makes the bandwidth trade-off between the control elements non-trivial. Conventionally, one might choose an appropriate nominal plant P_0, and fix the pre-filter as $F_0 = M_0/T_0 = M_0(1 + P_0G)/(P_0G)$. This however has the effect of coupling the design of G to the choice of P_0, which can introduce unnecessary conservatism in the resulting controller solution space [103]. To avoid this, we instead build on the work from [4], as follows: Assuming $G(j\omega_d)$ in (6.117) were known, we require $F(j\omega_d) \in \mathbb{S}_{F_i(j\omega_d)}$, where $\mathbb{S}_{F_i(j\omega_d)}$ is the local solution space of $F(j\omega_d)$ for plant case i, which satisfies (6.117). $\mathbb{S}_{F_i(j\omega_d)}$ describes a circle, centred at $M_i(j\omega_d)/T_i(j\omega_d)$, with a radius of $E(\omega_d)/|T_i(j\omega_d)|$,

$$|M_i/T_i - F|\Big|_{\omega_d} \leq \frac{E}{|T_i|}\Big|_{\omega_d}. \tag{6.118}$$

The existence of a global solution space of $F(j\omega_d)$, namely $\mathbb{S}_{F(j\omega_d)}$, requires a non-empty intersection across all local solution spaces of $F(j\omega_d)$,

$$\mathbb{S}_{F(j\omega_d)} = \cap_{i=1}^{m} \mathbb{S}_{F_i(j\omega_d)}. \tag{6.119}$$

With reference to Figure 6.21, this means that the distance between the centres of the solution spaces, $\mathbb{S}_{F_i(j\omega_d)}$ and $\mathbb{S}_{F_k(j\omega_d)}$, must be smaller than or equal to the sum of their respective radii, for all plant cases,

$$|M_i/T_i - M_k/T_k|\Big|_{\omega_d} \leq E\left(\frac{1}{|T_i|} + \frac{1}{|T_k|}\right)\Big|_{\omega_d}, \ \forall i, k, \ \ i \neq k. \tag{6.120}$$

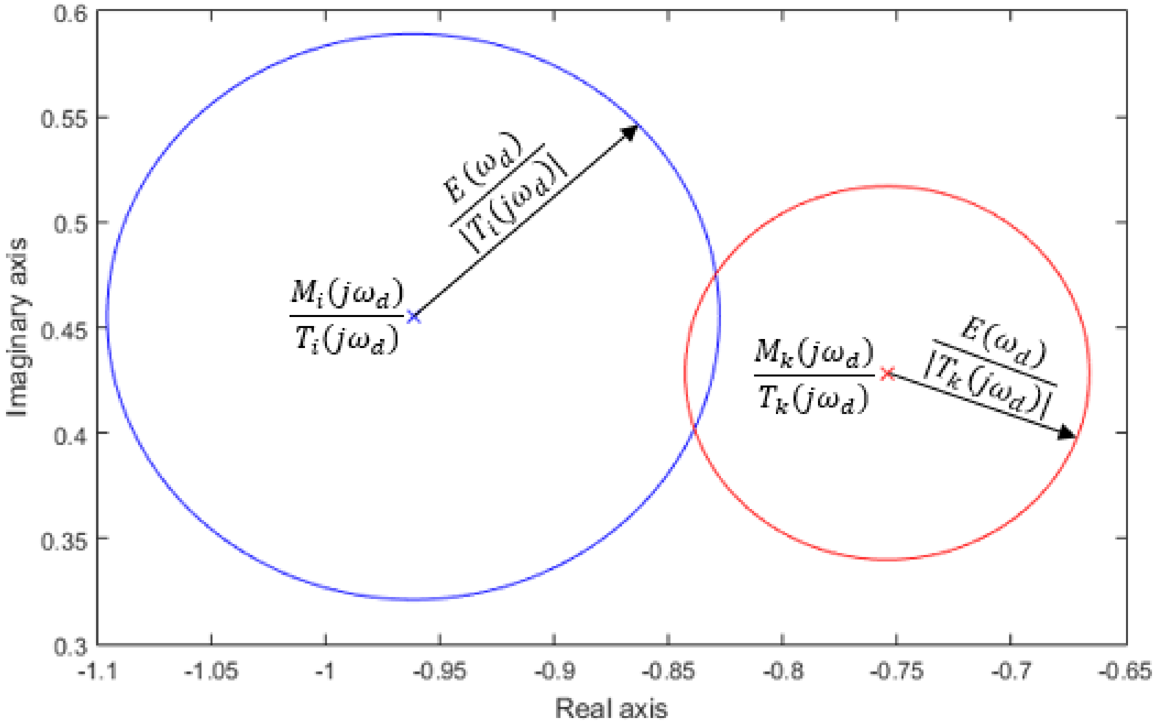


Figure 6.21: Arithmetic complex plane showing local solution spaces $\mathbb{S}_{F_i(j\omega_d)}$ (blue) and $\mathbb{S}_{F_k(j\omega_d)}$ (red). The intersection of the two local solution spaces describes the region where a potential solution, $F_{ik}(j\omega_d)$, can simultaneously satisfy both conditions.

Note that the model-tracking error radius in (6.120) is common between plant i and k. This facilitates equal model-tracking performance across all plants in the plant set. Equation (6.120) can be written as a function of design variable $G(j\omega_d)$,

$$|M_iP_k(1+P_iG) - M_kP_i(1+P_kG)|\Big|_{\omega_d} \leq E\left(|P_k(1+P_iG)| + |P_i(1+P_kG)|\right)\Big|_{\omega_d}, \ \forall i, k, \ \ i \neq k. \tag{6.121}$$

Equation (6.121) is solely a function of $G(j\omega_d)$, and if satisfied for all plant combinations, guarantees the existence of a non-empty global solution space for $F(j\omega_d)$. Note that if $M_i(j\omega_d) = M_k(j\omega_d)$ in (6.121), the above inequality matches the result from [4]. With reference to (6.120) and (6.121), the introduction of the per-plant models means that the existence of $\mathbb{S}_{F(j\omega_d)}$ may not be guaranteed as $|G(j\omega_d)| \rightarrow \infty$ (if

$|M_i(j\omega_d) - M_k(j\omega_d)| > 2E(\omega_d)$, for any one $\{i, k\}$ combination). Instead, there will be unique phase-gain regions of G for the particular plant combination, which will minimise the tracking error. The feedback controller therefore does not perform its traditional role of uniformly reducing closed-loop uncertainty across the plant set, but rather attempts to correctly position each plant's complementary-sensitivity function, in order to improve the overall plant set's model tracking behaviour.

Equation (6.121) requires some manipulation in order to find the solution space $\mathbb{S}_{G_{ik}(j\omega_d)}$. This is done by first converting each element of the inequality in (6.121) to its magnitude-phase form, for example; $P_k(j\omega_d) = |P_k|e^{j\theta_k}\Big|_{\omega_d}$, and $G(j\omega_d) = |G|e^{j\psi}\Big|_{\omega_d}$. The controller phase angle, ψ, is swept across its full range (0° to −360°), such that $|G(j\omega_d)|$ is the only unknown in the equation, for a given ω_d. At each controller phase, (6.121) can be represented as

$$|z_1| \leq |z_2| + |z_3|, \tag{6.122}$$

where z_p is a complex number, and the dependence on $|G(j\omega_d)|$, ψ, and ω_d, is omitted for sake of clarity. Given that $|z_p| = \sqrt{z_p z_p^*}$, squaring both sides will yield

$$z_1 z_1^* \leq z_2 z_2^* + 2\sqrt{z_2 z_2^* z_3 z_3^*} + z_3 z_3^*. \tag{6.123}$$

Next, leaving only the square rooted term on the right-hand side, and squaring again gives

$$(z_1 z_1^* - z_2 z_2^* - z_3 z_3^*)^2 \leq 4 z_2 z_2^* z_3 z_3^*. \tag{6.124}$$

Finally, subtracting the left-hand side from the right-hand side in (6.124) gives the quartic polynomial inequality [109] in $|G(j\omega_d)|$,

$$a_1(\omega_d,\psi)|G(j\omega_d)|^4 + a_2(\omega_d,\psi)|G(j\omega_d)|^3 + a_3(\omega_d,\psi)|G(j\omega_d)|^2 + a_4(\omega_d,\psi)|G(j\omega_d)| + a_5(\omega_d,\psi) \geq 0, \tag{6.125}$$

where the coefficients in (6.125) are functions of the design frequency point and controller phase angle (after evaluating all the plant and model elements). If $M_i(j\omega_d) = M_k(j\omega_d)$, as in [4], (6.125) reduces to a quadratic inequality. The four roots of (6.125) can be obtained using standard root-finding methods. Note that because of the squaring operations, there can be up to two valid roots which correspond to the original inequality of (6.121). These valid roots will then serve as upper or lower bounds on the solution space of $|G(j\omega_d)|$, for a particular ψ, depending on the profile of the quartic polynomial. This means that there can be up to two bounded regions where $|G(j\omega_d)|$ satisfies (6.121).

After sweeping across the entire controller phase range, one obtains the local solution space of $\mathbb{S}_{G_{ik}(j\omega_d)}$. The global solution space (if it exists), for a particular design frequency point, follows as the non-empty intersection across all plant combinations,

$$\mathbb{S}_{G(j\omega_d)} = \cap_{i=1,k=1}^{m,m} \mathbb{S}_{G_{ik}(j\omega_d)}. \tag{6.126}$$

Note that interchanging index i and k in (6.121) does not affect the inequality. Additionally, the case of $i = k$ satisfies (6.121) for all values of $G(j\omega_d)$. This means that there will be $\binom{m}{2} = m(m-1)/2$ unique plant combinations for a particular design frequency point. Once $\mathbb{S}_{G(j\omega_d)}$ is obtained for all frequency points, it can be visualised on the log-polar plane, for shaping the feedback controller. The underlying requirement is that $G(j\omega_d) \in \mathbb{S}_{G(j\omega_d)}$, for each design frequency point. Depending on the specification of $E(\omega_d)$, this may not be possible for all plant combinations. In addition, continuous G needs to satisfy Bode integral relations (i.e. be physically realisable). If $E(\omega_d)$ is not a user specification, but rather a free design parameter, one could instead use a level-set of error radii to provide insight with regard to what is achievable [19]. Design frequency point ω_d will then have a solution space, $\mathbb{S}_{G(j\omega_d,w)}$, corresponding to the chosen error radii $E_w(\omega_d) \in \{\mathbf{E}(\omega_d)\}$, where $w \in \mathbb{Z}^+$. The solution space for a particular $E_w(\omega_d)$ will be a subset of the solution space for any larger error radius, meaning that the admissible design regions will expand monotonically as the error radius is increased. Additionally, the relative expansion of the bounds, for linearly increasing error radii, is related to the gradient of the tracking tolerance, and corresponds to the sensitivity of the error tracking with respect to changes in the control element of interest, at the particular design frequency point. This gives the designer valuable insight into "cost-effective" loop shaping and design frequency performance trade-offs. The relationship between the controller complexity and frequency-based model tracking performance is also made transparent.

Once the feedback element has been chosen, the pre-filter can be designed. With $\mathbf{T}$ fixed, (6.117) describes a linear fractional mapping on $F(j\omega_d)$, which can be resolved to find $\mathbb{S}_{F_i(j\omega_d)}$ at each design frequency point [110]. The global solution space of $F(j\omega_d)$ can then be found using (6.119). The per-frequency solution spaces can be represented on a log-polar plot by comparing the logarithmic gain and phase requirements of $F(j\omega_d)$. With reference to (6.120) and Figure 6.21, the choice of G (and by association, $\mathbf{T}$) will affect the shape and size of $\mathbb{S}_{F(j\omega_d)}$. One could perform the pre-filter bound generation using the same per-frequency error sets from the feedback design stage. However, the additional knowledge gained from fixing G means that it may be more informative to reassess the achievable error tolerances. Referring to (6.121), the minimum achievable model tracking error at design frequency point ω_d is

$$E_{0ik}(\omega_d) = \left.\frac{|M_i P_k(1+P_i G) - M_k P_i(1+P_k G)|}{|P_k(1+P_i G)| + |P_i(1+P_k G)|}\right|_{\omega_d}, \tag{6.127}$$

for plant pairs i and k. By evaluating (6.127) for all plant combinations, one can find the extremum, which corresponds to the minimum tracking error achievable across all plant cases

$$E_0(\omega_d) = max\{E_{0ik}(\omega_d)\}, \forall\, i, k, \;\; i \neq k. \tag{6.128}$$

This can then be used as a starting point for growing the error contours for use in the pre-filter design. Although not directly specified, the original per-plant model-error tracking specifications non-arbitrarily bound the reference to plant input transfer behaviour, with respect to the corresponding model behaviour. Consulting the original specifications in (6.117), dividing both sides by $|P_i(j\omega_d)|$ gives

$$\left|\frac{M_i}{P_i} - \frac{GF}{1+P_iG}\right|\Bigg|_{\omega_d} \leq \frac{E(\omega_d)}{|P_i(j\omega_d)|}, \tag{6.129}$$

where M_i/P_i is the ideal reference to plant input behaviour (from the model optimisation stage), and $U_i/R = GF/(1+P_iG)$ is the closed-loop reference to plant input behaviour. It follows that U_i/R will approach the ideal reference to plant input behaviour as the model tracking error approaches zero. In stark contrast, if there were instead a single nominal reference model, U_i/R would approach M_0/P_i, which may be physically infeasible for low gain plants (in a signal constraint sense). It is worth emphasizing that this frequency-domain method is not a solution to the time-domain saturation problem. However, the inclusion of physically realistic reference models (using constrained optimisation), gives the designer a better handle on the problem.

6.4.4 Worked example

Consider the simple plant set,

$$\mathbf{P} = \frac{\mathbf{k}}{s(1+s/\alpha)}e^{-s\tau}, \tag{6.130}$$

where the plant has input magnitude and rate constraints of $|u(t)| \leq \overline{u} = 0.6$ and $|\dot{u}(t)| \leq \overline{\dot{u}} = 5$ respectively. For sake of simplicity, the expected reference signal is the unit step, $R(s) = 1/s$. Note that this is a simplifying choice to demonstrate the methodology, but other signals could be used, such as a ramp or sinusoid. With reference to Section 6.4.3, the associated model is chosen to be a 4^{th} order all-pole filter, with a pure delay term matching that of the particular plant

$$\mathbf{M} = \frac{1}{\mathbf{a}s^4 + \mathbf{b}s^3 + \mathbf{c}s^2 + \mathbf{d}s + 1}e^{-s\tau}. \tag{6.131}$$

The plant and reference model parameters from (6.229) and (6.131) are detailed in Table 6.1.

Table 6.1: Table of plant parameters and corresponding optimised model parameters.

i	1	2	3	4	5	6	7	8
k_i	10	10	30	30	10	10	30	30
α_i	25	35	25	35	25	35	25	35
τ_i	0	0	0	0	0.02	0.02	0.02	0.02
a_i	7.72×10^{-6}	6.21×10^{-6}	1.95×10^{-6}	1.52×10^{-6}	7.72×10^{-6}	6.21×10^{-6}	1.95×10^{-6}	1.52×10^{-6}
b_i	9.32×10^{-4}	7.65×10^{-4}	2.61×10^{-4}	2.07×10^{-4}	9.32×10^{-4}	7.65×10^{-4}	2.61×10^{-4}	2.07×10^{-4}
c_i	1.81×10^{-2}	1.63×10^{-2}	7.87×10^{-3}	6.75×10^{-3}	1.81×10^{-2}	1.63×10^{-2}	7.87×10^{-3}	6.75×10^{-3}
d_i	0.21	0.20	0.14	0.13	0.21	0.20	0.14	0.13

With the model set fully defined, the process of designing G and F can begin. It is worth mentioning that traditionally selecting a nominal pre-filter of $F_0 = M_0/T_0$ would be problematic here, as the variable plant delay in (6.229) makes the choice of a "good" nominal plant very difficult. Because there are no hard specifications on the allowable model-tracking error, it is treated as a free parameter, and gauged on a per-frequency basis. As previously mentioned, this is done by treating the model tracking error as a level-set, which results in contoured solution spaces. Because G is not yet known, the minimum achievable error tracking radius $E_0(\omega_d)$, is approximated by resolving (6.121) for monotonically increasing $E(\omega_d)$ values, starting from zero. The first value of $E(\omega_d)$ which yields a non-empty solution space for all plant combinations is then treated as the minimum achievable error tracking radius for the particular discrete design frequency point. This minimum can then be linearly scaled (for example) to generate a level-set of achievable error radii. Figure 6.22 shows a set of global solution spaces of G, for a few design frequency points. As previously mentioned, these per-frequency bounds describe a monotonically expanding design region or "contour map", which collectively indicates what level of model-error tracking is achievable across all discrete frequencies and plant cases. Note that even if a certain error bound at ω_d is satisfied by some $G(j\omega_d)$, it may be infeasible, depending on the collective bound requirements on continuous G, as the control elements must obey the Bode gain-phase constraints. This is also true for the design of F. An appropriate subset of the bounds shown in Figure 6.22 can be used to design G.

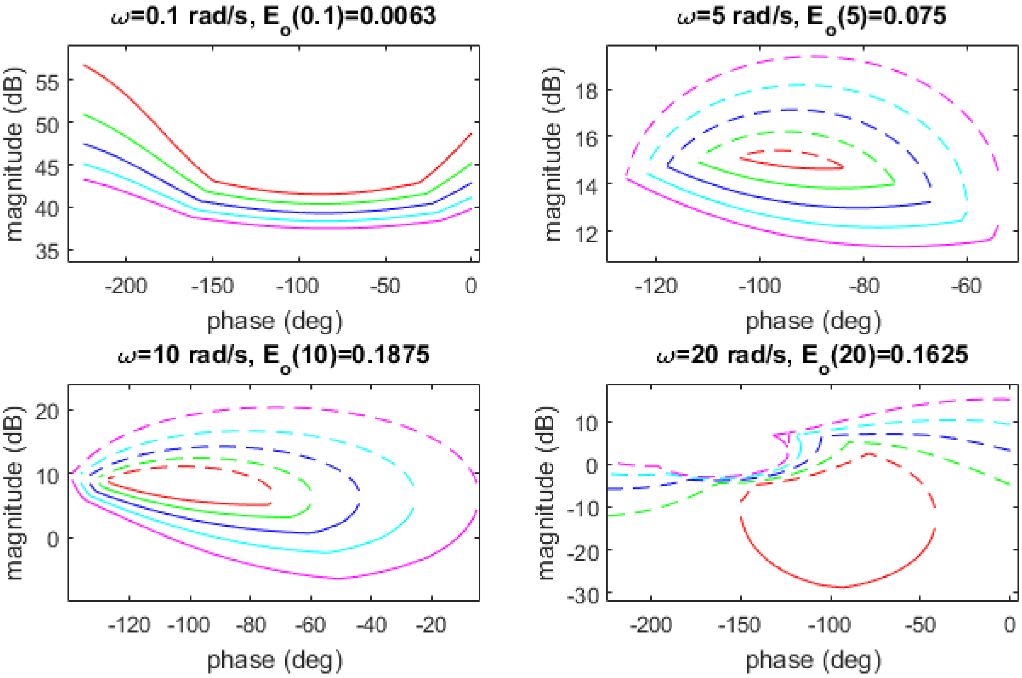


Figure 6.22: Level-set of solution spaces for $G(j\omega)$ at four discrete design frequency points. Solid line corresponds to "stay-above" bound, and dotted line corresponds to "stay-below" bound, for $\omega_d \in \{0.1, 5, 10, 20\}$ rad/s. The error radii at design frequency point ω_d is generated by scaling $E_0(\omega_d)$ by $[1, 1.2, 1.4, 1.6, 1.8]$.

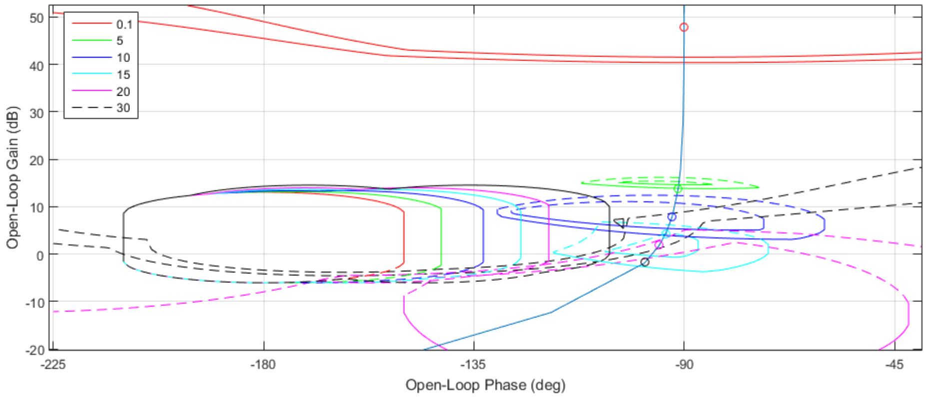


Figure 6.23: Nichols chart of $|L_0(j\omega)|_{db}$ vs $arg\{L_0(j\omega)\}$, showing sensitivity and model tracking bounds, on nominal loop $L_0(j\omega) = P_0(j\omega)G(j\omega)$, for $\omega_d \in \{0.1, 5, 10, 15, 20, 30\}$ rad/s. Two model tracking solution spaces are shown at each discrete design frequency point, using $E_w(\omega_d) \in \{E_0(\omega_d), 1.2E_0(\omega_d)\}$ as the error specification. The resulting nominal loop transfer behaviour, after designing G, is also shown.

For sake of stability robustness, a bound is placed on the closed-loop sensitivity of

$$|S_i(j\omega_d)|_{db} = \left|\frac{1}{1 + P_i(j\omega_d)G(j\omega_d)}\right|_{db} \leq 6 \text{ db}, \ \forall\ \omega_d \in \{\mathbf{\Omega}\},\ P_i \in \{\mathbf{P}\}. \tag{6.132}$$

Additionally, high frequency noise attenuation is addressed by requiring G to be strictly proper. The combination of the sensitivity bounds and model tracking bounds are shown in Figure 6.23 using $E_w(\omega_d) \in \{E_0(\omega_d), 1.2E_0(\omega_d)\}$ as the error specification. In both cases, the bounds on G are shifted and scaled by the same arbitrarily chosen $P_0 = P_4$, in order to show the nominal loop transfer behaviour on a Nichols chart. Although this does not affect the admissible design regions of G, it helps with visualisation for designers who are used to the Nichols chart. With reference to Figure 6.23, there are strict gain and phase constraints in the frequency band of 5 rad/s to 15 rad/s (near the gain-phase cross-over region). In contrast, the high and low frequency design requirements have more design freedom, as G has minimal impact at high frequencies, and is not required at low frequencies (for type 1 plants). With the secondary design goal of sufficiently low-order control design in mind, the resulting controller is chosen to be

$$G = 0.82\frac{(1 + s/33.94)}{(1 + s/529.5)(1 + s/287.9)}. \tag{6.133}$$

The feedback controller is 2nd order, and has the required high frequency roll-off of −20 db/dec. Note that the model-error tracking in Figure 6.23 could be improved, but at the cost of a more complex feedback controller. The trade-off however is transparent when using the level-set of error radii. Following the design of G, the nominal error tolerance, $E_0(\omega)$, is re-evaluated using (6.127) and (6.128), which will be different to that of Figure 6.22 if $G(j\omega)$ does not lie on the boundary condition for each design frequency of interest. $E_0(\omega)$ can then be scaled to facilitate the contoured design regions of $F(j\omega)$. The design of F is then formulated using

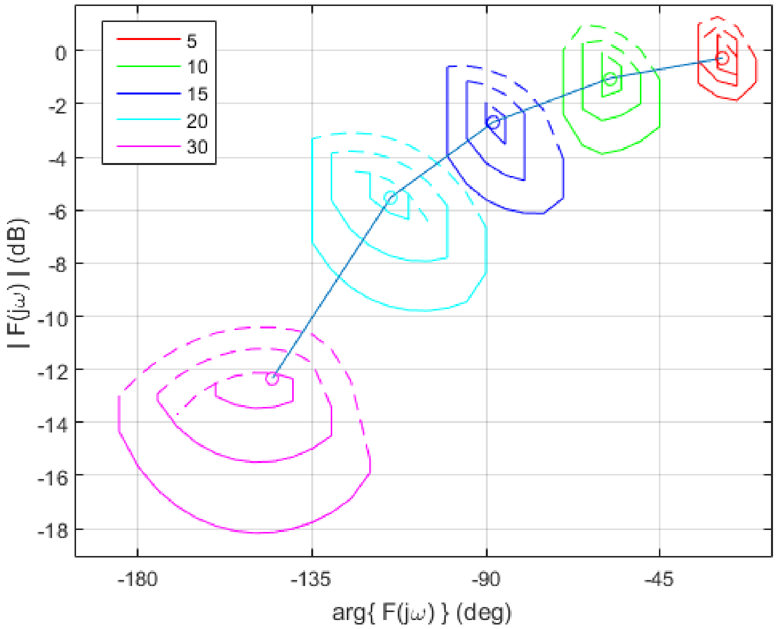


Figure 6.24: Log-polar plot of $|F(j\omega)|_{db}$ vs $arg\{F(j\omega)\}$, showing multiple global solution spaces of $F(j\omega_d)$, using three error specifications, for $\omega_d \in \{5, 10, 15, 20, 30\}$ rad/s. The designed pre-filter, F, is also shown.

the original per-plant model-error tracking inequality of (6.117). Figure 6.24 shows the per-frequency global solution spaces of $F(j\omega_d)$, visualised on a log-polar plot. A sufficiently low-order pre-filter which suitably reduces the frequency-based error tolerances is

$$F = \frac{(1 + s/35.36)(1 + s/387.5)}{(1 + s/15.29)(1 + 2(0.62)/19.05s + s^2/19.05^2)}. \tag{6.134}$$

Again, it should be noted that by using a level-set of error radii, one can easily trade pre-filter complexity with error tracking performance. Additionally, the Bode gain-phase requirements means that the smallest error bounds may not be achievable across all discrete design frequency points. Although the reference models are designed to satisfy the input constraints, because non-zero tracking error is allowed, time-domain simulation is required to check the input signal levels (and rates). The set of output responses, $\mathbf{y}(t)$, and corresponding model responses, $\mathbf{y}_m(t)$, is shown in Figure 6.25. With reference to Figure 6.25, the output step responses are located near their associated model responses. In fact, there appears to be two major groupings, which correspond to the high and low gain plants, respectively. The model-matching ability of the uncertain system is fundamentally limited by the "agreement" of the individual reference models, with respect to the plant set. For example, if one were to initially fix $G = G_0$ and $F = F_0$, and then use the resulting closed-loop performance as the reference model, the 2DOF controller solution space would contain a single solution of $G = G_0$ and $F = F_0$, which yields exact model-matching across the plant set (perfect "agreement"). The closed-loop input and input-rate responses are shown in Figure 6.26. The two time-domain responses of Figure 6.25 and Figure 6.26 reaffirm

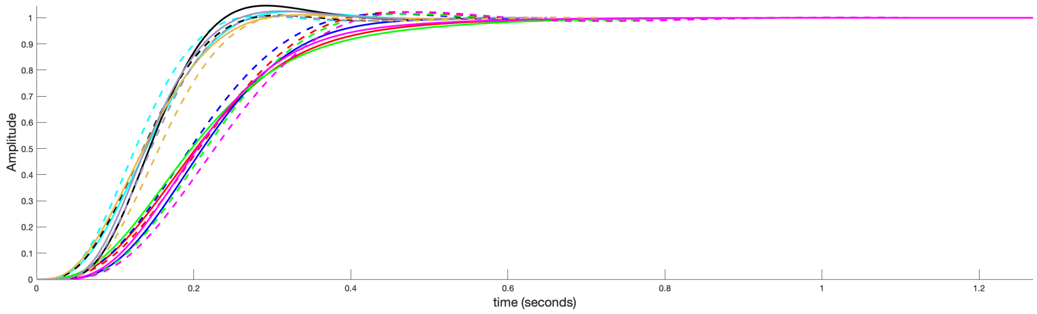


Figure 6.25: Step response of closed-loop plant set (solid lines) and corresponding model set (dashed lines), for a reference signal of $r(s) = 1/s$, applied at $t = 0$ seconds. Each closed-loop plant and model pair is shown with a distinct colour.

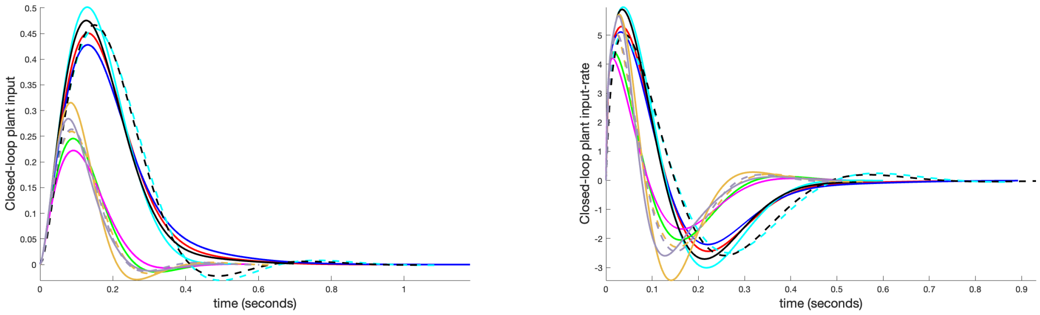


Figure 6.26: Closed-loop plant input response (left) and plant input-rate response (right) for plant set, shown with solid lines. The corresponding reference model responses are also shown as dotted lines. Each closed-loop plant and model pair is shown with a distinct colour.

that each closed-loop plant can operate near its corresponding reference model, without requiring exotic input and input-rate commands.

6.4.5 Conclusion

This Section has presented a new approach to designing a robust controller for an uncertain plant set. In particular, it was shown that by applying bespoke tracking specifications to each plant case, based on that plant's capability, the plant set could achieve improved, realistic overall performance. Additionally, making use of error radii level-sets, the frequency-based performance vs controller complexity trade-offs were made transparent. The method was illustrated using a simple problem, which yielded the expected per-plant model following, given the large plant set uncertainty. This design methodology is particularly useful for discrete plant sets with fundamentally different gain-phase profiles.

6.5 A complementary QFT solution to the 2x2 tracking error problem

6.5.1 Abstract

This Section presents a solution to the 2x2 multivariable tracking error problem. Current Quantitative Feedback Theory (QFT) methods commonly employ plant-inverting splittings in order to arrive at an approximately decoupled design on the feedback control elements. This results in suitable design regions in the high-gain, low-frequency range, but can result in conservative design at and above the gain-crossover frequency, resulting in over-designed feedback controllers. Our method aims to reduce this conservatism by supplementing a plant-inverting design with a non plant-inverting design, which tends to perform well at higher frequencies. Decoupling the design of the controller elements is facilitated by making use of appropriate existence conditions. The union of the two resulting design regions then describes a larger solution space, and by enlarging the admissible design regions for the feedback controller, lower gain feedback controller design at all design frequencies of interest may be possible. A benchmarking example is presented, showing the viability of the proposed method.

6.5.2 Introduction

In terms of the single-input-single-output (SISO) QFT tracking problem, nonconservative (ie exact) solutions exist, be it in the form of above/below tracking magnitude bounds [111], or tracking error bounds [103, 4]. However, in the case of solving multi-input-multi-output (MIMO) problems, the current state of the art involves varying levels of conservatism [77]. This comes as a result of two design challenges, namely (i) the bandwidth trade-off between feedback controller and pre-filter, and (ii) decoupling the design of the feedback control elements. (i) was originally addressed by using the method of above/below magnitude specifications on the reference to output transfer behaviour [82]. The structure of the design equations however requires the use of a diagonal pre-filter, which may artificially limit the potential gain saving in the feedback controller. Additionally, the lack of phase information present in the specifications incurs additional conservatism, and can reduce tracking precision [13]. The concept of using tracking error tolerances was introduced for SISO systems in [15]. This method handles the two-degree-of-freedom (2DOF) trade-off by setting the pre-filter equal to the prescribed model performance, prior to design of the feedback controller. This simple method captures phase information between the model and closed-loop response, and allows for direct design on the feedback controller. The sub-optimal selection of nominal plant however, places unnecessary over-design on the feedback controller. The multivariable equivalent method was developed in [112], whereby the nominal pre-filter additionally incorporates information from a selected nominal plant . Once the feedback controller has been designed, the pre-filter is adjusted (if need be) by inferring differential error specifications on the nominal pre-filter. Although this assists the bandwidth trade-off, selection of an appropriate nominal plant for the entire frequency range of interest is not obvious, and may lead to additional over-design on the feedback control design. A different design methodology was introduced in [113], whereby a sensitivity design could proceed, based on establishing necessary conditions for meeting the tracking error specifications. A diagonal pre-filter could then be designed subsequently, using the original tracking error specifications. This method has the benefit that the feedback control design is not biased by a selected nominal plant. The aforementioned necessary conditions however, do not guarantee that the design can be completed. More recently, a less conservative solution was proposed, which separates the element-wise tracking error problem into a simultaneous tracking and disturbance rejection problem, using manual bound balancing methods [5]. The design of the pre-filter and feedback controller can then be decoupled, by defining existence conditions on the element-wise pre-filter solution space. Successful design of the diagonal feedback controller enables the design of a fully populated pre-filter. This method is similarly implemented for the measured disturbance rejection problem in [81]. However, as an optimal bound balancing requires the scaling parameter to vary with the controller phase and particular plant instance (to ensure both design regions overlap perfectly), the implementation in [5] will, in general, incur additional design conservatism.

Although the aforementioned methods handle the 2DOF bandwidth trade-off of (i) with varying success, the feedback controller decoupling of (ii) is, in general, facilitated by using plant-inverting splittings. The natural drawback of this splitting is the required feedback design conservatism at higher frequencies where low loop gain is desired, as a result of the (potentially) improper plant inverse rolling up at high frequency. As the designer is primarily concerned with feedback design in the gain-phase crossover region, one can only benefit from having more design freedom at the higher frequencies, especially in the case of non-minimum phase systems. This Section therefore presents a novel method, geared at 2x2 plants, which combines a plant-inverting design routine with a non plant-inverting design. First, the bandwidth trade-off is facilitated by defining appropriate existence conditions for a feed-forward filter which can meet the tracking specifications. The plant-inverting method is then used to generate acceptable design regions for the feedback controller, which generally incur less design conservatism at low frequencies. Similarly, using a non plant-inverting method, one can arrive at design regions on the feedback controller, which are less conservative at high frequencies. As only one of the two resulting design regions needs to be adhered to, the union of the two regions describes an expanded solution space, on a discrete frequency-by-frequency basis. In this way, the designer has more freedom in shaping the feedback controller behaviour. Once the feedback control design is complete, the non-diagonal pre-filter is designed without the use of over-bounding methods, using similar existence conditions for meeting the underlying tracking error specifications. As this method is a QFT design tool which finds feasible regions at discrete design frequency points, the designer is still required to join these regions with a low order, causal controller, capable of achieving closed-loop plant stability. This method is bench-marked using a simple example, which demonstrates its utility, especially at the higher frequencies.

6.5.3 Problem statement

Consider the two 2DOF control configurations of Figure 6.27 and Figure 6.28, where $P^i(s)$ is a 2×2 plant instance in the uncertain set $\mathcal{P}(s)$; $r(s)$ is the 2×1 reference signal; and $M(s)$ is the 2×2 reference model to be tracked. The set $\mathcal{P}(s) = \{P^1(s), P^2(s), ..., P^n(s)\}$ is an *ε-net* of disjoint plants which sufficiently describe the uncertain plant. Diagonal feedback controller $G(s)$ =diag$\{g_1(s), g_2(s)\}$ acts to reduce uncertainty across the plant set, while 2×2 pre-filter $F(s)$ (feedforward filter $X(s)$ in Figure 6.28) is designed to adjust the closed-loop bandwidth of the plant set, as required.

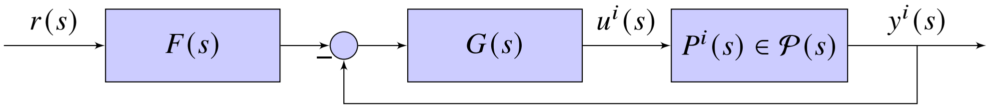


Figure 6.27: Block diagram of conventional control configurations with two degrees of freedom, $\{G(s), F(s)\}$.

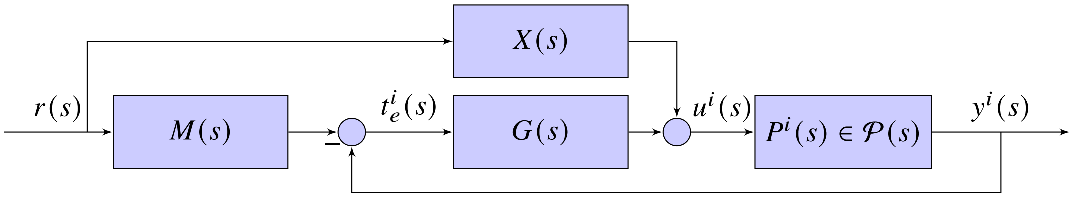


Figure 6.28: Block diagram of feed-forward control scheme (still with two degrees of freedom, $\{G(s), X(s)\}$).

If $F(s) = M(s) + G^{-1}(s)X(s)$, both schemes are equivalent. The structure in Figure 6.28 however, has the benefit in that it inherently captures the reference model tracking error

$$t_e^i(s) = E^i(s)r(s) = M(s)r(s) - y^i(s), \tag{6.135}$$

where

$$E^i(s) = \left(I + P^i(s)G(s)\right)^{-1}\left(M(s) - P^i(s)X(s)\right). \tag{6.136}$$

Following standard tracking error QFT methods [112], the element-wise magnitude of the matrix reference model tracking error frequency response in (6.136) can be constrained by a defined specification,

$$|e_{rc}^i(j\omega_d)| \le \beta_{rc}(\omega_d), \tag{6.137}$$

where $\beta_{rc}(\omega_d)$ is the user-specified reference model tracking error tolerance at row r and column c, and $\omega_d \in \Omega$ is the discrete design frequency point of interest. The design frequency set, Ω, is chosen based on engineering understanding of the control problem, and is required to be sufficiently rich, in order to capture the contiguous frequency behaviour of the closed-loop system. The objective is to design $G(s)$ and $X(s)$ in order to meet the constraints of (6.186) for all plant cases, $i \in \{1, 2, ..., n\}$. Additionally, cost-effective design requires that the bandwidth and complexity/order of $G(s)$ be minimised.

6.5.4 Method

Feedback control design

Plant-inverting splitting

The relationship between $X(j\omega_d)$ and $G(j\omega_d)$ can be assessed by using diagonal matrix splittings. This is commonly used to decouple the element-wise design of $G(s)$, at the cost of coupling the elements of $E^i(s)$. With reference to (6.136), pre-multiplying both sides by $\left(\hat{P}^i(s)+G(s)\right)$, where $\hat{P}^i(s)$ is the inverse of $P^i(s)$,

$$\left(\hat{P}^i(s)+G(s)\right)E^i(s)=\hat{P}^i(s)M(s)-X(s). \tag{6.138}$$

Next, splitting $\hat{P}^i(s)$ into its diagonal and off-diagonal components, $\hat{P}^i(s)=\hat{P}^i_d(s)+\hat{P}^i_n(s)$, (6.138) can be written as

$$\left(\hat{P}^i_d(s)+G(s)\right)\left(I+\left(\hat{P}^i_d(s)+G(s)\right)^{-1}\hat{P}^i_n(s)\right)E^i(s)=\hat{P}^i(s)M(s)-X(s). \tag{6.139}$$

Finally, $E^i(s)$ can be written implicitly, thereby becoming decoupled with respect to design variable $G(s)$,

$$E^i(s)=\left(\hat{P}^i_d(s)+G(s)\right)^{-1}\left(\hat{P}^i(s)M(s)-X(s)-\hat{P}^i_n(s)E^i(s)\right). \tag{6.140}$$

Making use of (6.186), the element-wise magnitude constraints of (6.140) for a 2x2 problem, evaluated at discrete design frequency point ω_d, follows as

$$\begin{bmatrix}|e_{11}| & |e_{12}|\\ |e_{21}| & |e_{22}|\end{bmatrix}^i_{\omega_d}=\begin{bmatrix}\dfrac{|(\hat{p}_{11}m_{11}+\hat{p}_{12}m_{21}-x_{11})-\hat{p}_{12}e_{21}|}{|\hat{p}_{11}+g_1|} & \dfrac{|(\hat{p}_{11}m_{12}+\hat{p}_{12}m_{22}-x_{12})-\hat{p}_{12}e_{22}|}{|\hat{p}_{11}+g_1|}\\ \dfrac{|(\hat{p}_{21}m_{11}+\hat{p}_{22}m_{21}-x_{21})-\hat{p}_{21}e_{11}|}{|\hat{p}_{22}+g_2|} & \dfrac{|(\hat{p}_{21}m_{12}+\hat{p}_{22}m_{22}-x_{22})-\hat{p}_{21}e_{12}|}{|\hat{p}_{22}+g_2|}\end{bmatrix}^i_{\omega_d}\leq\begin{bmatrix}\beta_{11} & \beta_{12}\\ \beta_{21} & \beta_{22}\end{bmatrix}_{\omega_d}. \tag{6.141}$$

The unknown $e^i_{rc}(j\omega_d)$ terms appearing implicitly in (6.141) necessitates an over-bounding, using the triangle inequality, which results in the conservative element-wise constraint [112] of

$$\begin{bmatrix}|e_{11}| & |e_{12}|\\ |e_{21}| & |e_{22}|\end{bmatrix}^i_{\omega_d}\leq\begin{bmatrix}\dfrac{|\hat{p}_{11}m_{11}+\hat{p}_{12}m_{21}-x_{11}|+|\hat{p}_{12}|\beta_{21}}{|\hat{p}_{11}+g_1|} & \dfrac{|\hat{p}_{11}m_{12}+\hat{p}_{12}m_{22}-x_{12}|+|\hat{p}_{12}|\beta_{22}}{|\hat{p}_{11}+g_1|}\\ \dfrac{|\hat{p}_{21}m_{11}+\hat{p}_{22}m_{21}-x_{21}|+|\hat{p}_{21}|\beta_{11}}{|\hat{p}_{22}+g_2|} & \dfrac{|\hat{p}_{21}m_{12}+\hat{p}_{22}m_{22}-x_{22}|+|\hat{p}_{21}|\beta_{12}}{|\hat{p}_{22}+g_2|}\end{bmatrix}^i_{\omega_d}\leq\begin{bmatrix}\beta_{11} & \beta_{12}\\ \beta_{21} & \beta_{22}\end{bmatrix}_{\omega_d}. \tag{6.142}$$

Multiplying through by the element-wise denominator of (6.142), the generalised conservative constraint for $|e^i_{rc}(j\omega_d)|$ is

$$|\hat{p}_{r1}m_{1c}+\hat{p}_{r2}m_{2c}-x_{rc}|^i_{\omega_d}\leq|\hat{p}_{rr}+g_r|\beta_{rc}\Big|^i_{\omega_d}-|\hat{p}_{rv}|\beta_{vc}\Big|^i_{\omega_d}, \tag{6.143}$$

where $v=(2,1)$ for $r=(1,2)$ respectively. The amount of conservatism induced by over-bounding in (6.142) is related to the size of $|\hat{p}_{rv}|\beta_{vc}\big|^i_{\omega_d}$, relative to the size of $\beta_{rc}|\hat{p}_{rr}+g_r|\big|^i_{\omega_d}$. In particular, if $|\hat{p}_{rv}|\beta_{vc}\big|^i_{\omega_d}<<\beta_{rc}|\hat{p}_{rr}+g_r|\big|^i_{\omega_d}$, there will be negligible conservatism. For the class of problems with zero steady-state tracking error specifications, $P^i(s)$ and/or $G(s)$ will contain at least one integrator in the channel of interest. This means that $\beta_{rc}|\hat{p}_{rr}+g_r|\big|^i_{\omega_d}$ will dominate $|\hat{p}_{rv}|\beta_{vc}\big|^i_{\omega_d}$ at low frequencies (implying low design conservatism), whereas the opposite will occur at high frequencies for sufficiently coupled plants (implying high design conservatism). This is a natural result of the plant-inverting splitting [77]. With reference to Figure 6.29, (6.143) can be written as

$$|c_i - x_{rc}(j\omega_d)| \leq r_i, \tag{6.144}$$

and geometrically describes the discoidal solution space of $x_{rc}(j\omega_d)$ (for plant instance i), with centre $c_i = (\hat{p}_{r1}m_{1c} + \hat{p}_{r2}m_{2c})\big|^i_{\omega_d}$, and radius $r_i = (\beta_{rc}|\hat{p}_{rr} + g_r| - |\hat{p}_{rv}|\beta_{vc})\big|^i_{\omega_d}$. Note that the radius of the admissible design region is reduced by the worst-case phase of $-\hat{p}_{rv}e_{vc}\big|^i_{\omega_d}$. In order for a $x_{rc}(j\omega_d)$ to exist which satisfies (6.143) for all plant cases, there must be a common intersection for the set of plant dependent discoidal design regions. This necessitates that the distance between each ordered (by plant index) pair of disc centres must be

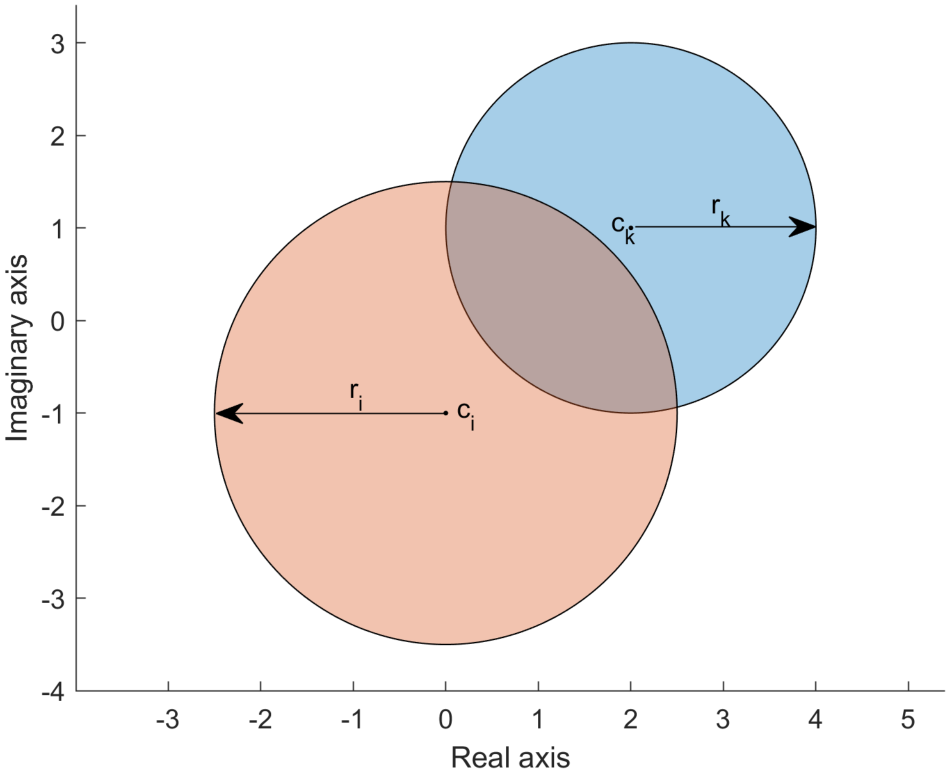


Figure 6.29: Arithmetic-complex plane of $x_{rc}(j\omega_d)$, showing non-empty intersection of two discoidal design regions. Overlapping occurs when $|c_i - c_k| \leq r_i + r_k$. A valid $x_{rc}(j\omega_d)$, that simultaneously satisfies plant instance i and k, is required to lie within the area of intersection.

smaller than or equal to the sum of their respective radii ($|c_i - c_k| \leq r_i + r_k$) [4],

$$\left|(\hat{p}^i_{r1} - \hat{p}^k_{r1})m_{c1} + (\hat{p}^i_{r2} - \hat{p}^k_{r2})m_{c2}\right|_{\omega_d} \leq \beta_{rc}(\omega_d)\left(|\hat{p}_{rr} + g_r|^i_{\omega_d} + |\hat{p}_{rr} + g_r|^k_{\omega_d}\right) - \beta_{vc}(\omega_d)\left(|\hat{p}_{rv}|^i_{\omega_d} + |\hat{p}_{rv}|^k_{\omega_d}\right), \ \forall i, k. \tag{6.145}$$

Ensuring that each ordered pair of discs individually intersect, does not, in general, guarantee that a common intersection exists across all discs (contrary to the affirmation originally stated in [4]. However, the likelihood of a common intersection occurring will increase as the number of discs increase (each disc's radius is parameterised by a common $g_r(j\omega_d)$). Equation (6.145) is solely a function of $g_r(j\omega_d)$ and can be used to generate design regions directly on $g_r(j\omega_d)$. Interestingly, the per-plant solution space of $g_r(j\omega)$ is a "stay-out" ellipse, with foci at $-\hat{p}^i_{rr}(j\omega_d)$ and $-\hat{p}^k_{rr}(j\omega_d)$, and a major axis of $\left.\left(|(\hat{p}^i_{r1} - \hat{p}^k_{r1})m_{c1} + (\hat{p}^i_{r2} - \hat{p}^k_{r2})m_{c2}| + \beta_{vc}(|\hat{p}_{rv}|^i + |\hat{p}_{rv}|^k)\right)/\beta_{rc}\right|_{\omega_d}$. This closely resembles the result in [4]. For a 2x2 plant, there will be two sets of constraints for the particular $g_r(j\omega_d)$ because two elements in (6.142) depend on $g_r(j\omega_d)$. It is worth noting that the design regions on $g_r(j\omega_d)$, resulting from (6.145), would be equivalent to that of [5] if the scaling parameters used in the bound balancing stage of [5] were selected optimally for the particular plant instance (as the "worst-case" plant may vary across the partitioned bounds) and controller phase angle (in order to ensure that one boundary does not dominate the other boundary at any phase angle). The non-empty intersection of the resulting design regions describes the (conservative) solution space of $g_r(j\omega_d)$, for the particular plant-inverting splitting, namely $\mathbb{S}^{\hat{P}}_{g_r(j\omega_d)}$. As with other QFT methods which incorporate the triangle inequality, the tracking tolerances may need to be adjusted, in order to arrive at a design equilibrium between loops [114]. Note that this method can be generalised for MIMO systems of any order. As previously mentioned, these design regions on $g_r(j\omega_d)$ tend to be favourable at low frequencies because of high loop gain, but may perform poorly at higher frequencies (for sufficiently coupled plants). To remedy this, we seek a complementary splitting, which will perform well at high frequencies.

Non plant-inverting splitting

A non plant-inverting splitting is applied to (6.136) as follows: Pre-multiplying both sides of (6.136) by $\left(I + P^i(s)G(s)\right)$ yields

$$\left(I + P^i(s)G(s)\right)E^i(s) = M(s) - P^i(s)X(s). \tag{6.146}$$

Using a diagonal splitting, $L^i(s)$ is written as $L^i(s) = P^i(s)G(s) = L^i_d(s) + L^i_n(s)$, and (6.146) can be expanded as

$$\left(I + L^i_d(s)\right)\left(I + \left(I + L^i_d(s)\right)^{-1} L^i_n(s)\right)E^i(s) = M(s) - P^i(s)X(s). \tag{6.147}$$

Finally, error matrix $E^i(s)$ can be described implicitly as

$$E^i(s) = \left(I + L^i_d(s)\right)^{-1}\left(M(s) - P^i(s)X(s) - L^i_n(s)E^i(s)\right). \tag{6.148}$$

The corresponding element-wise magnitude constraint for a 2×2 system, at design frequency point ω_d, follows as

$$\begin{bmatrix} |e_{11}| & |e_{12}| \\ |e_{21}| & |e_{22}| \end{bmatrix}^i_{\omega_d} = \begin{bmatrix} \left| \dfrac{(m_{11} - p_{12}x_{21} - p_{11}x_{11}) - p_{12}g_2e_{21}}{1 + p_{11}g_1} \right| & \left| \dfrac{(m_{12} - p_{12}x_{22} - p_{11}x_{12}) - p_{12}g_2e_{22}}{1 + p_{11}g_1} \right| \\ \left| \dfrac{(m_{21} - p_{22}x_{21} - p_{21}x_{11}) - p_{21}g_1e_{11}}{1 + p_{22}g_2} \right| & \left| \dfrac{(m_{22} - p_{22}x_{22} - p_{21}x_{12}) - p_{21}g_1e_{12}}{1 + p_{22}g_2} \right| \end{bmatrix}^i_{\omega_d} \leq \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}_{\omega_d}. \tag{6.149}$$

Applying the triangle inequality, (6.149) can be over-bounded (element-wise) by

$$\begin{bmatrix} |e_{11}| & |e_{12}| \\ |e_{21}| & |e_{22}| \end{bmatrix}^i_{\omega_d} \leq \begin{bmatrix} \dfrac{|m_{11} - p_{12}x_{21} - p_{11}x_{11}| + |p_{12}||g_2|\beta_{21}}{|1 + p_{11}g_1|} & \dfrac{|m_{12} - p_{12}x_{22} - p_{11}x_{12}| + |p_{12}||g_2|\beta_{22}}{|1 + p_{11}g_1|} \\ \dfrac{|m_{21} - p_{22}x_{21} - p_{21}x_{11}| + |p_{21}||g_1|\beta_{11}}{|1 + p_{22}g_2|} & \dfrac{|m_{22} - p_{22}x_{22} - p_{21}x_{12}| + |p_{21}||g_1|\beta_{12}}{|1 + p_{22}g_2|} \end{bmatrix}^i_{\omega_d} \leq \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}_{\omega_d}. \tag{6.150}$$

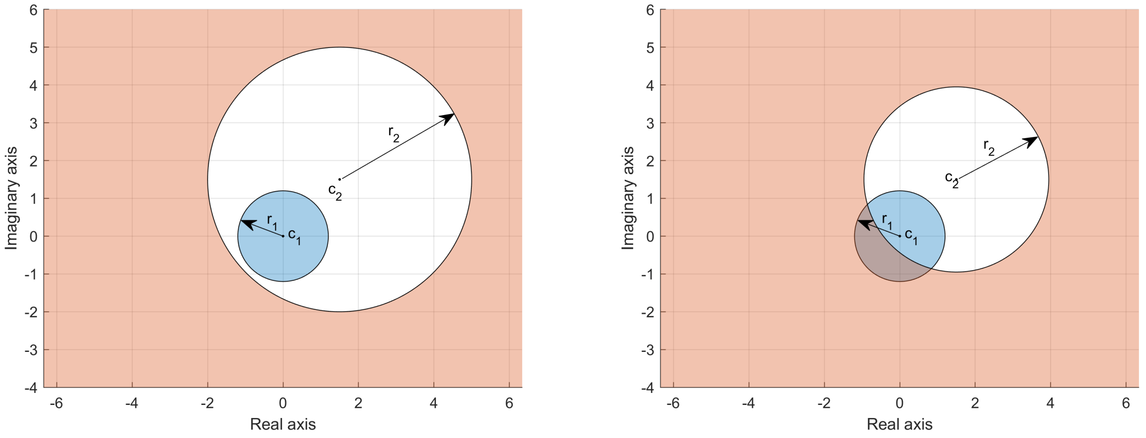
Multiplying through by the element-wise denominator of (6.150), the generalised constraint for $|e^i_{rc}(j\omega_d)|$ is

$$|m_{rc} - p_{r1}x_{1c} - p_{r2}x_{2c}|^i_{\omega_d} \leq \beta_{rc}|1 + p_{rr}g_r|\Big|^i_{\omega_d} - \beta_{vc}|p_{rv}||g_v|\Big|^i_{\omega_d}, \tag{6.151}$$

where $v = (2,1)$ for $r = (1,2)$, respectively. As the elements of $L^i(s)$ are expected to be strictly proper (by correct design), $\beta_{rc}|1 + p_{rr}g_r||^i_{\omega_d}$ will dominate $\beta_{vc}|p_{rv}||g_v||^i_{\omega_d}$ at sufficiently high frequencies (outside the closed-loop bandwidth), implying low design conservatism. The low frequency counterpart will however, experience conservatism (unless the plant is weakly coupled). Unlike in (6.143), the structure of (6.151) does not lend itself to closed-form existence conditions for separating the design of $G(s)$ and $X(s)$. Instead, the uncoupled design of the control elements is facilitated by selecting a frequency response of $X(j\omega_d)$ at each design point. This prototype $X(j\omega_d)$ allows for direct design on $G(s)$, after which $X(s)$ can be designed in earnest. With reference to (6.151), minimising $|m_{rc} - p_{r1}x_{1c} - p_{r2}x_{2c}|^i_{\omega_d}$ will reduce the amount of feedback gain required. Specifically, $x_{11}(j\omega_d)$ and $x_{21}(j\omega_d)$ are found to minimise $|m_{11} - p_{12}x_{21} - p_{11}x_{11}|^i_{\omega_d}$ and $|m_{21} - p_{22}x_{21} - p_{21}x_{11}|^i_{\omega_d}$ simultaneously. Similarly, $x_{12}(j\omega_d)$ and $x_{22}(j\omega_d)$ are selected to minimise $|m_{12} - p_{12}x_{22} - p_{11}x_{12}|^i_{\omega_d}$ and $|m_{22} - p_{22}x_{22} - p_{21}x_{12}|^i_{\omega_d}$. As there are multiple plant instances, the optimisation aims to minimise the worst-case occurrence across the enumerated plant set. The corresponding minimax problem follows as

$$\min_{x_{1c}(j\omega_d), x_{2c}(j\omega_d) \in \mathbb{C}} \max \left\{ \sum_{r=1}^{2} \frac{|m_{rc} - p_{r1}x_{1c} - p_{r2}x_{2c}|}{\beta_{rc}} \Bigg|^i_{\omega_d}, \forall i \in \{1, 2, ...n\} \right\}, \tag{6.152}$$

where $\beta_{rc}(\omega_d)$, from (6.186), is used to weight the terms to be minimised. Given that $x_{1c}(j\omega_d)$ and $x_{2c}(j\omega_d)$ are complex numbers, the optimisation routine for column c solves for four free parameters, and is computationally straightforward. This prototype selection of $X(j\omega_d)$ can be thought of as a sufficient condition for the existence of a $X(s)$, once $G(s)$ has been designed (assuming an $X(s)$ exists which can join the discrete



(a) $|c_2 - c_1| < r_2 - r_1$ (b) $|c_2 - c_1| \geq r_2 - r_1$

Figure 6.30: Arithmetic-complex plane, showing an example of the admissible design regions (shaded) of $g_2(j\omega_d)$: "stay-in" disc (blue) and "stay-out" disc (white). (a) An empty (invalid) design region when the "stay-out" disc envelops the "stay-in" disc. (b) A non-empty design region when the "stay-out" disc does not fully envelop the "stay-in" disc.

design regions), albeit a (potentially) more limiting one than that of (6.145).

Using prototype $X(j\omega_d)$ from (6.152), generation of the non plant-inverting feedback design regions can commence. Notably, the constraint in (6.151) is a function of both $g_1(j\omega_d)$ and $g_2(\omega_d)$. In order to remove the dependence on $g_2(j\omega_d)$ when designing $g_1(j\omega_d)$ first (for example), one can define necessary and sufficient conditions on $g_1(j\omega_d)$, which guarantee the existence of at least one valid $g_2(j\omega_d)$ satisfying (6.151) for plant i. Setting known $K^i(j\omega_d) = M(j\omega_d) - P^i(j\omega_d)X(j\omega_d)$ for sake of clarity, the constraints on $g_2(j\omega_d)$ from the first row of (6.150) can be written as

$$|g_2(j\omega_d)| \leq \left. \frac{\beta_{1c}|1 + p_{11}g_1| - |k_{1c}|}{\beta_{2c}|p_{12}|} \right|_{\omega_d}^{i}, \tag{6.153}$$

whereas the constraints on $g_2(j\omega_d)$ from the second row of (6.150) can be written as

$$\left| g_2 - \left. \frac{-1}{p_{22}} \right|_{\omega_d}^{i} \right| \geq \left. \frac{|k_{2c}| + \beta_{1c}|p_{21}||g_1|}{\beta_{2c}|p_{22}|} \right|_{\omega_d}^{i}. \tag{6.154}$$

With reference to Figure 6.30, the admissible design region of $g_2(j\omega_d)$ in (6.153) is a "stay-in" disc, centred at the origin ($c_1 = 0$), with a radius of $r_1 = (\beta_{1c}|1+p_{11}g_1| - |k_{1c}|)/(\beta_{2c}|p_{12}|)|_{\omega_d}^{i}$, whereas the design region of (6.154) is a "stay-out" disc, centred at $c_2 = -1/p_{22}^{i}(j\omega_d)$, with a radius of $r_2 = (|k_{2c}| + \beta_{1c}|p_{21}||g_1|)/(\beta_{2c}|p_{22}|)\big|_{\omega_d}^{i}$. The existence of a non-empty design region of $g_2(j\omega)$, which simultaneously satisfies (6.153) and (6.154) for plant instance i and column c demands that the "stay-out" disc does not fully envelop the "stay-in" disc.

Referring to Figure 6.30, this requires that the distance between the centres of the two discs must be greater than or equal to the radius of the "stay-out" disc, minus the radius of the "stay-in" disc ($|c_2 - c_1| \geq r_2 - r_1$), explicitly described as

$$\left|\frac{1}{p_{22}}\right|_{\omega_d}^{i} \geq \left.\frac{|k_{2c}| + \beta_{1c}|p_{21}||g_1|}{\beta_{2c}|p_{22}|}\right|_{\omega_d}^{i} - \left.\frac{\beta_{1c}|1 + p_{11}g_1| - |k_{1c}|}{\beta_{2c}|p_{12}|}\right|_{\omega_d}^{i}. \tag{6.155}$$

Satisfying the constraint in (6.155) will therefore guarantee the existence of a $g_2(j\omega_d)$, which can meet both constraints of (6.153) and (6.154), for plant i and column c. Given that the plant is 2×2, there will be two concentric "stay-in" discs at the origin, and two concentric "stay-out" discs, centred at $-1/p_{22}^{i}(j\omega_d)$ (one set from each row of (6.150)). A necessary condition for the existence of a $g_2(j\omega_d)$, which satisfies the four constraints in (6.150) for every plant instance, is generalised as

$$\left|\frac{1}{p_{22}}\right|_{\omega_d}^{i} \geq \left.\frac{|k_{2q}| + \beta_{1r}|p_{21}||g_1|}{\beta_{2q}|p_{22}|}\right|_{\omega_d}^{i} - \left.\frac{\beta_{1r}|1 + p_{11}g_1| - |k_{1r}|}{\beta_{2q}|p_{12}|}\right|_{\omega_d}^{k}, \ \forall q, r \in \{1, 2\},\ \forall i, k \in \{1, 2, ..., n\}. \tag{6.156}$$

Additionally, the generalised constraint on $g_1(j\omega_d)$ to ensure that the radii of the "stay-in" discs in (6.153) are positive, is

$$|1 + p_{11}g_1|_{\omega_d}^{i} \geq \left.\frac{|k_{1r}|}{\beta_{1r}}\right|_{\omega_d}^{k} i,\ \forall r \in \{1, 2\},\ \forall i \in \{1, 2, ..., n\}. \tag{6.157}$$

Note that if, at a particular phase angle of $g_1(j\omega_d)$, the radius of the "stay-in" disc is always greater than or equal to the radius of the "stay-out" disc ($r_1 \geq r_2$, $\forall |g_1(j\omega)|$), any value of $|g_1(j\omega_d)|$ will guarantee the existence of $g_2(j\omega_d)$, for the particular plant instance. By extension, if the radius of the smallest "stay-in" disc (across the enumerated plant set) is always greater than or equal to the radius of the largest "stay-out" disc (at a particular phase angle of $g_1(j\omega_d)$), any selection of $|g_1(j\omega_d)|$ will guarantee the existence of $g_2(j\omega_d)$, for the entire plant set. In other words, at certain phase angles of $g_1(j\omega)$, (6.156) may become a sufficient condition for the existence of a globally valid $g_2(j\omega)$ that satisfies (6.150). Equations (6.156) and (6.157) are applied to the enumerated plant set, $P^i(j\omega_d) \in \mathcal{P}(j\omega_d)$, at design frequency point ω_d, and the intersection of all resulting design regions will describe the global solution space of $g_1(j\omega_d)$, for the particular non plant-inverting splitting, namely $\mathbb{S}^{P}_{g_1(j\omega_d)}$. This solution space is expected to allow lower gain design in the mid to high frequencies, to the extent permitted by stability constraints.

Finding the design regions on $g_r(j\omega_d)$

Equations (6.145), (6.156), and (6.157) indirectly describe the constraints on the first controller to be designed (for example, $g_1(s)$). In order to find the explicit constraints on the feedback controller element in question, one can fix the phase of $g_1(s)$ as $\phi_1 \in [-\pi, \pi]$, and then use the method of repeatedly squaring and rearranging, until one arrives at a polynomial inequality in $|g_1(j\omega_d)|$ [109]. Explicitly, (6.145), (6.156) and (6.157) can be written generically at equality as

$$\gamma = |z_1(g_1)| + \lambda |z_2(g_1)|, \tag{6.158}$$

where γ and λ are constants, and $z_k(g_1) = z_k(|g_1|e^{j\phi_1})$ is a complex number, which behaves linearly with respect to $g_1(j\omega_d)$. Given that the magnitude of a complex number is $|z_k| = \sqrt{z_k z_k^*}$, we aim to arrive at an equation which contains only even powers of all z_k terms, which will correspond to a polynomial in $|g_1(j\omega_d)|$. Squaring both sides of (6.158) and rearranging yields

$$\gamma^2 - |z_1(g_1)|^2 - \lambda^2|z_2(g_1)|^2 = 2\lambda|z_1(g_1)||z_2(g_1)|. \tag{6.159}$$

Squaring once more generates the desired result of

$$\left(\gamma^2 - |z_1(g_1)|^2 - \lambda^2|z_2(g_1)|^2\right)^2 = 4\lambda^2|z_1(g_1)|^2|z_2(g_1)|^2. \tag{6.160}$$

The possible phase angles of $g_1(j\omega_d)$ can be swept through, across the allowable range of $[-\pi, \pi]$, thereby making (6.160) solely a function of $|g_1(j\omega_d)|$. As $z_k(g_1)$ is linear with respect to $g_1(j\omega_d)$, (6.160) will (after some rearranging) describe the quartic polynomial,

$$|g_1(j\omega_d)|^4 + a_1|g_1(j\omega_d)|^3 + a_2|g_1(j\omega_d)|^2 + a_3|g_1(j\omega_d)| + a_4 = 0, \tag{6.161}$$

where the coefficients in (6.161) are made up of elements of $P^i(j\omega_d)$, $P^k(j\omega_d)$, $\hat{P}^i(j\omega_d)$, $\hat{P}^k(j\omega_d)$, $M(j\omega_d)$, $\beta(\omega_d)$, and ϕ_1 (depending on the particular splitting). The roots of the quartic polynomial will correspond to the design boundaries on $|g_1(j\omega_d)|$, for the particular controller phase ϕ_1 and plant (or plant pairing). As the squaring process may yield roots which do not correspond to the original inequality (each squaring may double the number of roots with some of these being invalid), these roots need to be checked against the original inequality. To determine whether the boundaries are "stay-above" or "stay-below", one may need to evaluate the original inequality at a value of $|g_r(j\omega_d)|$ between two valid roots. If the result satisfies the inequality, the design region between the two roots is "stay-in", implying the adjacent regions are "stay-out", and vice versa.

Union of design regions

As the solution spaces from both the plant-inverting and non plant-inverting methods over-bound the viable design region of $g_1(j\omega_d)$, the union of the two individual solution spaces,

$$\mathbb{S}_{g_1(j\omega_d)} = \mathbb{S}^{\hat{P}}_{g_1(j\omega_d)} \cup \mathbb{S}^{P}_{g_1(j\omega_d)}, \tag{6.162}$$

can only enlarge the attainable design region of $g_1(j\omega_d)$. Applying the union operation at all discrete design frequency points also has the potential benefit of expanding the design regions at the mid frequencies (relative to current plant-inverting methods). This is especially important for efficient design in the gain-phase crossover region. The global solution space of (6.162) can be visualised in the log-polar plane, and used to design a $g_1(s)$ which satisfies all the discrete tracking requirements,

$$g_1(j\omega_d) \in \mathbb{S}_{g_1(j\omega_d)}, \ \forall\omega_d \in \Omega. \tag{6.163}$$

It must be stressed that the existence of a non-empty $\mathbb{S}_{g_1(j\omega_d)}$, $\forall \omega_d \in \Omega$, does not guarantee that a low-order, proper, rational, causal $g_1(s)$ can be designed which simultaneously meets all the discrete design frequency requirements (i.e. joining the dots may not be possible/feasible!).

Closure of the second feedback loop

With the first loop closed ($g_1(s)$ for example), the design of $g_2(s)$ can proceed. Again, using the constraint in (6.145), one can arrive at a global solution space for $g_2(j\omega_d)$, for the particular plant-inverting splitting, namely $\mathbb{S}^{\hat{P}}_{g_2(j\omega_d)}$. Note that this is a non-sequential approach, which does not take into account the selection of $g_1(s)$. Given that the non plant-inverting method is sequential, the order in which the controller elements are designed does affect the overall system performance. The preferred order depends on (i) the closed-loop bandwidth of each loop; (ii) the element-wise uncertainty; (iii) channel interactions; and (iv) the underlying tracking specifications. As such, the designer must exercise engineering understanding of the problem to select the design order for the particular problem. The full design region on $g_2(j\omega_d)$ follows as the union of the two individual regions,

$$\mathbb{S}_{g_2(j\omega_d)} = \mathbb{S}^{\hat{P}}_{g_2(j\omega_d)} \cup \mathbb{S}^{P}_{g_2(j\omega_d)}, \tag{6.164}$$

and successful design of $g_2(s)$ is necessitated by

$$g_2(j\omega_d) \in \mathbb{S}_{g_2(j\omega_d)}, \ \forall \omega_d \in \Omega, \tag{6.165}$$

assuming $g_2(s)$ is feasible and physically implementable.

Feed-forward controller design

Once $G(s)$ has been designed, $X(s)$ can be designed in earnest. To assess the effect of $X(j\omega_d)$ on $E(j\omega_d)$, the discretized matrix error equation in (6.136) can be written explicitly as

$$E^i(j\omega_d) = (I + PG)^{-1}(M - PX)\Big|^{i}_{\omega_d} = \begin{bmatrix} a_1 + a_2x_{11} + a_3x_{21} & b_1 + b_2x_{12} + b_3x_{22} \\ c_1 + c_2x_{11} + c_3x_{21} & d_1 + d_2x_{12} + d_3x_{22} \end{bmatrix}^{i}_{\omega_d}, \tag{6.166}$$

where

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}^{i}_{\omega_d} = S^i(j\omega_d)M(j\omega_d), \tag{6.167}$$

$$\begin{bmatrix} a_2 & a_3 \\ c_2 & c_3 \end{bmatrix}^{i}_{\omega_d} = \begin{bmatrix} b_2 & b_3 \\ d_2 & d_3 \end{bmatrix}^{i}_{\omega_d} = -S^i(j\omega_d)P^i(j\omega_d), \tag{6.168}$$

and $S^i(j\omega_d) = \left[I + P^i(j\omega_d)G(j\omega_d)\right]^{-1}$. Notably, $x_{11}(j\omega_d)$ and $x_{21}(j\omega_d)$ only affect the first column of (6.166), whereas $x_{12}(j\omega_d)$ and $x_{22}(j\omega_d)$ only affect the second column. Applying the original tracking specification of (6.186) to the first column of (6.166) gives the two constraints of

$$\begin{bmatrix} |a_1 + a_2x_{11} + a_3x_{21}| \\ |c_1 + c_2x_{11} + c_3x_{21}| \end{bmatrix}^i_{\omega_d} \leq \begin{bmatrix} \beta_{11}(\omega_d) \\ \beta_{21}(\omega_d) \end{bmatrix}. \tag{6.169}$$

Next, dividing the first row of (6.169) by $|a_3^i(j\omega_d)|$ and the second row by $|c_3^i(j\omega_d)|$ gives

$$\begin{bmatrix} |(a_1 + a_2x_{11})/a_3 + x_{21}| \\ |(c_1 + c_2x_{11})/c_3 + x_{21}| \end{bmatrix}^i_{\omega_d} \leq \begin{bmatrix} \beta_{11}/|a_3| \\ \beta_{21}/|c_3| \end{bmatrix}^i_{\omega_d}. \tag{6.170}$$

The admissible design regions of $x_{21}(j\omega_d)$ in (6.170), with respect to the constraints on plant i, are two "stay-in" discs, where the two centres are at $-(a_1+a_2x_{11})/a_3\big|^i_{\omega_d}$ and $-(c_1+c_2x_{11})/c_3\big|^i_{\omega_d}$, and the corresponding radii are $\beta_{11}(\omega_d)/|a_3^i(j\omega_d)|$ and $\beta_{21}(\omega_d)/|c_3^i(j\omega_d)|$ respectively. Simultaneously satisfying both constraints requires a non-empty intersection of the two "stay-in" discs. As in the case of (6.145) and Figure 6.29, this requires that the distance between the two centres must be smaller than or equal to the sum of the respective radii,

$$|(a_1 + a_2x_{11})/a_3 - (c_1 + c_2x_{11})/c_3|^i_{\omega_d} \leq \beta_{11}(\omega_d)/|a_3^i(j\omega_d)| + \beta_{21}(\omega_d)/|c_3^i(j\omega_d)|. \tag{6.171}$$

Satisfying the inequality in (6.171) will guarantee the existence of a non-empty solution space for $x_{21}(j\omega_d)$, which can achieve (6.170) for plant instance i. Given that plant pair $\{i, k\}$ will collectively generate four discoidal solution spaces on $x_{21}(j\omega_d)$, based on (6.170), the set of necessary conditions for the existence of a common $x_{21}(j\omega_d)$ which can simultaneously satisfy all four design regions, is

$$\left|(a_1^i + a_2^ix_{11})/a_3^i - (c_1^k + c_2^kx_{11})/c_3^k\right|_{\omega_d} \leq \beta_{11}(\omega_d)/|a_3^i(j\omega_d)| + \beta_{21}(\omega_d)/|c_3^k(j\omega_d)|, \tag{6.172}$$

$$\left|(a_1^i + a_2^ix_{11})/a_3^i - (a_1^k + a_2^kx_{11})/a_3^k\right|_{\omega_d} \leq \beta_{11}(\omega_d)\left(1/|a_3^i(j\omega_d)| + 1/|a_3^k(j\omega_d)|\right), \tag{6.173}$$

$$\left|(c_1^i + c_2^ix_{11})/c_3^i - (c_1^k + c_2^kx_{11})/c_3^k\right|_{\omega_d} \leq \beta_{21}(\omega_d)\left(1/|c_3^i(j\omega_d)| + 1/|c_3^k(j\omega_d)|\right). \tag{6.174}$$

Equations (6.172), (6.173), and (6.174) describe a set of discoidal "stay-in" design regions on $x_{11}(j\omega_d)$ for plant pair $\{i, k\}$. Applying (6.172), (6.173), and (6.174) to the enumerated plant set ($\{i, k\} \in \{1, 2, ..., n\}$), and then taking the intersection of the resulting design regions, describes the global solution space of $x_{11}(j\omega_d)$, namely $\mathbb{S}_{x_{11}(j\omega_d)}$. Visualisation of $\mathbb{S}_{x_{11}(j\omega_d)}$ and design of $x_{11}(s)$ can take place in the log-polar plane, and upon finding an $x_{11}(s)$ which satisfies (6.172), (6.173), and (6.174), for every ordered plant pair, the corresponding discretized design regions of $x_{21}(s)$ will form in the arithmetic-complex plane, based on (6.170) (if it exists). Similarly, the necessary set of constraints on $x_{12}(j\omega_d)$ governing the existence of an $x_{22}(j\omega_d)$ that satisfies the tracking requirements of plant pair $\{i, k\}$, is

$$\left|(b_1^i + b_2^ix_{12})/b_3^i - (d_1^k + d_2^kx_{12})/d_3^k\right|_{\omega_d} \leq \beta_{12}(\omega_d)/|b_3^i(j\omega_d)| + \beta_{22}(\omega_d)/|d_3^k(j\omega_d)|. \tag{6.175}$$

$$\left|(b_1^i + b_2^i x_{12})/b_3^i - (b_1^k + b_2^k x_{12})/b_3^k\right|_{\omega_d} \leq \beta_{11}(\omega_d) \left(1/|b_3^i(j\omega_d)| + 1/|b_3^k(j\omega_d)|\right), \tag{6.176}$$

$$\left|(d_1^i + d_2^i x_{12})/d_3^i - (d_1^k + d_2^k x_{12})/d_3^k\right|_{\omega_d} \leq \beta_{21}(\omega_d) \left(1/|d_3^i(j\omega_d)| + 1/|d_3^k(j\omega_d)|\right). \tag{6.177}$$

The design of $x_{11}(s)$ and $x_{12}(s)$ takes place independently, and upon adhering to the aforementioned constraints, allows for the subsequent design of $x_{21}(s)$ and $x_{22}(s)$ (if the corresponding solution space is non-empty). It is worth emphasizing that the resulting design regions on $x_{rc}(j\omega_d)$ do not possess any design conservatism as a result of over-bounding. As in the case of designing $G(s)$, it may not be feasible/possible to find a $X(s)$ which can simultaneously meet all the discrete design specifications, as $X(s)$ must obey the Bode gain-phase constraints. Note that by dividing the first row of (6.169) by $|a_2^i(j\omega_d)|$ and the second row by $|c_2^i(j\omega_d)|$, one can arrive at constraints on $x_{21}(j\omega_d)$, instead of $x_{11}(j\omega_d)$. In other words, the designer can choose the order in which the elements of $X(s)$ are designed. As in the sequential non plant-inverting design methodology of $G(s)$, the design order of $X(s)$ must be selected by the designer, based on engineering understanding of the problem.

Section 6.5.4 describes how bounds on $x_{rc}(s)$ are obtained sequentially. Because $X(s)$ in Figure 6.28 is not part of the feedback structure, if pre-view of $r(t)$ is available (e.g. following a known trajectory), in principle $X(s)$ does not need to be a causal function. One might then instead design the signal $v(s) = X(s)r(s)$ directly. This will not be provided in detail here.

6.5.5 Worked example

In order to demonstrate the efficacy of the proposed method, we apply our design routine to a simple problem [115], with an existing (recently published) solution [5]. The uncertain plant is

$$\mathbf{P}(s) = \frac{1}{s}\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}, \ \{k_{11}, k_{22}\} \in [2, 6], \ \{k_{12}, k_{21}\} \in [0.5, 1.5]. \tag{6.178}$$

The corresponding reference model and tracking error bounds, in the frequency band of $\omega \leq 10$ rad/s, are given as

$$M(s) = \operatorname{diag}\left\{\frac{1}{1+s/3}, \frac{1}{1+s/3}\right\}, \ \beta_{rc}(\omega) = 0.2\omega\sqrt{1+\omega^2/9}. \tag{6.179}$$

Additionally, the high frequency sensitivity constraint ($\omega \geq 10$ rad/s) is

$$\left.\frac{1}{|1 + g_{rr}/\hat{p}_{rr}|}\right|_{\omega_d}^{i} \leq 1.67, \ \forall i = \{1, 2, ..., N\}. \tag{6.180}$$

The plant-inverting design methods in [115] and [5] resulted in highly conservative mid- to high-frequency constraints on the feedback controllers, meaning that a universal sensitivity constraint was never required (low sensitivity was implicitly enforced by the conservative tracking boundaries at the chosen design frequencies).

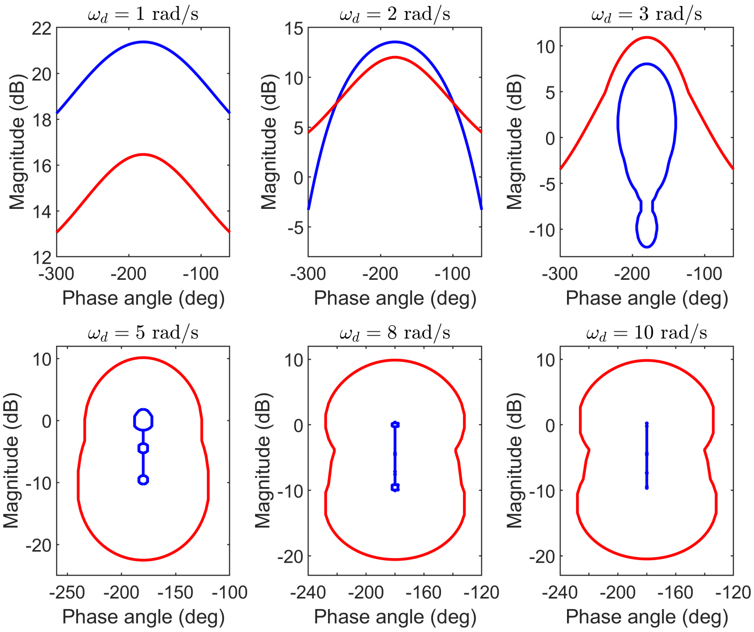


Figure 6.31: Log-polar plot showing per-frequency "stay-out" tracking design regions resulting from the plant-inverting method (red), and non plant-inverting method (blue), for $l_{11}^0(j\omega)$, and $\Omega = \{1, 2, 3, 5, 8, 10\}$ rad/s.

As the proposed non plant-inverting design routine is expected to yield less conservative design regions on the feedback controller at the mid to high frequencies, the sensitivity specification from (**??**) is adjusted to cover the entire frequency spectrum. With reference to Section 6.5.4, two sets of design regions on the first feedback controller ($g_1(s)$ in this case) are generated. In the case of the non plant-inverting method, the optimised frequency response of $X(j\omega_d)$, at each discrete design frequency point, is shown in Table 6.2. Note that the symmetry of the plant and specifications set results in a symmetrical $X(j\omega_d)$ at each discrete design frequency point. Figure 6.31 shows the resulting tracking design regions, by individually applying the aforementioned methods, on (arbitrarily selected) nominal loop transfer $l_{11}^0(s) = p_{11}^0(s)g(s)$, for 256 plant cases, where $p_{11}^0(s) = 2/s$. Note that the choice of nominal plant $p_{11}^0(s)$ has no effect on the design region of $g_1(s)$, and is only required for producing the traditional Nichols Chart visualisation. As expected, the plant-inverting method produces lower gain requirements on $g_1(s)$ at low frequencies, whereas the non plant-inverting method performs better at higher frequencies (including the gain-phase crossover region). In fact, the "stay-out" regions at $\omega_d \geq 8$ rad/s, from the non plant-inverting method, are barely visible (as a result of the ease of meeting the large high frequency tracking tolerance specified). The union of the design regions in Figure 6.31 will then

ω_d	1	2	3	5	8	10
$x_{11}(j\omega_d), x_{22}(j\omega_d)$	0.0800 + 0.2400i	0.2049 + 0.3074i	0.3388 + 0.3389i	0.5882 + 0.3529i	0.7013 + 0.2630i	0.6023 + 0.1807i
$x_{12}(j\omega_d), x_{21}(j\omega_d)$	-0.0142 - 0.0426i	-0.0615 - 0.0923i	-0.1000 - 0.1000i	-0.0947 - 0.0568i	-0.1135 - 0.0426i	-0.1171 - 0.0351i

Table 6.2: Optimised selection of $X(j\omega_d)$ for non plant-inverting method, evaluated at $\omega_d \in \{1, 2, 3, 5, 8, 10\}$ rad/s, over 256 plant cases.

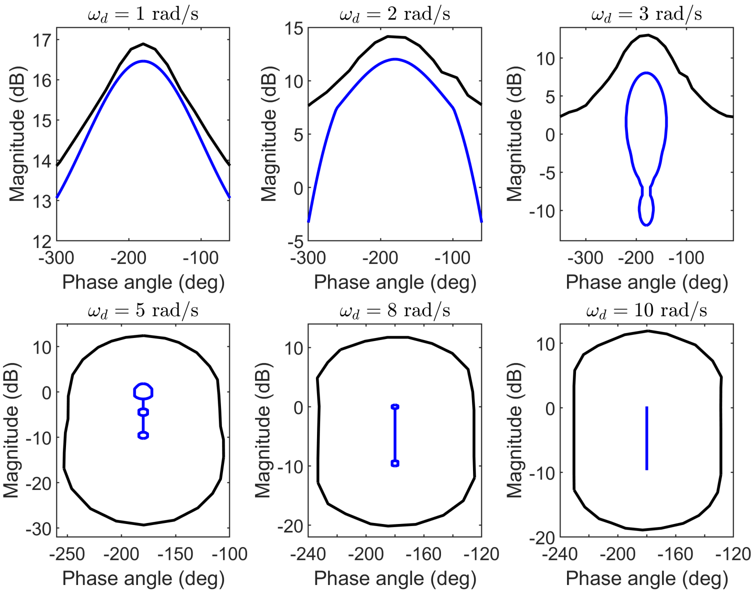


Figure 6.32: Log-polar plot showing per-frequency "stay-out" tracking design regions resulting from proposed complementary design method (blue), and method detailed in [5] (black), for $l^0_{11}(j\omega)$, and $\Omega = \{1, 2, 3, 5, 8, 10\}$ rad/s.

describe the expanded design region of the proposed complementary design method. Figure 6.32 compares this resulting design region with the method in [5]. The design boundaries of the proposed method at $\omega_d = 1$ rad/s are similar to that of [5], but the significant boundary differences can be seen at the higher frequencies, that is, at and above the gain crossover frequency, where the conservatism can be most problematic. With reference to (6.157), the "stay-out" region on $g_1(j\omega_d)$ will approach the set of plant dependent critical points, for sufficiently high frequencies, which can be seen as a vertical line in Figure 6.31 and Figure 6.32 (in stark contrast to the high frequency design regions resulting from plant-inverting methods). As is evident from the lower gain requirements in Figure 6.32, the proposed method outperforms the method in [5] at all design frequencies and feedback controller phase angles.

The resulting union of tracking boundaries on $g_1(s)$, as well as the adjusted sensitivity requirements from (6.180) are shown in Figure 6.33. A viable $g_1(s)$, which meets all the design requirements, is also shown in Figure 6.33 (after shifting and scaling by $p^0_{11}(s) = 2/s$), and is described by

$$g_1(s) = \frac{4\left(1 + \dfrac{s}{11.26}\right)}{\left(1 + \dfrac{s}{2.173}\right)\left(1 + 2\dfrac{0.32s}{25.20} + \dfrac{s^2}{25.20^2}\right)}. \qquad (6.181)$$

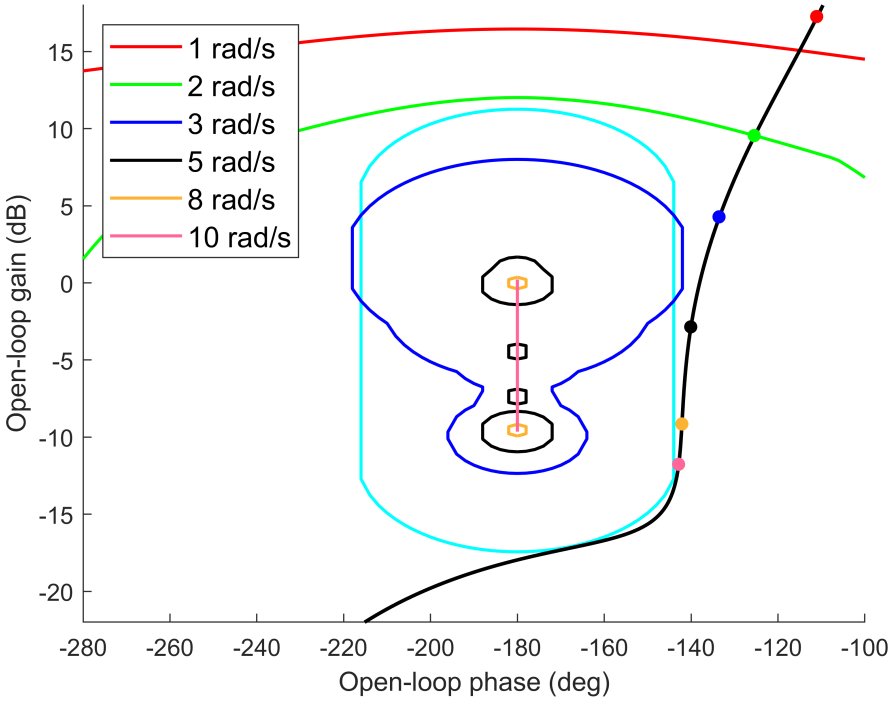


Figure 6.33: Log-polar plot, showing the union of "stay-out" tracking bounds for discrete design frequency range $\Omega = \{1, 2, 3, 5, 8, 10\}$ rad/s, as well as universal "stay-out" sensitivity bound (cyan), on $l_{11}^0(s)$. The designed $l_{11}^0(s)$ is also shown.

With reference to Figure 6.33, the universal sensitivity bound dominates the tracking bounds for $\omega > 3$ rad/s, resulting in over-design in the mid to high frequencies (as a result of the poor compatibility between the tracking and sensitivity specifications). Over-design at $\omega_d = 1$ rad/s is required to accommodate the low-frequency tracking requirements for $\omega \leq 0.1$ rad/s (not shown in Figure 6.33). Following the successful design of $g_1(s)$, the design of $g_2(s)$ proceeds. Figure 6.34 shows the union of tracking bounds as well as the sensitivity constraint for $g_2(s)$. The designed $g_2(s)$ is also shown in Figure 6.34 (after shifting and scaling by $p_{22}^0(s) = 2/s$), and is described by

$$g_2(s) = \frac{4\left(1 + \dfrac{s}{6.48}\right)}{\left(1 + \dfrac{s}{1.59}\right)\left(1 + 2\dfrac{0.49s}{23.41} + \dfrac{s^2}{23.41^2}\right)}. \tag{6.182}$$

As in the case of Figure 6.33, the tracking bounds in Figure 6.34 are dominated by the sensitivity constraint for $\omega \geq 3$ rad/s, resulting in over-design in the gain-phase crossover region ($g_2(s)$ has to roll off later to accommodate the universal sensitivity constraint). With the feedback design complete, the last step is to design the feed-forward filter $X(s)$. With reference to Section 6.5.4, one can choose the order in which the elements of $X(s)$ are to be designed. The diagonal elements of $X(s)$ are designed first, after which the off-diagonal terms can be designed. The resulting design boundaries on $X(s)$ are shown in Figure 6.35. Given the control structure

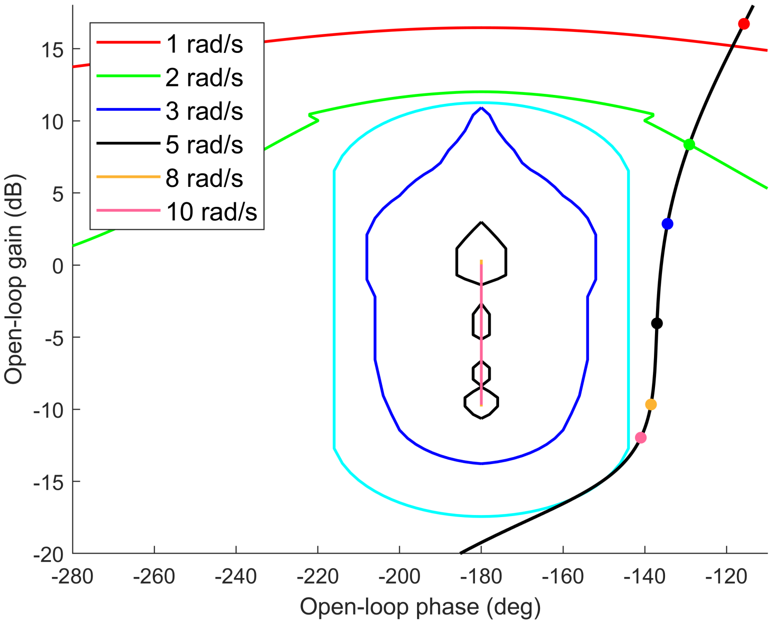
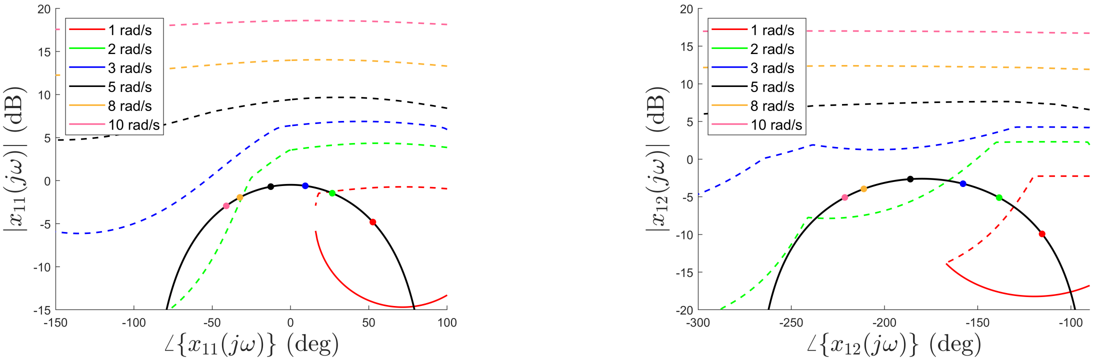


Figure 6.34: Log-polar plot, showing the union of "stay-out" tracking bounds for discrete design frequency range $\Omega = \{1, 2, 3, 5, 8, 10\}$ rad/s, as well as universal "stay-out" sensitivity bound (cyan), on $l_{22}^0(s)$. The designed $l_{22}^0(s)$ is also shown.

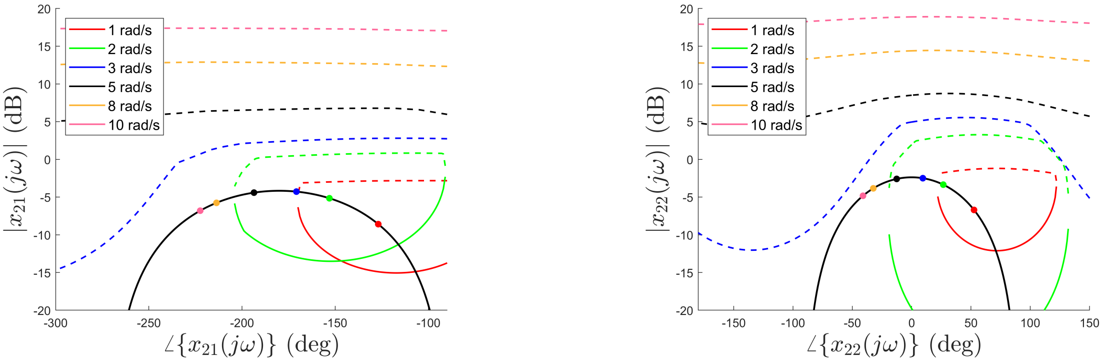
in Figure 6.28, a subtle condition on $X(s)$ to achieve zero-error tracking at steady-state, is for $X(s)$ to have zero DC gain (requiring derivative action). This is analogous to $F(s)$ requiring unitary DC gain in Figure 6.27. An appropriate $X(s)$, which meets all the design constraints, is

$$X(s) = \begin{bmatrix} \dfrac{0.65s}{(1+s/1.70)(1+s/8.19)} & \dfrac{-0.47s}{(1+s/3.76)(1+s/5.32)} \\ \dfrac{-0.41s}{(1+s/1.77)(1+s/7.68)} & \dfrac{0.53s}{(1+s/1.70)(1+s/8.19)} \end{bmatrix}, \tag{6.183}$$

and is also shown in Figure 6.35. The ease of meeting the per-frequency design boundaries in Figure 6.35 (especially at the mid to high frequencies) is attributed to the over-design of the feedback controller, resulting from the dominant sensitivity specifications in Figure 6.33 and Figure 6.34. Successful design of $G(s)$ and $X(s)$ is validated by plotting the magnitude of the set of closed-loop error frequency responses against the error tolerances from (6.230), as shown in Figure 6.36. It is worth pointing out that the designed feedback controllers in (6.232) and (6.182) are of the same order as that of the feedback solution in [5], and possess equivalent or lower gain at all frequencies. With reference to Figure 6.36, the amount of over-design is most noticeable at the mid to high frequencies, as a result of the sensitivity requirements dominating the tracking requirements in that frequency band. The compatibility between the enumerated plant set and the prescribed tracking specifications will also naturally affect the amount of potential over-design at each design frequency [73]. This is analogous to the poor compatibility between the sensitivity and tracking bounds (the large tracking error tolerances were



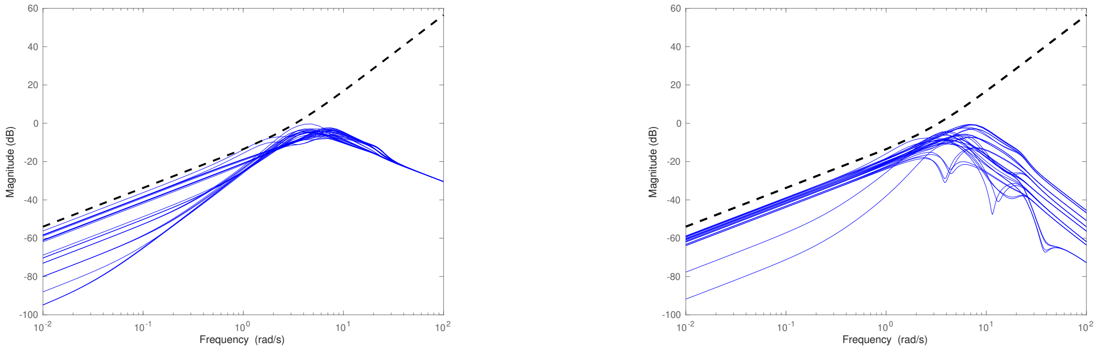
(a) $x_{11}(s)$ (b) $x_{12}(s)$



(c) $x_{21}(s)$ (d) $x_{22}(s)$

Figure 6.35: Log-polar plane showing design boundaries on $X(s)$, for discrete design frequency set $\Omega = \{1, 2, 3, 5, 8, 10\}$ rad/s. Solid lines indicate "stay-above" bounds, whereas dashed lines indicate "stay-below" bounds. The designed feed-forward controllers for the respective design conditions are also shown.

easy to achieve in the gain-phase crossover region using the non plant-inverting design method, which was previously not possible using plant-inverting methods).



(a) $\mathbf{e}_{11}(s)$ and $\mathbf{e}_{22}(s)$ (b) $\mathbf{e}_{12}(s)$ and $\mathbf{e}_{21}(s)$

Figure 6.36: Bode magnitude plots, comparing closed-loop error tracking behaviour (solid blue lines) with adjusted error tolerances (dashed black line). (a) Diagonal tracking behaviour. (b) Non-diagonal tracking behaviour.

6.5.6 Conclusion

This Section has presented a complementary QFT design method, intended for 2x2 systems, which is capable of reducing feedback control over-design across the entire frequency range of interest. In particular, the combination of a plant-inverting and non plant-inverting bound generation routine reduces the conservatism at the gain-phase crossover region, allowing for more efficient feedback control design. Following the successful feedback controller design, a non-diagonal feed-forward filter was designed, without requiring over-bounding approximations. This method has been applied to a simple, but suitably coupled 2x2 system, and was shown to outperform pre-existing methods, especially at higher frequencies of interest.

6.6 A refinement approach to the multivariable tracking error problem

6.6.1 Abstract

This Section presents a new refinement design routine aimed at solving the robust multivariable tracking error problem with reduced conservatism. The prevailing multivariable Quantitative Feedback Theory (QFT) approach is to implicitly overbound the model-error tracking set frequency response in magnitude using the triangle inequality, with the intention of arriving at a set of univariate design constraints that independently describe the feedback controller solution space. While this method is effective in the low-frequency range (where the loop gain tends to be large), the mid- to high-frequency design regions (where diagonal dominance is not possible in general) suffers from arbitrarily large design conservatism. This inhibits minimum gain-phase solutions and necessitates undesirable over-design in the frequency band that can contribute to large, expensive control action. The proposed method follows a refinement approach that makes use of information from an *a priori* feedback control design. In this way, the tracking error problem is reposed as a differential design, and gain-phase information from the previous iteration can be captured to reduce the conservatism imposed when applying the triangle inequality. Additionally, a constrained optimisation routine is used to select a prototype feedforward filter that can relax the constraint set, thereby increasing the accessible solution space of the diagonal feedback controller. Finally, a nondiagonal, multivariable feedforward filter bound generation routine is defined that relies on existence conditions and is free of induced design conservatism. The viability of this design methodology is demonstrated on two benchmark problems of varying complexity, in order to demonstrate the widespread efficacy.

6.6.2 Introduction

Within the QFT research sphere, the multivariable tracking problem has been met with varying efficacy [111]. Broadly speaking, the two approaches are to either (i) apply above/below tracking magnitude bounds [82], or (ii) use tracking error bounds [112]. Research that falls under (i) has benefit in being computationally simpler, but commonly requires a diagonal pre-filter to facilitate the design. Work in [116] uses the classical approach of under- and overbounding the closed-loop behaviour to design a diagonal pre-filter and feedback controller for a 2×2 heat exchanger. Additionally, the way in which tracking tolerances couple across the design loops is exploited to tighten constraints on a particular loop whilst relaxing the constraints in the other loops. This has the benefit that, when chosen appropriately, non-dominant boundaries in one channel will become stricter, but remain non-dominant, whilst easing the dominant boundaries in the other channels. As noted in [114], there is benefit in having a fully populated pre-filter, as it can aid in the loop decoupling and reduce the burden on the feedback controller elements. Additionally, there is the potential for diminished time-domain tracking precision as a result of not capturing relative phase information between the closed-loop system and above/below specifications [13]. The more recent introduction of multivariable tracking error bounding methods has comparable magnitude boundaries on the closed-loop system, but additionally imposes differential phase constraints. A common approach is to predefine the pre-filter using information from an appropriately selected nominal plant [112]. This removes one degree of freedom from the control scheme and allows for direct design on the feedback controller elements when using plant-inverting splittings. The work in [18] additionally proposes a refinement of the pre-filter once the feedback controller has been designed, which can alleviate the burden of implementing a potentially irrational or ill-conditioned pre-filter. While this design procedure is computationally simple, the sub-optimal pre-filter synthesis can place additional design conservatism on the feedback controller. This is worsened by the fact that the nominal plant selection is not obvious for complex systems with irregular uncertainty template shapes, and this can exacerbate the contracted feedback controller solution space [20]. A less conservative routine was proposed in [5], which separates the tracking error problem into a simultaneous tracking and disturbance rejection problem using iterative bound balancing methods. This design method allows for a fully populated pre-filter, and the design regions are independent of the nominal plant selection. This method is similarly implemented for the measured disturbance rejection problem in [81]. The aforementioned design routines all rely on using a plant-inverting splitting, which can incur large design conservatism at the mid- to high-frequencies as a result of poor loop diagonal dominance [117]. More recently, a complementary design routine was proposed that combines a plant-inverting design with a non plant-inverting design [20]. This method is able to reduce design conservatism at all frequencies, especially at and above the gain-crossover frequency, but only applies to the special case of 2×2 systems.

The state of the art of multivariable QFT is fundamentally limited by having to assume a worst-case gain-phase over-bounding of the tracking error matrix when applying the triangle inequality. If knowledge of a prior valid design were known, the corresponding phase and gain information could be captured, thereby reducing the amount of overbounding required. This contribution seeks to exploit the design insight from this relationship to repose the tracking error constraint set in a form that incrementally increases the feedback controller design regions. Information from a successful feedback controller refinement is then incorporated into a subsequent feedback control design pass, which reduces the amount of conservatism on the feedback controller elements incurred when using the triangle inequality. Following the successful design of the feedback controller, the solution space of one row of the feedforward filter is ipso facto defined, based on the existential link between the feedback controller and feedforward filter. Existence conditions are then chained to sequentially design each row of the feedforward filter. While informed from the feedback controller selection, the solution space of the feedforward filter is evaluated without using overbounding and does not carry any induced conservatism. As this method is a QFT design tool that finds feasible regions at discrete design frequency points, the designer is still required to join these regions with a sufficiently low-order, causal controller, capable of achieving closed-loop plant stability. This is a generalised design method that is applicable to $n \times n$ systems, and is bench-marked using two different 2×2 plants of varying complexity, for sake of clarity and demonstration.

6.6.3 Problem statement

Consider the two degree of freedom (2DOF) control configuration of Figure 6.37, where $P^i(s) \in \mathcal{P}(s)$ is an $m \times m$ plant instance in the uncertain set $\mathcal{P}(s)$; $r(s)$ is the $m \times 1$ reference signal; and $M(s)$ is the $m \times m$ reference model that describes the ideal closed-loop transfer behaviour. Set $\mathcal{P}(s) = \{P^1(s), P^2(s), ..., P^n(s)\}$ is an *ε-net* of disjoint plants that sufficiently describes the uncertain plant. Diagonal feedback controller $G(s)$ =diag$\{g_1(s), g_2(s), ..., g_m(s)\}$ acts to reduce uncertainty across the plant set, while $m \times m$ feedforward filter $X(s)$ is designed to adjust the closed-loop bandwidth of the plant set and distribute reference signal information across the control loops.

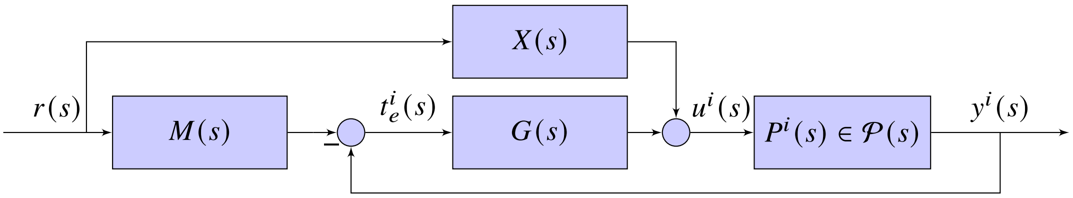


Figure 6.37: Block diagram of feedforward control scheme with two degrees of freedom, $\{G(s), X(s)\}$).

The structure in Figure 6.37 is analogous to traditional pre-filter approaches, as seen in [111] for example, with the notable benefit that it inherently captures the reference model tracking error within the loop:

$$t_e^i(s) = M(s)r(s) - y^i(s) = E^i(s)r(s), \tag{6.184}$$

where

$$E^i(s) = \left[I + P^i(s)G(s)\right]^{-1} \left(M(s) - P^i(s)X(s)\right). \tag{6.185}$$

Following standard quantitative feedback tracking error methods [112], the element-wise magnitude of the matrix reference model tracking error frequency response in (6.185) can be constrained by a defined specification,

$$|e_{ab}^i(j\omega_d)| \leq \beta_{ab}(\omega_d), \tag{6.186}$$

where $\beta_{ab}(\omega_d)$ is the user-specified reference model tracking error tolerance at row a and column b, and $\omega_d \in \Omega$ is the discrete design frequency point of interest. The design frequency set, Ω, is chosen based on engineering understanding of the control problem, and is required to be sufficiently rich in order to capture the contiguous frequency behaviour of the closed-loop system [78]. The objective is to design $G(s)$ and $X(s)$ in (6.185) in order to meet the constraints of (6.186) for all plant cases, $i \in \{1, 2, ..., n\}$. Additionally, cost-effective design requires that bandwidth minimisation and simplicity of $G(s)$ be prioritised over that of $X(s)$ [114].

6.6.4 Method

Preliminary feedback control design

A preliminary $G(j\omega)$ prototype is required in order to proceed with the refinement process in Section 6.6.4. This $G(j\omega)$ can be obtained by any means, such as synthesis, with the only requirement being that a non-empty design region of the corresponding $X(j\omega)$ exists that can satisfy (6.186).

A preliminary feedback control design is proposed here as one applicable method and follows the existential plant-inverting method first introduced in [20]. Using the frequency response definition of $s = j\omega$, the tracking error for plant instance i follows from (6.185) as

$$E^i(j\omega) = \left[I + P^i(j\omega)G(j\omega)\right]^{-1}\left[M(j\omega) - P^i(j\omega)X(j\omega)\right]. \tag{6.187}$$

Assuming the plant inverse, $P^i(j\omega)^{-1} = \hat{P}^i(j\omega)$, is non-singular at the chosen design frequencies, the conventional plant-inverting splitting approach [77] is applied

$$\left[\hat{P}^i(j\omega) + G(j\omega)\right]E^i(j\omega) = \hat{P}^i(j\omega)M(j\omega) - X(j\omega). \tag{6.188}$$

The plant inverse is then separated into diagonal and off-diagonal parts respectively, $\hat{P}^i(j\omega) = \hat{P}^i_d(j\omega) + \hat{P}^i_n(j\omega)$, and the resulting equation is rearranged to give the implicit form of

$$\hat{P}^i(j\omega)M(j\omega) - X(j\omega) - \hat{P}^i_n(j\omega)E^i(j\omega) = \left[\hat{P}^i_d(j\omega) + G(j\omega)\right]E^i(j\omega). \tag{6.189}$$

Given the element-wise matrix magnitude constraint of $|E|^i_\omega \leqslant \beta(\omega)$ from (6.186), where

$$|E|^i_\omega = \begin{bmatrix} |e^i_{11}(j\omega)| & |e^i_{12}(j\omega)| & \dots & |e^i_{1m}(j\omega)| \\ |e^i_{21}(j\omega)| & |e^i_{22}(j\omega)| & & \vdots \\ \vdots & & \ddots & \\ |e^i_{m1}(j\omega)| & \dots & & |e^i_{mm}(j\omega)| \end{bmatrix}, \tag{6.190}$$

the nonconservative tracking constraint related to (6.189) is posed as

$$\left|\hat{P}^i M - \hat{P}^i_n E^i - X\right|_\omega \leqslant \left|\hat{P}^i_d + G\right|_\omega \beta(\omega). \tag{6.191}$$

Rewriting (6.191) compactly as $|C^i - X|_\omega \leqslant R^i(\omega)$, where $C^i(j\omega) = \left[\hat{P}^i M - \hat{P}^i_n E^i\right]_\omega$ and $R^i(\omega) = \left|\hat{P}^i_d + G\right|_\omega \beta(\omega)$, the i^{th} plant solution space of $x_{ab}(j\omega)$ — the element of $X(j\omega)$ at row a and column b — is a *stay-in* disc, with a centre of $c^i_{ab}(j\omega)$, and a radius of $r^i_{ab}(\omega)$, as shown in Figure 6.38. Generalised non-singleton plant sets require that every plant-instanced *stay-in* disc of $x_{ab}(j\omega)$ has a common overlap, thereby guaranteeing that a nonempty set of $x_{ab}(j\omega)$ points exist in the arithmetic-complex plane that can

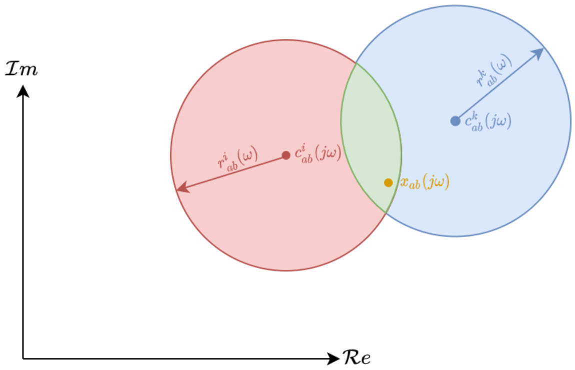


Figure 6.38: Visual representation of two discoidal *stay-in* design regions of $x_{ab}(j\omega)$ in the arithmetic-complex plane. Each plant instance generates a *stay-in* disc from (6.191), and a viable solution of $x_{ab}(j\omega)$ is required to lie within the intersection of the two discs. Existence of a nonempty intersection is predicated upon satisfying $\left|c^i_{ab} - c^k_{ab}\right|_\omega \leq r^i_{ab}(\omega) + r^k_{ab}(\omega)$.

simultaneously satisfy the constraint set in (6.191) for two different plant cases. With reference to Figure 6.38 and [4], this necessitates that the distance between any two plant-instanced centres must be smaller than or equal to the sum of their corresponding radii $\left|c^i_{ab} - c^k_{ab}\right|_\omega \leq r^i_{ab}(\omega) + r^k_{ab}(\omega)$. The element-wise matrix equivalent, $\left|C^i - C^k\right|_\omega \preccurlyeq R^i(\omega) + R^k(\omega)$, follows as

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}^k_n E^k - \hat{P}^i_n E^i\right|_\omega \preccurlyeq \left[|\hat{P}^i_d + G|_\omega + |\hat{P}^k_d + G|_\omega\right] \beta(\omega), \quad \forall i, k \in [1, n]. \tag{6.192}$$

Note that (6.192) is a necessary but not sufficient condition for an $X(j\omega)$ to exist across the entire plant set, as there is no guarantee that a common intersection exists for all *stay-in* discs. The plant-instanced tracking error matrices — $E^i(j\omega)$ and $E^k(j\omega)$ in (6.192) — are unknown prior to fixing $G(j\omega)$ and $X(j\omega)$ and this necessitates over-bounding. Applying the triangle inequality to the left-hand side of (6.192), whilst grouping all known frequency response information, results in

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}^k_n E^k - \hat{P}^i_n E^i\right|_\omega \preccurlyeq \left|\left(\hat{P}^i - \hat{P}^k\right) M\right|_\omega + \left(\left|\hat{P}^i_n\right|_\omega + \left|\hat{P}^k_n\right|_\omega\right) \beta(\omega). \tag{6.193}$$

Equation (6.193) is then used to form the conservative constraint of

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}^k_n E^k - \hat{P}^i_n E^i\right|_\omega \preccurlyeq \left|\left(\hat{P}^i - \hat{P}^k\right) M\right|_\omega + \left(\left|\hat{P}^i_n\right|_\omega + \left|\hat{P}^k_n\right|_\omega\right) \beta(\omega) \preccurlyeq \left(\left|\hat{P}^i_d + G\right|_\omega + \left|\hat{P}^k_d + G\right|_\omega\right) \beta(\omega), \tag{6.194}$$

where satisfying the right-hand inequality in (6.194) of

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M\right|_\omega + \left(\left|\hat{P}^i_n\right|_\omega + \left|\hat{P}^k_n\right|_\omega\right) \beta(\omega) \preccurlyeq \left(\left|\hat{P}^i_d + G\right|_\omega + \left|\hat{P}^k_d + G\right|_\omega\right) \beta(\omega), \quad \forall i, k \in [1, n] \tag{6.195}$$

will implicitly satisfy (6.192). The conservatism of the applied triangle inequality in (6.193) is problematic when $\left(\left|\hat{P}^i_n\right|_\omega + \left|\hat{P}^k_n\right|_\omega\right) \beta(\omega)$ is not small relative to $\left|\left(\hat{P}^i - \hat{P}^k\right) M\right|_\omega$ and results in diminished design regions for the diagonal elements of $G(j\omega)$, based on the structure in (6.195). This commonly occurs in the mid- to high-frequency band as a result of $\left|\hat{P}_n(j\omega)\right|$ rolling up while $M(j\omega)$ rolls off. Equation (6.195) is solely a function of decoupled $G(j\omega)$, and this allows for direct design of the feedback controller elements in a non-sequential manner. As detailed in [20], the per-frequency plant-instanced design regions of $G(j\omega)$ in (6.195) are *stay-out* ellipses (when represented in the arithmetic-complex plane). Note that this bound generation constraint is comparable to the proposed approach in [5], albeit with no requirement for iterative manual inspection and parameter adjustment. Additionally, because the method in [5] does not attempt to balance the boundaries at every feedback controller phase angle, there will, in general, exist phase-dependent design regions where one design constraint dominates the other, thereby inducing additional conservatism relative to (6.195).

Refining an existing design

In this approach, we assume that there is a preliminary feedback controller design, $G_o(s)$, that gives rise to nonempty design regions for the frequency response of $X_o(s)$ at each design frequency. Adherence to the tracking specifications is verified by evaluating (6.187) at $X(j\omega) = X_o(j\omega)$ and $G(j\omega) = G_o(j\omega)$:

$$E_o^i(j\omega) = (I + P^i G_o)^{-1}(M - P^i X_o)\Big|_\omega, \ \forall i \in [1,n], \tag{6.196}$$

with the requirement that $|E_o^i|_\omega \leqslant \beta(\omega), \ \forall i \in [1,n]$. The design conservatism in (6.195) is a result of having to take the worst-case gain and phase behaviour of the implicit per-plant tracking error terms when applying the triangle inequality. If prior gain and phase information of the tracking error can be extracted from a preliminary controller design using (6.196), this can be incorporated into the constraint in (6.192) before applying the triangle inequality, with the objective of reducing the amount of overbounding. Defining $E^i(j\omega) = E_o^i(j\omega) + \Delta E^i(j\omega)$, (6.192) can be written as

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}_n^k(E_o^k + \Delta E^k) - \hat{P}_n^i(E_o^i + \Delta E^i)\right|_\omega \leqslant \left(\left|\hat{P}_d^i + G\right|_\omega + \left|\hat{P}_d^k + G\right|_\omega\right)\beta(\omega), \ \forall i,k \in [1,n]. \tag{6.197}$$

While the corresponding incremental feedback design follows as $G(j\omega) = G_o(j\omega) + \Delta G(j\omega)$, boundaries are derived on absolute $G(j\omega)$ without loss of generality. Applying the triangle inequality to the left-hand side of (6.197), while retaining all known frequency response information in the first term, results in

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}_n^k(E_o^k + \Delta E^k) - \hat{P}_n^i(E_o^i + \Delta E^i)\right|_\omega \leqslant \left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}_n^k E_o^k - \hat{P}_n^i E_o^i\right|_\omega + \left|\hat{P}_n^i\right|_\omega \left|\Delta E^i\right|_\omega + \left|\hat{P}_n^k\right|_\omega \left|\Delta E^k\right|_\omega . \tag{6.198}$$

Equation 6.198 is then used to form the overbounded constraint set of

$$\left|\left(\hat{P}^i - \hat{P}^k\right) M + \hat{P}_n^k E_o^k - \hat{P}_n^i E_o^i\right|_\omega + \left|\hat{P}_n^i\right|_\omega \left|\Delta E^i\right|_\omega + \left|\hat{P}_n^k\right|_\omega \left|\Delta E^k\right|_\omega \leqslant \left(\left|\hat{P}_d^i + G\right|_\omega + \left|\hat{P}_d^k + G\right|_\omega\right)\beta(\omega), \ \forall i,k \in [1,n], \tag{6.199}$$

where (6.199) is a sufficient condition in satisfying (6.197). $\left|\Delta E^i\right|_\omega$ is unknown and can be overbounded by $\left|\Delta E^i\right|_\omega \leqslant \Delta\beta(\omega), \ \forall i \in [1,n]$, where $\Delta\beta(\omega) \geqslant 0$ is an adjustable parameter that is detailed in Section 6.6.4.

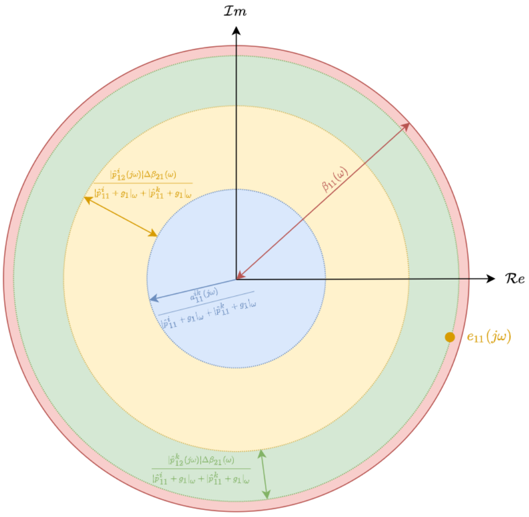


Figure 6.39: Arithmetic-complex plane showing geometric representation of the overbounded tracking error constraint from (6.203) for $\{r,c\}=\{1,1\}$ of a 2×2 system. The sum of the terms on the left-hand side of (6.203) form a disc (green) that is required to lie within the tracking tolerance (red disc).

The univariate design constraint set on $G(j\omega)$ follows as

$$\left|\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}^k_nE^k_o-\hat{P}^i_nE^i_o\right|_\omega+\left(\left|\hat{P}^i_n\right|_\omega+\left|\hat{P}^k_n\right|_\omega\right)\Delta\beta(\omega)\preccurlyeq\left(\left|\hat{P}^i_d+G\right|_\omega+\left|\hat{P}^k_d+G\right|_\omega\right)\beta(\omega),\quad\forall i,k\in[1,n].\tag{6.200}$$

While the right-hand side matrices in (6.195) and (6.200) are identical, the left-hand sides are different. Moreover, if $\Delta\beta(\omega)\preccurlyeq\beta(\omega)-\left|E^i_o\right|_\omega$, $\forall i\in[1,n]$, then the left-hand side of (6.200) will be strictly smaller than or equal to the left-hand side of (6.195):

$$\left|\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}^k_nE^k_o-\hat{P}^i_nE^i_o\right|_\omega+\left(\left|\hat{P}^i_n\right|_\omega+\left|\hat{P}^k_n\right|_\omega\right)\Delta\beta(\omega)\preccurlyeq\left|\left(\hat{P}^i-\hat{P}^k\right)M\right|_\omega+\left(\left|\hat{P}^i_n\right|_\omega+\left|\hat{P}^k_n\right|_\omega\right)\beta(\omega).\tag{6.201}$$

Denoting $A^{ik}(j\omega)=\left[\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}^k_nE^k_o-\hat{P}^i_nE^i_o\right]_\omega$ and noting that $\hat{P}^i_n(j\omega)$ is a hollow matrix, (6.200) can be represented in element-wise form as

$$\left|a^{ik}_{rc}(j\omega)\right|+\sum_{v=1,v\neq r}^{m}\left[|\hat{p}^i_{rv}|+|\hat{p}^k_{rv}|\right]_\omega\Delta\beta_{vc}(\omega)\leq\left(|\hat{p}^i_{rr}+g_r|_\omega+|\hat{p}^k_{rr}+g_r|_\omega\right)\beta_{rc}(\omega).\tag{6.202}$$

The overbounded element-wise tracking error constraint follows as

$$\frac{\left|a^{ik}_{rc}(j\omega)\right|}{|\hat{p}^i_{rr}+g_r|_\omega+|\hat{p}^k_{rr}+g_r|_\omega}+\frac{\sum_{v=1,v\neq r}^{m}\left[|\hat{p}^i_{rv}|+|\hat{p}^k_{rv}|\right]_\omega\Delta\beta_{vc}(\omega)}{|\hat{p}^i_{rr}+g_r|_\omega+|\hat{p}^k_{rr}+g_r|_\omega}\leq\beta_{rc}(\omega),\tag{6.203}$$

and is shown graphically in Figure 6.39. Dividing both sides of (6.202) by $\beta_{rc}(\omega)$ yields

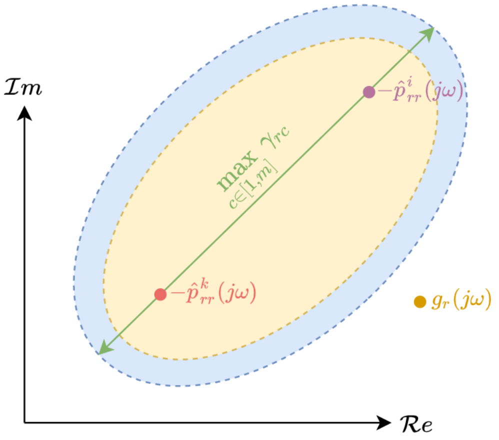


Figure 6.40: Arithmetic-complex plane showing elliptical *stay-out* design regions of $g_r(j\omega)$ for plant-pair $\{ik\}$, derived from (6.204), for a 2×2 plant. A valid $g_r(j\omega)$ is required to lie outside the union of the two ellipses with coincident foci, one resulting from each column in (6.200), which is equivalent to lying outside the larger blue ellipse. The blue ellipse is obtained directly from (6.205).

$$\frac{\left|a^{ik}_{rc}(j\omega)\right|}{\beta_{rc}(\omega)} + \frac{\sum_{v=1,v\neq r}^{m} \left[|\hat{p}^i_{rv}| + |\hat{p}^k_{rv}|\right]_\omega \Delta\beta_{vc}(\omega)}{\beta_{rc}(\omega)} \leq |\hat{p}^i_{rr} + g_r|_\omega + |\hat{p}^k_{rr} + g_r|_\omega, \tag{6.204}$$

and geometrically describes an elliptical *stay-out* design region of $g_r(j\omega)$ for plant pair $\{i, k\}$, with foci at $-\hat{p}^i_{rr}(j\omega)$ and $-\hat{p}^k_{rr}(j\omega)$, and a corresponding major axis of $\gamma_{rc}(\omega) = \frac{\left|a^{ik}_{rc}(j\omega)\right|}{\beta_{rc}(\omega)} + \frac{\sum_{v=1,v\neq r}^{m} \left[|\hat{p}^i_{rv}|+|\hat{p}^k_{rv}|\right]_\omega \Delta\beta_{vc}(\omega)}{\beta_{rc}(\omega)}$, as seen in Figure 6.40. As such, reducing the left-hand side of (6.204) will monotonically expand the admissible design region of $G(j\omega)$. There are two mechanisms that can be used to achieve this, and these are detailed in Sections 6.6.4 and 6.6.4. With reference to Figure 6.40, feedback controller element $g_r(j\omega)$ will have m ellipses with coincident foci. As such, the union of the *stay-out* ellipses is equivalent to the ellipse with the largest major axis,

$$\max_{c\in[1,m]} \gamma_{rc} \leq |\hat{p}^i_{rr} + g_r|_\omega + |\hat{p}^k_{rr} + g_r|_\omega. \tag{6.205}$$

Bounding the incremental error

As introduced in (6.200), $\left|\Delta E^i\right|_\omega \preccurlyeq \Delta\beta(\omega)$ serves as an overbounding on the incremental error allowed, where the current error is $E^i_o(j\omega)$. It is tempting to reduce $\Delta\beta(\omega)$ in order to minimise the left-hand side in (6.200), but this places additional design strictures on $\Delta G(j\omega)$ that are not immediately apparent. In essence, constraining $\Delta\beta(\omega)$ implicitly places constraints on $\Delta G(j\omega)$ that must be adhered to in order to not violate (6.197). An extreme example would be that if $\Delta\beta(\omega) = 0$, then strictly speaking, $\Delta G(j\omega)$ would have to be selected such that $|\Delta E^i(\omega)| = 0,\ \forall i$, implying that there may be no design flexibility available.

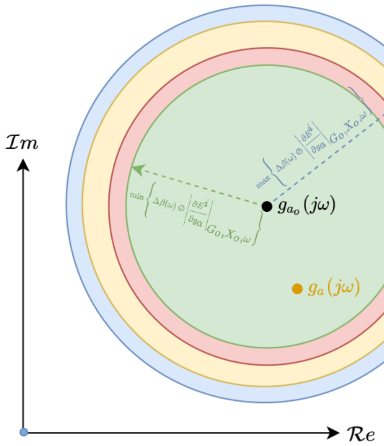


Figure 6.41: Geometric representation of (6.207) in the arithmetic-complex plane for an arbitrary 2 × 2 plant. The local solution space of $g_a(j\omega)$ for plant instance i is a *stay-in* disc that is dictated by the smallest (green) of the concentric discs centred at $g_{a_o}(j\omega)$. An $m \times m$ plant will result in m^2 concentric discs.

To gain a handle on the incremental feedback controller change allowed, first-order Taylor series expansion is used to approximate the incremental error constraint as

$$\left|\Delta E^i\right| \approx \left|\left.\frac{\partial E^i}{\partial g_a}\right|_{G_o,X_o}\left(g_a - g_{a_o}\right)\right|_{\omega} \preccurlyeq \Delta\beta(\omega),\ \forall a \in [1, m], \tag{6.206}$$

where $g_{a_o}(j\omega)$ is the preliminary, known feedback controller design for row a, and $g_a(j\omega)$ is the corresponding, yet to be designed, updated feedback controller element. The structure in (6.206) allows for element-wise division by $\left.\frac{\partial E^i}{\partial g_a}\right|_{G_o,X_o,\omega}$. Denoting $K_a^i(\omega) = \Delta\beta(\omega) \oslash \left|\frac{\partial E^i}{\partial g_a}\right|_{G_o,X_o,\omega}$, the constraint set on $g_a(j\omega)$ for plant instance i can be reposed as

$$\left|g_a - g_{a_o}\right|_{\omega} \leq \min_{r,c\in[1,m]} k_{a_{rc}}^i(\omega), \tag{6.207}$$

where $\oslash$ is the Hadamard division operator, and $k_{a_{rc}}^i(\omega)$ is the (r, c) element of $K_a^i(\omega)$. With reference to Figure 6.41, (6.207) describes a *stay-in* disc for $g_a(j\omega)$, with a centre at $g_{a_o}(j\omega)$ and a radius equal to the smallest element in $m \times m$ matrix $K_a^i(\omega)$ (as the intersection of the concentric *stay-in* discs will be equivalent to the disc with the smallest radius). In this way, the divergence of $g_a(j\omega)$ from the preliminary $g_{a_o}(j\omega)$ is dictated by $\Delta\beta(\omega)$ (the incremental error magnitude allowed) and the element-wise reciprocal of $\left|\frac{\partial E^i}{\partial g_a}\right|_{G_o,X_o,\omega}$ (the sensitivity of the model tracking error with respect to the feedback controller element). Highly sensitive model tracking errors will then only allow small feedback controller adjustments and insensitive model tracking behaviour will allow for larger controller gain/phase adjustments. $\Delta\beta(\omega)$ terms also appears on the left-hand side of (6.204), thereby enforcing an inherent trade-off between increasing the discoidal *stay-in* design regions of $g_a(j\omega)$ in (6.207) (by increasing $\Delta\beta(\omega)$), and decreasing the elliptical *stay-out* design regions from (6.204)

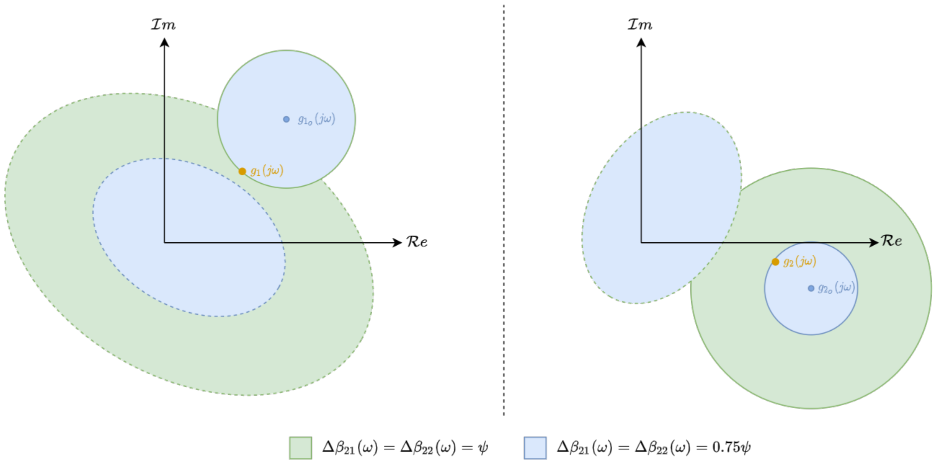


Figure 6.42: An example of the feedback controller design trade-off between $g_1(j\omega)$ (left) and $g_2(j\omega)$ (right), for a 2×2 plant, as a result of altering elements in the second row of $\Delta\beta(\omega)$. A valid feedback controller solution is required to lie outside of the ellipses from (6.204) (dashed lines) and within the discs from (6.207) (solid lines). Reducing $\Delta\beta_{21}(\omega)$ or $\Delta\beta_{22}(\omega)$ will reduce the size of the *stay-out* ellipses for $g_1(j\omega)$ based on (6.204) and Figure 6.40, but this may have the effect of reducing the discoidal *stay-in* design region of $g_2(j\omega)$ based on (6.207). Note that the *stay-in* discs of $g_1(j\omega)$ and *stay-out* ellipses of $g_2(j\omega)$ do not change as $\Delta\beta_{2k}(\omega)$ is reduced (the respective design regions are coincident).

(by decreasing $\Delta\beta(\omega)$). As evidenced in (6.204), the structure of hollow matrix $\left|\hat{P}^i_n\right|$ in (6.200) prevents row a elements of $\Delta\beta(\omega)$ from transitioning into row a of (6.200). This suggests that the designer can adjust specific elements of $\Delta\beta(\omega)$ to trade off design freedom between the different channels. This highlights an appealing aspect of this design approach — while the bound generation method is technically non-sequential, adjusting $\Delta\beta(\omega)$ enables prioritisation of one loop over others on a per-frequency basis, which is made transparent by visualising the feedback controller design regions in the log-polar plane. An example of the design region trade-off for a 2×2 system is shown in Figure 6.42. This method is conceptually similar to that of [116], albeit framed in a model-error tracking context, with *a priori* gain-phase information also included in the updated design equations. Nominal selection of $\Delta\beta(\omega)$ is assisted by the tracking tolerance, $\beta(\omega)$, namely

$$\Delta\beta(\omega) = \lambda(\omega) \odot \beta(\omega), \tag{6.208}$$

where $\lambda(\omega) \preccurlyeq 1$ is a nonnegative matrix that is used to adjust element-wise values of $\Delta\beta(\omega)$, and $\odot$ is the Hadamard product operator. Note that $\beta(\omega)$ is used in the above formulation to provide appropriate weighting in terms of the original tracking specifications. $\lambda(\omega)$ is used to refine and trade the design regions across the various control loops. As an example, when designing $g_1(j\omega)$, one can shrink the *stay-out* ellipses from (6.204) by reducing $\lambda_{ab}(\omega), a \neq 1$. However, the inherent reduction of $\Delta\beta_{ab}(\omega)$ means that the stay-in design region from (6.207) is likely to also shrink, thereby limiting the incremental design space of $g_2(j\omega)$. The designer is then tasked with finding a balance between the *stay-in* and *stay-out* design regions, which is facilitated by adjusting $\lambda(\omega)$ and then visualising the resulting design region in the log-polar plane.

Synthesizing $X_o(j\omega)$

Given that the magnitude of $\left|a_{rc}^{ik}(j\omega)\right|$ in (6.204) contributes to the size of the *stay-out* design regions, it is appealing and conceivable to minimise the worst-case magnitude of $\left|a_{rc}^{ik}(j\omega)\right|$ over all plant cases, which is equivalent to minimising the first term in (6.200), $\left|\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}_n^kE_o^k-\hat{P}_n^iE_o^i\right|_\omega$, by adjusting $E_o(j\omega)$ using $X_o(j\omega)$ from (6.196). This would have the effect of increasing the admissible design region of $G(j\omega)$ constrained by (6.200), by reducing the major axes of the *stay-out* ellipses in (6.204). Setting $E_o^i(j\omega) = V_o^i(j\omega) - Z_o^i(j\omega)X_o(j\omega)$, (6.196) can be used to rewrite the first term of (6.200) as

$$\left|\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}_n^kE_o^k-\hat{P}_n^iE_o^i\right|_\omega=\left|\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}_n^k\left(V_o^k-Z_o^kX_o\right)-\hat{P}_n^i\left(V_o^i-Z_o^iX_o\right)\right|_\omega, \tag{6.209}$$

where $V_o^i(j\omega)=\left[I+P^i(j\omega)G_o(j\omega)\right]^{-1}M(j\omega)$, and $Z_o^i(j\omega)=\left[I+P^i(j\omega)G_o(j\omega)\right]^{-1}P^i(j\omega)$. The overarching minimax problem is to find a prototype $X_o(j\omega)$ such that (6.209) is element-wise minimised (in some sense), subject to the tracking error constraint in (6.186). One could also include the diagonal elements of $G_o(j\omega)$ as part of this free parameter set, but this has the risk of yielding a poorly-posed cost function. Denoting $Q_o^{ik}(j\omega)=\left(\hat{P}^i-\hat{P}^k\right)M+\hat{P}_n^kV_o^k-\hat{P}_n^iV_o^i$, and $R_o^{ik}=\hat{P}_n^kZ_o^k-\hat{P}_n^iZ_o^i$, the modulus of the (a,b) element in (6.209) is $\left|q_{ab}^i-\sum_{d=1}^m r_{ad}^i x_{db}\right|_\omega$. It follows that all the elements in column b of $X_o(j\omega)$ will appear in each element of column b of the set of resulting constraints (we cannot optimise for each element of $X_o(j\omega)$ individually). The adjusted minimax problem that finds all elements of $X_o(j\omega)$ in column b is

$$\{x_{1b}(j\omega),x_{2b}(j\omega),...,x_{mb}(j\omega)\}=\min\left\{\max_{i\in[1,n]}\left\{\sum_{a=1}^m\frac{\left|q_{ab}^i-\sum_{j=1}^m r_{aj}^i x_{jb}\right|_\omega}{\beta_{ab}(\omega)}\right\}\right\}, \tag{6.210}$$

where $\beta_{ab}(\omega)$ is used to weight the elements of column b based on the multivariable tracking priority. This is one approach to weighting the cost function in (6.210). Note that the structure in (6.210) is equivalent to

$$\{x_{1b}(j\omega),x_{2b}(j\omega),...,x_{mb}(j\omega)\}=\min\left\{\max_{i\in[1,n]}\left\{\sum_{a=1}^m\frac{\left|a_{ab}^{ik}\right|_\omega}{\beta_{ab}(\omega)}\right\}\right\}, \tag{6.211}$$

where $\frac{\left|a_{ab}^{ik}\right|_\omega}{\beta_{ab}(\omega)}$ is the first term in (6.204). In order to convert (6.210) into a form that can be interpreted by a constrained optimisation routine, an additional free parameter, $\kappa_b(\omega)>0$, is introduced, which replaces the objective function,

$$\{x_{1b}(j\omega),x_{2b}(j\omega),...,x_{mb}(j\omega)\}=\min\left\{\kappa_b(\omega)\right\}. \tag{6.212}$$

The required constraint that relates (6.212) to (6.210) is

$$\sum_{a=1}^m\frac{\left|q_{ab}^i-\sum_{k=1}^m r_{ak}^i x_{jb}\right|_\omega}{\beta_{ab}(\omega)}\leq\kappa_b(\omega),\ \forall i\in[1,n]. \tag{6.213}$$

The objective is to minimise $\kappa_b(\omega)$, which in turn represents the maximum, or worst-case occurrence (it overbounds the plant-indexed set in question). In practice, each column element of $X_o(j\omega)$ is separated into the real-valued magnitude and phase components, $x_{ab}(j\omega) = R_{ab}(\omega)e^{j\theta_{ab}(\omega)}$, which necessitates the constraints, $R_{ab}(\omega) \geq 0$, $\theta_{ab}(\omega) \in [-\pi, \pi)$. Following this formulation, an $m \times m$ feedforward filter results in m optimisation problems per frequency design point (one for each column), each with $2m + 1$ free parameters (two parameters for each element in a specific column of $X_o(j\omega)$ and one parameter for $\kappa_b(\omega)$ as detailed in (6.213)). The search space therefore increases linearly with m and does not suffer from higher-order scalability issues when handling plants with large dimensions.

Generating the tracking design constraint set

Once $X_o(j\omega)$ has been synthesized, the resulting tracking error can be calculated using (6.196). Using knowledge of this prototype $E_o(j\omega)$ set, as well as defining the incremental error bounds from (6.207), the left-hand side of (6.204) is populated. Equation (6.204) and (6.207) are then used to generate the decoupled tracking design regions of the diagonal elements of $G(s)$. Note that when $|E_o^i(j\omega)| = 0$, $\forall i \in [1, n]$ (implying $\Delta\beta^i(\omega) = \beta^i(\omega)$), (6.195) and (6.200) become equivalent. This corresponds to the infinite feedback controller gain solution, which is analogous to having no *a priori* design information/insight.

The resulting plant-paired elliptical *stay-out* design regions from (6.204) are combined with the discoidal *stay-in* design regions from (6.207), which collectively determine the globally admissible solution space of $G(j\omega)$. As shown in Figure 6.43, the closed-form geometric structure of the individual design regions can be approximated by a polygon with sufficient boundary points. The intersection of all *stay-in* polygons will then define the resulting frequency-dependent design regions that can be visualised in the arithmetic-complex plane, or mapped to the log-polar plane. This is different to the traditional QFT approach of iteratively fixing the controller phase angle and then finding the roots of the polynomial that describes the underlying constraint at equality on the controller gain. The primary appeal in using polygonal approximations is in the way that the phase gridding is performed. With reference to Figure 6.43, the traditional approach approximates design regions by discretising the continuous range of controller phase angles that span $[-\pi, \pi)$, and this has the potential to result in poorly approximated design regions, depending on the relative size and location of the design region. By stark contrast, discretisation using triangular segments with a common point at the polygon centre is less susceptible to underapproximation and requires less phase partitions to sufficiently capture an individual design region. This has the benefit of being computationally efficient, relative to traditional methods, and is more consistent in terms of the amount of conservatism present when approximating the design regions.

Feedback control refinement via iteration

The procedure detailed from Section 6.6.4 to 6.6.4 may be iterated to try to improve on the feedback control design. A high-level summary of the algorithm is illustrated in Table 6.3. As indicated in Section 6.6.4, the model-error tracking sensitivity with respect to the controller elements will predominantly affect the

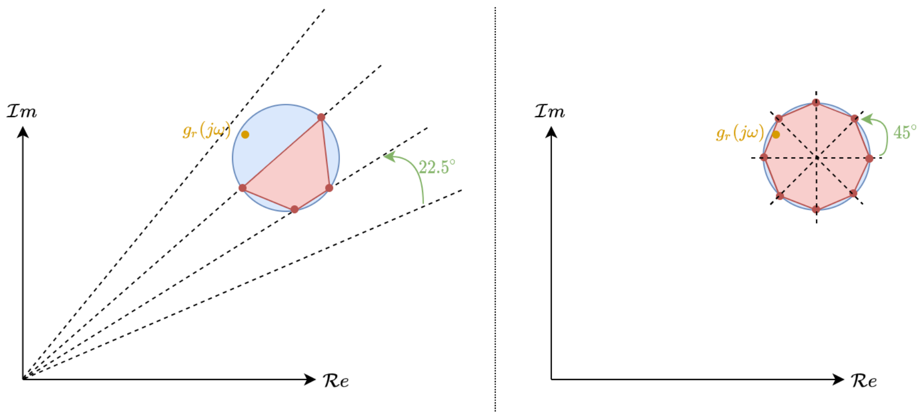


Figure 6.43: A graphical example of the mathematically derived (blue) and approximated (red) *stay-in* design regions, represented in the arithmetic-complex plane. The traditional QFT approach (left) grids the arithmetic-complex plane based on the phase angle of $g_r(j\omega)$. The mathematically derived design region (blue disc) is approximated by the quadrilateral polygon, which can exhibit large underapproximation. The proposed approach (right) grids the same design region using triangular segments that originate from the design region centre. A valid $g_r(j\omega)$ is shown to lie in the blue disc, but it would be perceived as invalid when using traditional phase gridding with a 22.5° phase angle step.

Step	Operation
1	Set current feedback controller iteration equal to valid *a priori*, design, e.g. based on Section 6.6.4: $G_o(s) \leftarrow G(s)$.
2	Synthesize prototype $X_o(j\omega)$ using current $G_o(s)$, based on (6.212): $\{x_{1b}(j\omega), x_{2b}(j\omega), ..., x_{mb}(j\omega)\} = \min\{\kappa_b(\omega)\}$.
3	Select/adjust $\lambda(\omega)$ and update incremental tracking tolerance, as detailed in Section 6.6.4: $\Delta\beta(\omega) = \lambda(\omega) \odot \beta(\omega)$.
4	Generate design regions for $G(j\omega)$ based on (6.204) and (6.207).
5	Return to *Step 3* and refine $\lambda(\omega)$ if required, based on observation of design regions for $G(j\omega)$ in *Step 4*.
6	Design $G(s)$ using design reqions from *Step 4*.
7	Update feedback controller prototype: $G_o(s) \leftarrow G(s)$.
8	Return to *Step 2* if further feedback controller shaping is required. Otherwise, end refinement process.

Table 6.3: Iterative feedback control refinement algorithm

number of iterations required when refining the feedback controller elements. That is, large incremental steps in the feedback controller are allowed when the sensitivity is low, but small increments are required when the sensitivity is relatively high. One approach is to set the elements in $\lambda(\omega)$ to be very small ($\lambda_{ab}(\omega) << 1$, $\forall\{a, b\} \in [1, m]$), which would allow small differential changes in $g_a(j\omega)$, based on (6.207), whilst simultaneously having reduced bound dominance from the elliptical *stay-out* tracking specification (the left-hand side of (6.200) would approach the non-conservative form). The obvious downside is that the increased number of design iterations would require longer time investment from the designer (as the bound generation routine would need to be executed repeatedly). It may be possible to automate the incremental controller refinements, but this is not investigated in this article.

Incremental refinement of the feedback controllers requires that the updated design constraints from (6.204) yield design regions that allow lower gain and more phase flexibility. Geometrically, this requires the elliptical stay-out regions from (6.204) to shrink in size (or at least the plant-pair-indexed ellipses that define the boundary conditions of the global *stay-out* design regions). As previously mentioned, setting $\Delta\beta(\omega) \to 0$ implies that the constraint set of (6.200) is nonconservative (as there is no gain and phase overbounding present in the constraint). Under these conditions, the feedforward filter synthesis step, from Section 6.6.4, will find a prototype $X_o(j\omega)$ that is not biased by conservatism from the triangle inequality, and will free up design space for the feedback controller elements, if it exists. Specifically, the left-hand side of (6.200) will continue to become smaller with each design iteration, allowing for iterative adjustment of $G(j\omega)$ on the corresponding right-hand side terms, until such time that application of (6.213) no longer shrinks the elliptical stay-out regions from (6.204).

Feedforward filter design

Assuming a $G(s)$ can be designed that conforms to the aforementioned constraints in terms of the Bode gain-phase relationship, the final step is to design continuous $X(s)$. The proposed approach is to exploit the linear structure of the model tracking error constraint to define separate existence conditions on each column of $X(s)$. Defining (now known) $V^i(j\omega) = \left[I + P^i(j\omega)G(j\omega)\right]^{-1} M(j\omega)$, and $Z^i(j\omega) = \left[I + P^i(j\omega)G(j\omega)\right]^{-1} P^i(j\omega)$, the model tracking error for plant i can be written as

$$E^i(j\omega) = V^i(j\omega) - Z^i(j\omega)X^i(j\omega). \tag{6.214}$$

Column c of $E^i(j\omega)$ can be written in expanded form as

$$E^i_{\{c\}}(j\omega) = V^i_{\{c\}}(j\omega) - Z^i(j\omega)X_{\{c\}}(j\omega) = V^i_{\{c\}}(j\omega) - \sum_{k=1}^{m} Z^i_{\{k\}}(j\omega)x_{kc}(j\omega), \tag{6.215}$$

where subscript $\{c\}$ indicates column (vector) c of a particular matrix. The corresponding element-wise design constraint follows as

$$\left| V^i_{\{c\}} - \sum_{k=1}^{m} Z^i_{\{k\}}x_{kc} \right|_{\omega} \preccurlyeq \beta_{\{c\}}(\omega),\ \forall i \in [1, n], \tag{6.216}$$

where $V^i_{\{c\}}(j\omega), Z^i_{\{k\}}(j\omega) \in \mathbb{C}^{m\times 1}$. Given that (6.216) must be satisfied for all plant cases ($\forall i \in [1, n]$), the above requirement can be reposed as

$$\left| \tilde{V}^1_{\{c\}} - \sum_{k=1}^{m} \tilde{Z}^1_{\{k\}}x_{kc} \right|_{\omega} \preccurlyeq \tilde{\beta}_{\{c\}}(\omega), \tag{6.217}$$

where

$$\tilde{V}^1_{\{c\}}(j\omega) = \begin{bmatrix} V^1_{\{c\}}(j\omega) \\ V^2_{\{c\}}(j\omega) \\ \vdots \\ V^n_{\{c\}}(j\omega) \end{bmatrix} \in \mathbb{C}^{nm\times 1} \tag{6.218}$$

and

$$\tilde{Z}^1_{\{k\}}(j\omega) = \begin{bmatrix} Z^1_{\{k\}}(j\omega) \\ Z^2_{\{k\}}(j\omega) \\ \vdots \\ Z^n_{\{k\}}(j\omega) \end{bmatrix} \in \mathbb{C}^{nm\times 1} \tag{6.219}$$

are stacked vectors that comprise every plant-dependent instance for column c and k, respectively. Similarly, $\tilde{\beta}(j\omega) = \mathbf{I}_{(nm\times m)}\beta(j\omega)$, where $\mathbf{I}_{(nm\times m)}$ is an $nm \times m$ matrix that contains n vertically stacked $m \times m$ identity matrices. Equation (6.217) can be rewritten as

$$\left| \tilde{V}^1_{\{c\}} - \sum_{k=1}^{m-1} \tilde{Z}^1_{\{k\}} x_{kc} - \tilde{Z}^1_{\{m\}} x_{mc} \right|_{\omega} \leqslant \tilde{\beta}^1_{\{c\}}(\omega) \tag{6.220}$$

without loss of generality. Following this, element-wise division on both sides of (6.220) by $\tilde{Z}^1_{\{m\}}(j\omega)$ exposes the design regions of $x_{mc}(j\omega)$, namely

$$\left| \tilde{V}^1_{\{c\}} \oslash \tilde{Z}^1_{\{m\}} - \sum_{k=1}^{m-1} \left[\tilde{Z}^1_{\{k\}} \oslash \tilde{Z}^1_{\{m\}} \right] x_{kc} - x_{mc}\mathbf{1}_{(nm\times 1)} \right|_{\omega} \leqslant \tilde{\beta}^1_{\{c\}}(\omega) \oslash \left| \tilde{Z}^1_{\{m\}}(j\omega) \right|, \tag{6.221}$$

where $\mathbf{1}_{(nm\times 1)}$ is an $nm \times 1$ column vector of ones. Each row of (6.221) describes an independent constraint on $x_{mc}(j\omega)$, which takes the geometric form of a discoidal *stay-in* design region in the arithmetic-complex plane. Specifically, the design region resulting from row a of (6.221) will have a centre located at $\left. \frac{\tilde{V}^1_{ac} - \sum_{k=1}^{m-1} \tilde{Z}^1_{ak} x_{kc}}{\tilde{Z}^1_{am}} \right|_{\omega}$, with a corresponding radius equal to $\frac{\tilde{\beta}^1_{ac}(\omega)}{|\tilde{Z}^1_{am}(j\omega)|}$. Ensuring a common intersection between any two *stay-in* discs, based on Figure 6.38, defines the necessary condition for the existence of a globally (across the entire plant set) feasible $x_{mc}(j\omega)$, which follows as

$$\left| \tilde{V}^1_{ac} \oslash \tilde{Z}^1_{am} - \tilde{V}^1_{bc} \oslash \tilde{Z}^1_{bm} - \sum_{k=1}^{m-1} \left(\left[\tilde{Z}^1_{ak} \oslash \tilde{Z}^1_{am} \right] - \left[\tilde{Z}^1_{bk} \oslash \tilde{Z}^1_{bm} \right] \right) x_{kc} \right|_{\omega} \leqslant \tilde{\beta}^1_{ac}(\omega) \oslash \left| \tilde{Z}^1_{am} \right|_{\omega} + \tilde{\beta}^1_{bc}(\omega) \oslash \left| \tilde{Z}^1_{bm} \right|_{\omega}, \tag{6.222}$$

where $\forall\{a, b\} \in [1, nm]$. Noting the structural similarities with (6.217), (6.222) can be written compactly as

$$\left| \tilde{V}^2_{\{c\}} - \sum_{k=1}^{m-1} \tilde{Z}^2_{\{k\}} x_{kc} \right|_{\omega} \leqslant \tilde{\beta}^2_{\{c\}}(\omega), \tag{6.223}$$

where row $r = 2(a-1) + b$, column c of $\tilde{V}^2(j\omega)$, $\tilde{Z}^2(j\omega)$, and $\tilde{\beta}^2(\omega)$ are described respectively as

$$\tilde{V}^2_{rc}(j\omega) = \left[\tilde{V}^1_{ac} \oslash \tilde{Z}^1_{am} - \tilde{V}^1_{bc} \oslash \tilde{Z}^1_{bm} \right]_{\omega}, \tag{6.224}$$

$$\tilde{Z}^2_{rc}(j\omega) = \left[\tilde{Z}^1_{ac} \oslash \tilde{Z}^1_{am} \right]_{\omega} - \left[\tilde{Z}^1_{bc} \oslash \tilde{Z}^1_{bm} \right]_{\omega}, \tag{6.225}$$

and

$$\tilde{\beta}^2_{rc}(j\omega) = \tilde{\beta}^1_{ac}(\omega) \oslash \left|\tilde{Z}^1_{am}\right|_{\omega} + \tilde{\beta}^1_{bc}(\omega) \oslash \left|\tilde{Z}^1_{bm}\right|_{\omega}. \tag{6.226}$$

The number of combinations results in $\tilde{V}^2_{\{c\}}(j\omega), \tilde{Z}^2_{\{c\}}(j\omega), \tilde{\beta}^2_{\{c\}}(j\omega) \in \mathbb{C}^{n^2m^2\times 1}$, but after excluding redundant (repeated) combinations, there will actually be $nm(nm-1)/2$ combinations. Referring to (6.223), one can then follow the same sequential procedure as in (6.220) and (6.221) to expose the design region of $x_{(m-1)c}(j\omega)$:

$$\left|\tilde{V}^2_{\{c\}} \oslash \tilde{Z}^2_{\{m-1\}} - \sum_{k=1}^{m-2} \left[\tilde{Z}^2_{\{k\}} \oslash \tilde{Z}^2_{\{m-1\}}\right] x_{kc} - x_{(m-1)c}\mathbf{1}_{(n^2m^2\times 1)}\right|_{\omega} \leqslant \tilde{\beta}^2_{\{c\}}(\omega) \oslash \left|\tilde{Z}^2_{\{m-1\}}(j\omega)\right|. \tag{6.227}$$

This method is applied $m-1$ times in total to obtain the set of univariate design constraints on $x_{1c}(j\omega)$,

$$\left|\tilde{V}^m_{\{c\}} - x_{1c}\mathbf{1}_{(n^m m^m \times 1)}\right|_{\omega} \leqslant \tilde{\beta}^m_{\{c\}}(\omega). \tag{6.228}$$

All discoidal *stay-in* design regions of $x_{1c}(j\omega)$ in (6.228) must be satisfied simultaneously, which then gives rise to design regions of $x_{2c}(j\omega)$. Selecting a valid $x_{2c}(j\omega)$ then defines a non-empty design region for $x_{3c}(j\omega)$, and so on, until $x_{mc}(j\omega)$ is eventually designed. This constitutes sequentially chaining together existence conditions that informs the design of $x_{(k+1)c}$ based on the successive design choices for rows 1 to k. Importantly, the columns of $X(j\omega)$ are designed independently as there is no interaction in terms of meeting the tracking specifications (as evidenced by (6.215)). This approach is therefore applicable to arbitrarily large multivariable tracking designs and is equivalent to the bound generation routine outlined in [20] when $m = 2$. Note that this feedforward design process is independent of the feedback control design routine that is applied. In other words, it can be incorporated into any 2DOF control design scheme. The feedforward filter row design order of $\{1 \rightarrow 2 \rightarrow \ldots \rightarrow m\}$ is used here for sake of clarity, but the designer can choose the order in which the feedforward elements are designed based on the order in which existence conditions are specified. Changing the design order may have benefit in completing the design, but this is not included in this study.

6.6.5 Worked examples

This Section provides solutions to two different benchmark problems [115],[5] with varying complexity and design requirements: the first to illustrate the method on a simple problem with only gain uncertainty; and the second, the ill-conditioned distillation column. The emphasis is placed on the bound generation and analysis, with the controller design only included to illustrate that a causal, pragmatic control solution exists.

Example 1: Benchmark problem with pure gain uncertainty

In order to demonstrate the efficacy of the proposed method, the proposed design routine is first applied to a fully specified problem [115], with an existing solution [5]. The uncertain plant under consideration is

$$\mathbf{P}(s) = \frac{1}{s}\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}, \ \{k_{11}, k_{22}\} \in [2, 6], \ \{k_{12}, k_{21}\} \in [0.5, 1.5], \tag{6.229}$$

which exhibits approximately 10 dB of gain uncertainty on each plant element and has an open-loop interaction index of ≤ 0.75 [94]. The corresponding reference model and tracking error bounds, in the frequency band of $\omega \leq 10$ rad/s, are given as

$$M(s) = \mathrm{diag}\left\{\frac{1}{s/3+1}, \frac{1}{s/3+1}\right\}, \ \beta_{ab}(\omega) = 0.2\omega\sqrt{1+\omega^2/9}, \ \forall\{a, b\} \in [1, 2]. \tag{6.230}$$

Additionally, the high frequency ($\omega \geq 10$ rad/s) sensitivity constraint is

$$\frac{1}{\left|1 + g_a/\hat{p}^i_{aa}\right|_\omega} \leq 4.5 \text{ dB}, \ \forall i = \{1, 2, ..., n\}. \tag{6.231}$$

A preliminary $G_o(s)$ is designed using the constraint set in (6.195), whilst also incorporating the high-frequency sensitivity specification in (6.231). With reference to Section 6.6.4, 360 boundary points are used to approximate each plant- or plant-paired-indexed design region resulting from (6.204) and (6.207). Appropriate union and intersect operations are then applied to the *stay-out* and *stay-in* polygonal approximations, respectively, which collectively forms the design space of $g_r(j\omega)$ within the arithmetic-complex plane. The design regions of the nominal open-loop transfer function are shown in the log-polar plane in Figure 6.44a, with

$$g_{1_0}(s) = 3.43\frac{\frac{s}{22}+1}{\left(\frac{s}{8.5}+1\right)\left(\left(\frac{s}{40}\right)^2+\frac{0.8s}{40}+1\right)}. \tag{6.232}$$

Importantly, the arbitrarily selected nominal plant in Figure 6.44 of $P^0(s) = \frac{1}{s}\left[\begin{smallmatrix} 2 & 0.5 \\ 0.5 & 2 \end{smallmatrix}\right]$ has no effect on the solution space of $g_{1_0}(s)$ and is only included to preserve the traditional open-loop logarithmic gain-phase locus behaviour. Note that Figure 6.44a and (6.232) also represent a valid design of non-sequentially designed $g_{2_0}(s)$, as the plant and tracking specifications, in (6.229) and (6.230) respectively, are symmetric.

With prototype $G_o(s) = \mathrm{diag}\{g_{1_0}(s), g_{2_0}(s)\}$ defined, the feedforward filter synthesis proceeds as detailed in Section 6.6.4. Each column of $X_o(j\omega)$ is optimally selected based on the objective of minimising the maximum of the weighted sum of the particular column's elements in (6.210), and this results in two optimisation problems (one per column), with five parameters per optimisation problem. $G_o(j\omega)$ and optimally selected $X_o(j\omega)$ are used to calculate the set of $E_o(j\omega)$, with $\lambda(\omega)$ initially set to 0.5. Following this, the

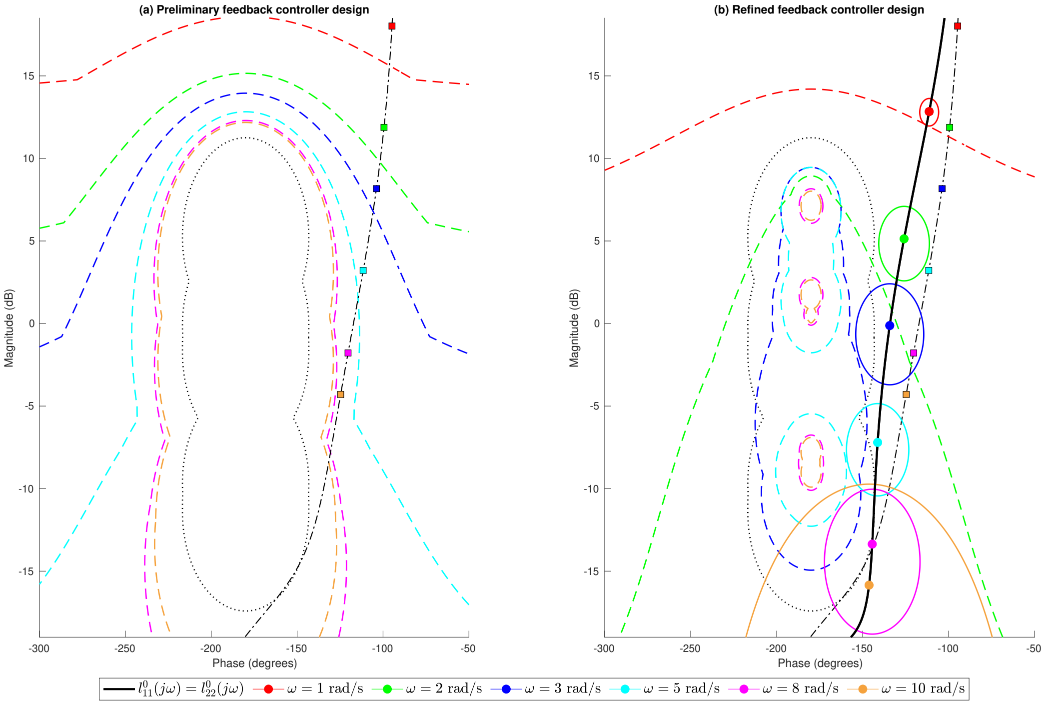


Figure 6.44: Universal *stay-out* sensitivity design regions (dotted line) and model-error tracking design regions on nominal loop transfer function, $l^0_{11}(j\omega) = \frac{2}{j\omega}g_1(j\omega)$ for (a) preliminary feedback controller design (dashed-dotted black line) using (6.195) and (b) refined feedback controller design (solid black line) using (6.204) and (6.207). The *stay-in* design regions from (6.207) are denoted with solid lines and the *stay-out* design regions from (6.195) and (6.204) are shown with dashed lines. The nominal open-loop loop transfer behaviour from the preliminary design is also shown in (b) for sake of visual comparison.

design constraints in (6.204) and (6.207) are populated and yield *stay-out* and *stay-in* design regions on $G(j\omega)$, respectively. Referring to Table 6.3, $\lambda(\omega)$ is then adjusted until a satisfactory balance between the per-frequency *stay-out* and *stay-in* design regions are found. A refinement is then made to $G_o(s)$, with the objective of reducing the gain and phase at all frequencies. The refined feedback controller is then used to synthesize an updated $X_o(j\omega)$, which in turn gives rise to updated design constraints for $G_o(j\omega)$. The design regions in Figure 6.44b are the result of two refinement iterations, with an improved feedback controller solution of

$$g_1(s) = g_2(s) = 2.2\frac{\frac{s}{11.26} + 1}{\left(\frac{s}{2.173} + 1\right)\left((\frac{s}{22})^2 + \frac{0.64s}{22} + 1\right)}, \tag{6.233}$$

with the corresponding $\lambda(\omega)$ matrix and synthesized $X_o(j\omega)$ shown in Table 6.4 and Table 6.5, respectively.

The preliminary and adjusted feedback controller solutions are also shown in Figure 6.44b to illustrate the substantial relative gain/phase reduction after refining $G_o(s)$. The high-frequency sensitivity specifications from (6.231) are used as universal sensitivity constraints in the refined design (Figure 6.44b) to safeguard

ω (rad/s)	1	2	3	5	8	10
$\lambda_{ab}(\omega),\ \forall\{a,b\} \in [1,2]$	0.1	0.3	0.4	0.4	0.16	0.135

Table 6.4: Frequency-dependent scaling parameters used when setting $\Delta\beta(\omega) = \lambda(\omega) \odot \beta(\omega)$ from (6.208). The symmetrical structure is a natural result of the symmetrical plant uncertainty and tracking specifications.

ω (rad/s)	1	2	3	5	8	10
$X_{11}(j\omega), X_{22}(j\omega)$	$0.0202 + 0.599j$	$-0.129 + 0.689j$	$0.128 + 0.115j$	$-0.305 + 0.072j$	-0.219	-0.314
$X_{12}(j\omega), X_{21}(j\omega)$	$0.073 - 0.455j$	$0.321 - 0.227j$	$0.439 + 0.355j$	$0.991 + 0.241j$	$1.168 + 0.358j$	$1.348 + 0.309j$

Table 6.5: Synthesized $X_o(j\omega)$, after two iterations, using minimax procedure in Section 6.6.4. The feedback controller from (6.233) is used to populate $V_o^i(j\omega)$ and $Z_o^i(j\omega)$ from (6.209).

against undesirable sensitivity characteristics in the mid-frequency band ($\omega \in [2, 10]$). This was not required in the preliminary design as the tracking specifications from (6.195) dominated the gain-phase allowance (the sensitivity constraints were superceded by the mid- to high-frequency tracking constraints). This highlights the dramatic bound improvements from using *a priori* model-error tracking information. The relative gain and phase reductions in the refined feedback controller are significant, with frequency-dependent gain reductions in the feedback controllers of 4 to 10.3 dB, and phase reductions ranging from 16 to 29.4 degrees (for $\omega \in [1, 10]$).

Completing the 2DOF control scheme requires selecting a suitable $X(s)$ that adheres to the requirements in (6.186). This is facilitated using the procedure detailed in Section 6.6.4. Specifically, (6.228) is used directly to generate independent design regions for $x_{1c}(j\omega),\ \forall c \in [1, 2]$, that must be satisfied simultaneously. Selection of a valid $x_{1c}(s)$ then engenders non-empty design regions for the corresponding $x_{2c}(j\omega)$ element. Figure 6.45 shows the *stay-in* design regions of $X(s)$ in the log-polar plane of $\arg\{X(j\omega)\}$ vs $|X(j\omega)|_{\text{dB}}$. A viable, symmetrical, second-order solution of

$$X(s) = \begin{bmatrix} 0.66\frac{s}{(s/2.5+1)(s/5+1)} & -0.46\frac{s}{(s/3.4+1)(s/5.98+1)} \\ -0.46\frac{s}{(s/3.4+1)(s/5.98+1)} & 0.66\frac{s}{(s/2.5+1)(s/5+1)} \end{bmatrix} \tag{6.234}$$

is overlaid in Figure 6.45 and is shown to meet all frequency-dependent design regions. As highlighted in [20], the feedforward structure in Figure 6.37 necessitates a feedforward filter with zero DC gain (i.e., derivative action), in order to achieve zero-error reference tracking at steady-state for *Type-0* feedback controllers, when considering step-like reference signals. The *stay-in* design regions in Figure 6.45 are smallest at the low frequencies (especially at $\omega = 1$ rad/s), which is related to how closely $G(j\omega)$ approached the frequency-dependent boundaries in Figure 6.44b. This is attributed to the fact that the design constraints on $g_a(j\omega)$ in (6.200) determine the existence of the corresponding $x_{ac}(j\omega)$ elements, and satisfying said constraint at equality for at least one plant instance causes the solution space of $x_{ac}(j\omega)$ to collapse to a point. By extension, setting $g_a(j\omega)$ such that the equality condition in (6.200) is approached will monotonically shrink the design region of $x_{ac}(j\omega)$. This, combined with the fact that design conservatism in (6.200) is most prevalent in the mid- to high-frequency band (as a result of the plant-inverting formulation) means that satisfying the low-frequency

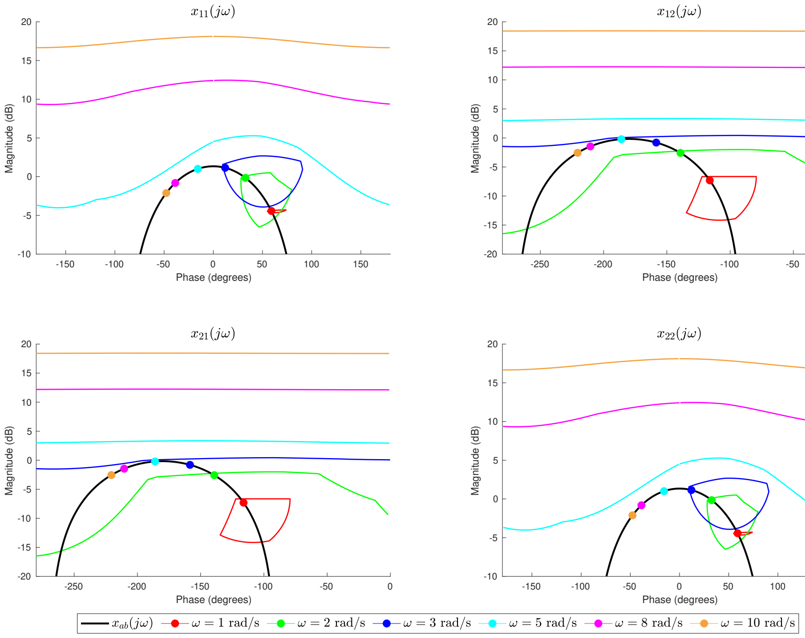


Figure 6.45: *Stay-in* design regions of $x_{ab}(j\omega)$ that arise from application of (6.228), described in the log-polar plane of $\arg\{x_{ab}(j\omega)\}$ vs $|x_{ab}(j\omega)|_{\text{dB}}$. $x_{12}(j\omega) = x_{21}(j\omega)$ is designed first, which dictates the corresponding solution space of $x_{11}(j\omega) = x_{22}(j\omega)$, based on satisfying the existence condition from (6.228).

design frequency requirements for $X(j\omega)$ in Figure 6.45 are most challenging, but the high-frequency design becomes trivial. This illustrates the efficacy of the proposed feedforward design procedure in Section 6.6.4, as there is no reliance on conservative overbounding, such as the triangle inequality. Verification of meeting the tracking specifications in (6.186) is demonstrated by plotting the frequency response magnitude of the model-error tracking set against the tracking tolerance, as shown in Figure 6.46. The strict adherence to the model-error tracking tolerance in Figure 6.46 (within the low- to mid-frequency band where model tracking is prioritised) exemplifies the proposed method's ability to reduce design conservatism via iterative refinement and existential feedforward filter design.

Example 2: Distillation column

The poorly conditioned distillation column problem [6] is used as a final demonstration of the proposed method. The purpose is to focus less on the actual design and rather draw attention to the potential gain-phase saving, as well as the expanded solution spaces for both the feedback and feedforward control designs. The linearized plant is described by

$$P(s) = \frac{1}{1+75s}\begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}\begin{bmatrix} k_1 e^{-sT_1} & 0 \\ 0 & k_2 e^{-sT_2}, \end{bmatrix}, \tag{6.235}$$

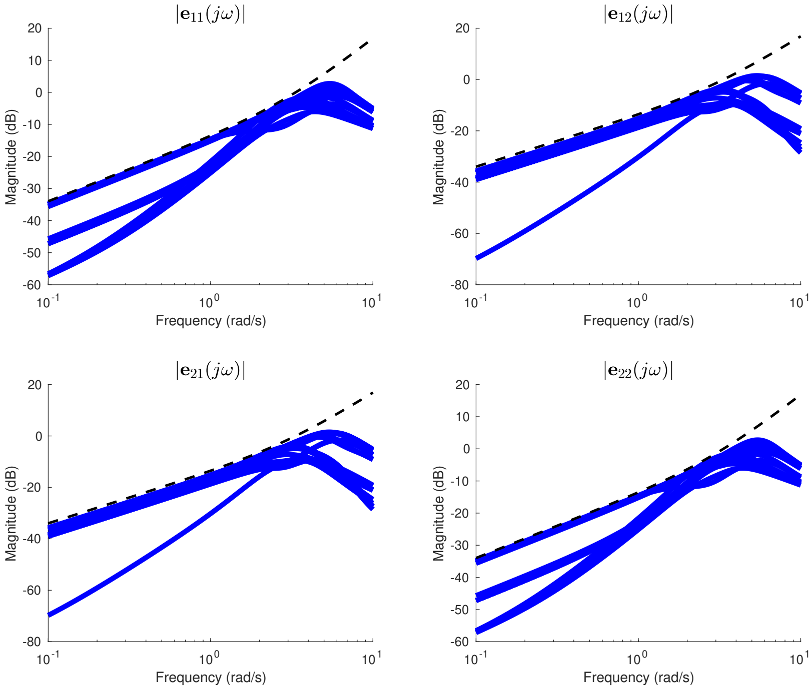


Figure 6.46: Bode magnitude plot showing 2×2 model-error tracking set (blue) from (6.187) and its adherence to the prescribed tracking tolerance (dashed black line) in (6.230).

ω	**0.01**	**0.02**	**0.04**	0.055	0.08	**0.1**	0.13	0.15	0.2	**0.3**	0.4	**0.5**	1	1.5
$\beta_{11}(\omega)$	**0.12**	**0.2**	**0.33**	0.48	0.6	**0.6**	0.6	0.6	0.6	**0.6**	0.5	**0.65**	0.58	0.5
$\beta_{12}(\omega)$	**0.5**	**0.5**	**0.5**	0.5	0.5	**0.5**	0.5	0.5	0.5	**0.5**	0.5	**0.5**	0.55	0.55
$\beta_{21}(\omega)$	**0.5**	**0.5**	**0.5**	0.5	0.5	**0.5**	0.5	0.5	0.4	**0.4**	0.45	**0.6**	0.7	0.6
$\beta_{22}(\omega)$	**0.1**	**0.1**	**0.15**	0.25	0.3	**0.3**	0.32	0.33	0.33	**0.31**	0.43	**0.55**	0.65	0.65

Table 6.6: Discrete tracking tolerances at chosen design frequencies from [5]. The frequency is expressed in radians per minute. Boldface entries denote the design frequencies and corresponding tracking tolerances that are used in Figure 6.47.

where $\{k_1, k_2\} \in [0.8, 1.2]$, and $\{T_1, T_2\} \in [0, 1]$ min, and has an open-loop interaction index of ≤ 0.985 [94]. For sake of transparent bench-marking, the design specifications defined in [5] are used in this worked example. The reference model specifications are

$$M(s) = \frac{1}{(\frac{s}{0.11})^2 + \frac{1.4s}{0.11} + 1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{6.236}$$

with the corresponding tracking tolerances detailed in Table 6.6. Note that for sake of clarity only a subset of the design frequency points in Table 6.6 were used. The robust stability specifications from [5] follow as

$$\frac{1}{|1 + g_1/\hat{p}_{11}|_\omega} \leq 1.3 \text{ dB}, \quad \frac{1}{|1 + p_{11}g_1|_\omega} \leq 8 \text{ dB}, \tag{6.237}$$

and

$$\frac{1}{\left|1 + g_2/\hat{p}^*_{22}\right|_\omega} \leq 1.3 \text{ dB}, \tag{6.238}$$

where $\hat{p}^*_{22}$ is detailed in [5] as an equivalent plant that is used in the sequential design of $G(s)$ and includes information from $g_1(s)$. The solution from [5], repeated in (6.239) below, is used as the preliminary design and represents an appropriate baseline to design from:

$$G_o(s) = \begin{bmatrix} \frac{2.2(0.008)(0.018)(0.45)(0.94,0.94)}{s(0.165)(0.25,0.64)(20,0.49)} & 0 \\ 0 & \frac{-3(0.02)(0.8)(3.2)(0.7,0.4)}{s(0.4)(10)(10)(0.24,0.65)} \end{bmatrix}, \tag{6.239}$$

where the short-hand notation of (a) and (b, c) corresponds to $(s/a + 1)$ and $(s^2/b^2 + 2cs/b + 1)$, respectively.

Using the preliminary feedback controller design in (6.239), and noting that the existence of a corresponding valid $X_o(s)$ is ratified based on the completed solution in [5], a prototype $X_o(j\omega)$ is synthesized using (6.213). The procedure detailed in Table 6.3 is followed, whereby $G_o(s)$ and $X_o(j\omega)$ are iteratively designed and synthesized, respectively, with $\lambda(\omega)$ used to trade design strictures between the two control channels. This is especially invaluable in the distillation column problem, as the two channels have fundamentally different gain-phase requirements (which was not the case for the symmetrical benchmark example in Section 6.6.5). Equation (6.238) is a function of selected $g_1(j\omega)$ and is reevaluated after each iteration to update the stability

ω (rad/min)	0.01	0.02	0.04	0.1	0.3	0.5
$X_{11}(j\omega)$	$14.315 - 36.413j$	$7.707 - 53.387j$	$2.827 - 75.054j$	$25.339 - 95.259j$	$16.153 + 3.004j$	$3.173 + 1.264j$
$X_{12}(j\omega)$	$54.902 + 88.736j$	$48.216 + 73.299j$	$31.342 + 81.438j$	$-16.403 + 79.710j$	$-10.767 - 0.924j$	$-1.117 \times 10^{-6} - 1.857 \times 10^{-8}j$
$X_{21}(j\omega)$	$-8.359 - 1.084j$	$-2.979 - 80.971j$	$12.242 - 77.853j$	$25.506 - 14.437j$	$1.113 \times 10^{-6} + 8.128 \times 10^{-7}j$	$7.682 \times 10^{-7} - 4.351 \times 10^{-7}j$
$X_{22}(j\omega)$	$-14.397 + 26.208j$	$-12.636 + 25.641j$	$-17.720 + 35.710j$	$-25.733 + 8.768j$	$-1.462 - 2.808j$	$0.021 - 0.567j$

Table 6.7: Synthesized $X_o(j\omega)$, after two iterations, using minimax procedure in Section 6.6.4. The feedback controller from (6.233) is used to populate $V_o^i(j\omega)$ and $Z_o^i(j\omega)$ from (6.209).

ω (rad/min)	0.01	0.02	0.04	0.1	0.3	0.5
$\lambda_{11}(\omega), \lambda_{12}(\omega)$	0.35	0.1	0.075	0.3	0.15	0.1
$\lambda_{21}(\omega), \lambda_{22}(\omega)$	0.5	0.5	0.5	0.1	0.05	0.025

Table 6.8: Frequency-dependent scaling parameters used when setting $\Delta\beta(\omega) = \lambda(\omega) \odot \beta(\omega)$ from (6.208).

requirements of $g_2(j\omega)$. As in Section 6.6.5, 360 boundary points are used when forming the polygonal approximations that define the individual design regions. The resulting nominal loop transfer design regions are shown in Figure 6.47 after two refinement iterations, with the corresponding feedback controller solution of

$$G(s) = 0.6G_o(s) = \begin{bmatrix} \frac{1.32(0.008)(0.018)(0.45)(0.94,0.94)}{s(0.165)(0.25,0.64)(20,0.49)} & 0 \\ 0 & \frac{-1.8(0.02)(0.8)(3.2)(0.7,0.4)}{s(0.4)(10)(10)(0.24,0.65)} \end{bmatrix}, \tag{6.240}$$

and corresponding prototype feedforward filter as shown in Table 6.7. Note that arbitrarily selected $P^0(s) = \frac{0.8}{1+75s}\begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix}$ is used to display the nominal open-loop design regions in Figure 6.47. The corresponding $\lambda(\omega)$ values are shown in Table 6.8. Note that for sake of simplicity, only the gain of $G(s)$ was altered when performing the design refinements. Importantly, the reduced gain solution in (6.240) and Figure 6.47 (4.44 dB gain reduction) could have been further refined, but the allocated headroom (meeting the requirements in (6.200) away from equality) allows for additional design freedom when shaping $X(s)$, based on the existential link between the feedback controller and feedforward filter.

Following the successful design of $G(s)$, the feedforward filter is designed as detailed in Section 6.6.4. The resulting design regions are shown in Figure 6.48, overlaid with a valid solution of

$$X(s) = \begin{bmatrix} \frac{37.57(0.036)}{(0.113,0.41)} & \frac{86.14(0.055,0.5)}{(0.02)(0.55,0.31)(0.11,0.3)} \\ \frac{2233s(0.13,0.43)(0.63,0.38)}{(0.69,0.83)(0.13,0.32)(0.08,0.64)} & \frac{-12.13(0.011)(0.37,0.19)}{(0.24)(0.14)(0.26,0.90)(0.37,0.50)} \end{bmatrix}. \tag{6.241}$$

As noted in [20] and [5], the span of the frequency-dependent design regions in Figure 6.48 are directly dictated by emphasizing (or de-emphasizing) the disturbance rejection characteristics over the model-error tracking characteristics. In other words, overdesigning the feedback controllers in Figure 6.47 would lead to larger design regions on the feedforward filter, based on the existence conditions in (6.200), whereas minimum gain-phase design of the feedback controller results in heavily constrained pre-filter shaping flexibility. The latter case can be seen in the second row of Figure 6.48, at the mid-frequencies. Given that the feedforward filter design procedure outlined in Section 6.6.4 is also facilitated by existence conditions, the selection of

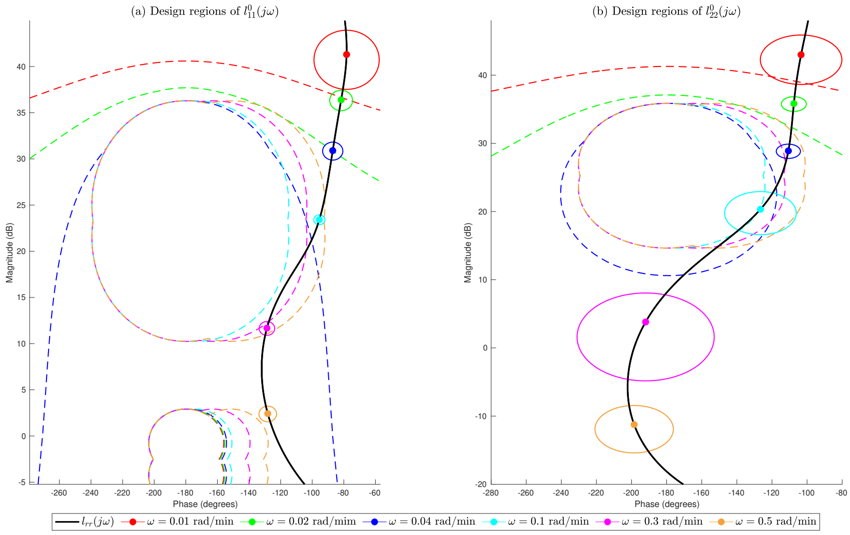


Figure 6.47: Log-polar plane showing the design regions, from (6.204), (6.237), and (6.238), on (a) nominal loop transfer function, $l_{11}^0(j\omega) = \frac{0.7024}{75\omega j+1} g_1(j\omega)$ and (b) nominal loop transfer function, $l_{22}^0(j\omega) = -\frac{0.88}{75\omega j+1} g_2(j\omega)$. *Stay-out* design regions (intersection of tracking and stability requirements) are depicted with dashed lines and *stay-in* design boundaries from (6.207) are shown with solid lines. For sake of clarity, a subset of the design frequencies from Table 6.6 are shown.

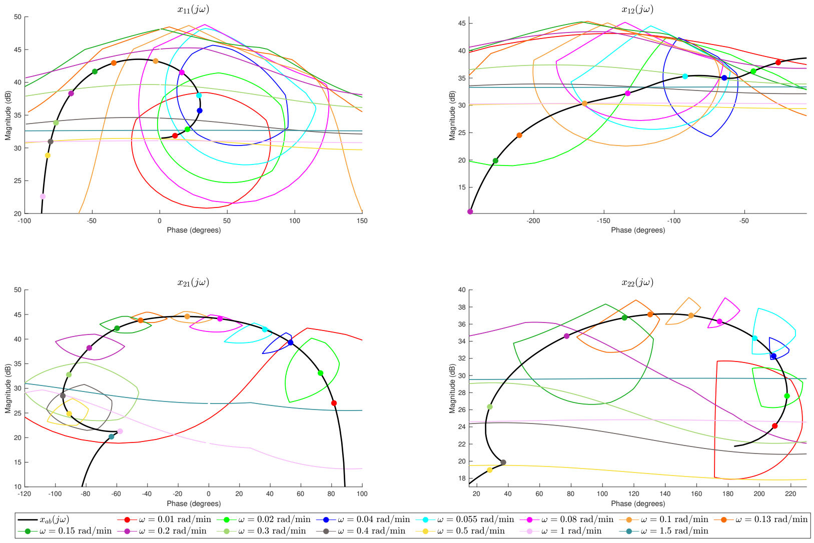


Figure 6.48: *Stay-in* design regions of $x_{ab}(j\omega)$, described in the log-polar plane of $\arg\{x_{ab}(j\omega)\}$ vs $|x_{ab}(j\omega)|_{\text{dB}}$. The $x_{1b}(j\omega)$ elements are designed first, which gives rise to valid solution spaces for the $x_{2b}(j\omega)$ elements, based on satisfying the existence condition from (6.228).

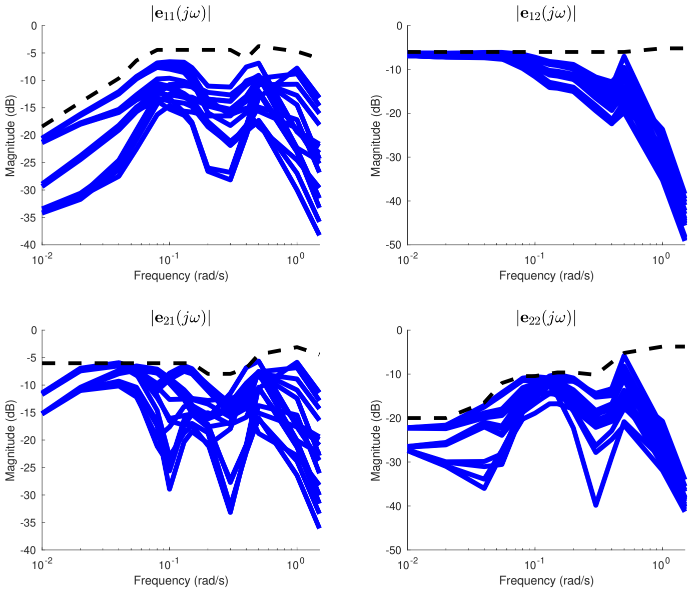


Figure 6.49: Bode magnitude plot showing 2 × 2 model-error tracking set (blue) from (6.187) and its adherence to the prescribed tracking tolerances (dashed black) in Table 6.6.

$x_{1b}(j\omega)$ in Figure 6.48 ultimately determines the corresponding $x_{2b}(j\omega)$ solution space based on how strictly the necessary condition in (6.228) is adhered to. The designer is therefore required to constantly allocate additional headroom to all interim controller designs, in order to prevent subsequent loop shaping issues as a result of the Bode gain-phase relationship. An appealing option could be to incorporate filter synthesis from [118], which would theoretically allow for lower gain-phase feedback controller solutions, but this is not encapsulated in the scope of this article.

The frequency-based design routine is ratified in Figure 6.49 by comparing the modulus of the model-error tracking behaviour with the tracking specifications in Table 6.6. The proposed nonconservative feedforward filter design methodology is exemplified by the fact that selection of an $x_{ab}(j\omega_d)$ lying on the edge of the *stay-in* design region at $\omega = \omega_d$ (in Figure 6.48) will result in $\beta_{cb}(j\omega_d) \in |\mathbf{e}_{cb}(j\omega_d)|$, where row c is dictated by the most stringent constraint in (6.216). An example of this is at $\omega = 0.1$ rad/min, whereby $x_{22}(j0.1)$ lies on the edge of the *stay-in* design region in Figure 6.48 and results in $\beta_{22}(j0.1) \in |\mathbf{e}_{22}(j0.1)|$, as shown in Figure 6.49. For sake of completeness, the time-domain unit step responses of the closed-loop plant set are shown in Figure 6.50. Note that the purpose of this approach is not to demonstrate strict adherence to the time-domain tracking specifications from [6], but to rather show that the frequency-domain specifications from [5] are enforced with reduced overdesign, as evidenced in 6.49.

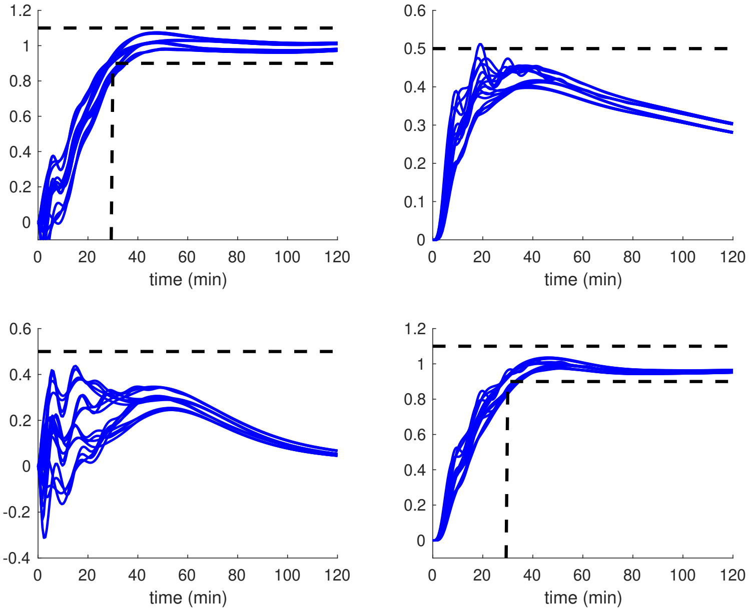


Figure 6.50: Element-wise step responses of closed-loop plant set. The time-domain specifications from [6] are also shown for sake of completeness. Minor violations of the time-domain requirements can be seen, which would ordinarily necessitate tightening of the tracking tolerances.

6.6.6 Conclusion

This Section has presented a new, iterative feedback controller refinement approach to multivariable model-error tracking design problems. Additionally, a generalised feedforward filter design routine has been proposed that does not rely on the triangle inequality and assists in reducing the feedback controller gain-phase requirements. This combined method has been applied to two 2×2 benchmark problems of varying complexity and design requirements, and was shown to significantly reduce the gain and phase requirements of the feedback controller across all relevant design frequencies.

Chapter 7

Conclusion

This chapter concludes the thesis by providing a summary of the key findings of the research, and recommending avenues for future research that make use of the concepts introduced in the preceding chapters.

7.1 Summary of findings

7.1.1 Variable-pitch quadcopter

The mechanical and mathematical modelling, design, state estimation, low-level control, and preliminary testing of a novel variable-pitch quadcopter has been presented. The multi-rotor platform encompasses a wide range of engineering disciplines, including control engineering, robotics, software design, aeronautics, and electromechanical engineering.

Chapter 2 provided the experimental framework of the project. The mechanical and electronic design of the variable-pitch quadcopter was detailed, including the preceding design iterations that informed future physical alterations. A vision-based motion capture system was designed and implemented to provide absolute pose information of the quadcopter. A high-fidelity simulation environment was also developed, as well as a thrust rig, which was used to perform system identification, calibration, and mock flight testing.

Chapter 3 developed rotor thrust and torque equations from first principles, followed by deriving the fundamental rectilinear and rotation equations of motion of the variable-pitch quadcopter. Experimental methods were then used to characterise the low-level quadcopter dynamics, which ratified the expected rotor thrust and torque equations. Rotor thrust and speed plant sets were also developed using system identification techniques.

Chapter 4 detailed the designed and constructed visual tracking system, which was able to estimate translation and orientation of a rigid-body vehicle within the flight space. The approach relied on asynchronously fusing camera information in a quasi-continuous-discrete extended Kalman filter algorithm. An on-board attitude estimator was also developed, which combined the quaternion estimate from the camera system with on-board information from a gyroscope, which resulted in an improved attitude estimate. Finally, a novel rotor thrust observer was designed and implemented, which made use of motor current to approximate the quadcopter rotor thrusts.

Chapter 5 comprised of low-level control design and implementation of the rotor thrust and speed loops. Appropriate feedback control design had the effect of reducing plant uncertainty and disturbance effects, which effectively decoupled the multivariable quadcopter dynamics. Information from the low-level loop closures was then used to form plant sets that could be used in the design of the attitude and position controllers. A yaw-free thrust vectoring approach was also introduced, which discarded the yaw channel when making inertial-frame position corrections. This had the benefit of simplifying the attitude dynamics. Following this, a simulation of the quadcopter position control system was presented, which made use of the high-fidelity simulation environment. A simple cross axis position manoeuvre was shown, which demonstrated the efficacy of the various internal control authority allocation and feedback control schemes.

7.1.2 Contributions to QFT

This thesis has provided incremental contributions to the field of quantitative feedback theory, with the intention of overcoming some of the pitfalls of the QFT boundary generation methodology. The investigation leans on mathematical and engineering insights to reformulate the QFT-based tracking error problem. Three novel bound generation routines have been presented that are applicable to single-input-single-output, two-input-two-output, and multi-input-multi-output systems, respectively.

Section 6.4 illustrated that a per-plant model reference approach is effective in gaining a handle on plant input constraints when optimising the reference models for a single-input-single-output system. The tracking error design constraints take the form of quartic inequalities and are defined based on existence conditions for the pre-filter. This unique approach results in *stay-in* design regions of the feedback controller that precludes infinite gain solutions.

Moving to the class of two-input-two-output systems, Section 6.5 addressed design conservatism that is prevalent in conventional QFT design routines. The provided solution supplements traditional plant-inverting bound generation routines with a novel non plant-inverting formulation that has the potential to substantially reduces the level of overbounding at the mid to high frequencies — where the loopshaping is often most challenging. As the cost of feedback generally is critical at the gain-phase crossover region, this method is also effective in reducing the amount of control action required to satisfy the design specifications.

Section 6.6 focused on the generalised class of multi-input-multi-output systems, and introduced an iterative bound generation and loop-shaping routine that exploited the mathematical structure of the tracking error constraints. This has the potential to substantially reduce design conservatism that arises from applying the triangle inequality and enables the designer to take a preexisting control solution and systematically reduce gain and phase requirements without violating the performance specifications. A nonconservative feedforward design routine is also provided that makes use of existence conditions to sequentially design each row of the controller.

These investigations resulted in three novel bound generation routines that collectively target any class of linear time-invariant control system (SISO, TITO, and MIMO). Each design routine was applied to a simulated control problem with appropriate closed-loop specifications, and were shown to be effective in achieving the required performance with minimum gain and phase. These results have implications for any realistic control system that desires cost effective and pragmatic implementation requirements.

7.2 Future research

The research contained in this thesis may be useful for further contributions related to quadcopters and quantitative feedback bound generation routines.

7.2.1 Variable-pitch quadcopter

The developed variable-pitch quadcopter platform and motion capture system serve as a useful test-bench for further research. The logical next step is to perform flight tests based on the simulation results in Section 5.9. This could begin with attitude-only control (using a pilot to close the position loop and avoid crashes), followed by position control flight testing. QFT design insights can be used to design SISO control schemes that are able to provide robust stability under various operating conditions.

7.2.2 Contributions to QFT

The main innovation in Section 6.4, of introducing per-plant model specifications, can be extended to the generalised multivariable problem. This may expose similar benefits in terms of designing for signal constraints in the time-domain, but a more interesting research avenue would be to investigate if a multi-model approach reduces design conservatism when establishing tracking design constraints. Given that the largest contributor to design conservatism in a QFT formulation is lack of implicit gain and phase information of the model tracking error, selecting appropriate per-plant reference models may have the effect of reducing the worst-case gain-phase assumptions. One approach could be to take a completed solution using a single-model approach, and then allocate discoidal subsets of the original *stay-in* boundary for each plant instance. In this way, satisfying the per-plant model specifications implicitly satisfies the original model-error specification. Comparison of the single-model and multi-model feedback controller gains can then be used to infer under what conditions this approach is viable. This will be investigated for SISO and MIMO systems in a subsequent research study.

Another unexplored aspect of the boundary generation method in Chapter 4 is that unconventional *stay-in* feedback controller design regions arise as a result of using multiple reference models. Unlike traditional boundaries that inherently include the infinite gain solution, these frequency response design regions can be geometrically convex in shape when represented in the arithmetic-complex plane, and contain a finite region in which the parameterised feedback controller frequency responses must lie. This has the potential to work in unison with an optimisation routine, such as that of [118], that can automatically search the finite solution space and synthesize a control solution.

The refinement approach that was introduced in Chapter 6 is semi-autonomous in that the designer is still required to guide the evolution of the feedback controller elements and make decisions about the gain and phase trade-off between the control loops. There is potential to near-fully automate this procedure in a way that allows minimal manual interaction. This would have multiple benefits: Incremental steps in the design iterations of the feedback controller elements can be made significantly smaller, which will further reduce the design conservatism brought on by application of the triangle inequality. The designer can also take more a hands-off approach and rely on refinement that is dictated by an overarching cost function. This will be explored in a future research study.

Appendix A

Rotation matrices and quaternions

Quaternions and rotation matrices are used to encode the orientation of one reference frame with respect to another. Specifically, when attaching a reference frame to the rigid-body quadcopter, the orientation of the quadcopter in the inertial frame is equivalent to determining the orientation of the body-frame axis system with respect to the defined inertial-frame system.

A.1 Rotation matrices

A.1.1 Mathematical properties

The generalised 3 × 3 rotation matrix [68], **R**, comprises column vectors that have unitary length and are strictly orthogonal with respect to each other. The rotation matrix therefore obeys the orthogonality property

$$\mathbf{R}\mathbf{R}^T = \mathbf{I}, \tag{A.1}$$

and has a determinant of $\det(\mathbf{R}) = \pm\mathbf{1}$. By extension, this implies that the transpose of the rotation matrix is equal to its inverse, $\mathbf{R}^T = \mathbf{R}^{-1}$. Expressing the rotation matrix as

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{A.2}$$

the unit-norm constraint in (A.1) is described explicitly as

$$r_{11}^2 + r_{21}^2 + r_{31}^2 = 1, \tag{A.3}$$

$$r_{12}^2 + r_{22}^2 + r_{32}^2 = 1,$$

$$r_{13}^2 + r_{23}^2 + r_{33}^2 = 1,$$

whereas the orthogonality condition is

$$r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} = 0, \tag{A.4}$$

$$r_{12}r_{13} + r_{22}r_{23} + r_{32}r_{33} = 0,$$

$$r_{11}r_{13} + r_{21}r_{23} + r_{31}r_{33} = 0,$$

which corresponds to setting the inner product of each column vector pair to zero. When applied to state estimation and control, the rotation matrix describes the encoded orientation of one reference frame, with respect to another. For example, if frame $\{B\}$ is attached to a robot of interest, and frame $\{W\}$ is attached to the stationary environment (such as the centre of the room), then the orientation of the frame $\{B\}$ with respect to frame $\{W\}$ is indicated the rotation matrix ${}^{W}\mathbf{R}_{B}$. Given that the inverse of a rotation matrix is equal to its transpose, it can be shown that

$${}^{W}\mathbf{R}_{B}{}^{-1} = {}^{W}\mathbf{R}_{B}{}^{T} = {}^{B}\mathbf{R}_{W}, \tag{A.5}$$

where ${}^{B}\mathbf{R}_{W}$ describes the encoded orientation of frame $\{W\}$ with respect to frame $\{B\}$.

The rotation matrix is commonly parameterised using Euler angles, based on their relative simplicity. Note that the type and ordering of the Euler angle parameterisation used will affect the internal structure of the rotation matrix. The intrinsic *ZYX* Euler angle formulation is shown below,

$${}^{W}\mathbf{R}_{B} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \cos\phi\sin\psi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \cos\theta\sin\psi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix}, \tag{A.6}$$

where ϕ, θ, and ψ correspond respectively with the roll, pitch, and yaw Euler angles of frame $\{B\}$ relative to frame $\{W\}$. The angles can also be extracted from the rotation matrix using

$$\phi = \arctan(\frac{r_{21}}{r_{11}}), \tag{A.7}$$

$$\theta = \arcsin(-r_{31}), \tag{A.8}$$

$$\psi = \arctan(\frac{-r_{32}}{r_{33}}). \tag{A.9}$$

With reference to (A.1), rotation matrices have the constant relationship of $\mathbf{R}\mathbf{R}^{T} = \mathbf{I}$. Taking the time derivative of both sides of $\mathbf{R}\mathbf{R}^{T} = \mathbf{I}$ yields

$$\dot{\mathbf{R}}\mathbf{R}^{T} + \mathbf{R}\dot{\mathbf{R}}^{T} = \mathbf{0}. \tag{A.10}$$

Defining $\mathbf{S} = \dot{\mathbf{R}}\mathbf{R}^{T}$, (A.10) can be written as

$$\mathbf{S} + \mathbf{S}^{T} = \mathbf{0}, \tag{A.11}$$

with $\dot{\mathbf{R}} = \mathbf{S}\mathbf{R}$. Equation (A.11) suggests that $\mathbf{S}$ is a skew-symmetric matrix as $\mathbf{S}^T = -\mathbf{S}$. By definition, the diagonal elements of $\mathbf{S}$ must all equal zero in order to satisfy $\mathbf{S}^T = -\mathbf{S}$. Skew-symmetric matrices are useful in reposing a vector cross product operation as a matrix-vector multiplication. Given an arbitrary vector $\boldsymbol{\Omega} = [\Omega_x \ \Omega_y \ \Omega_z]^T$, the corresponding skew-symmetric matrix operator is

$$\lfloor \boldsymbol{\Omega} \rfloor_\times = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \omega_x & 0 \end{bmatrix}. \tag{A.12}$$

Evaluating a cross product operation of two arbitrary vectors, $\boldsymbol{\Omega}$ and $\mathbf{p}$, is therefore equivalent to the product of the matrix in (A.12) and vector $\mathbf{p}$

$$\omega \times \mathbf{p} = \lfloor \boldsymbol{\Omega} \rfloor_\times \mathbf{p}. \tag{A.13}$$

Given that $-\omega \times \mathbf{p} = \mathbf{p} \times \omega$, it follows that

$$-\lfloor \boldsymbol{\Omega} \rfloor_\times \mathbf{p} = \lfloor \mathbf{p} \rfloor_\times \boldsymbol{\Omega}. \tag{A.14}$$

Consider an arbitrary vector ${}^B\mathbf{p}$, fixed in frame $\{B\}$, with the corresponding frame $\{W\}$ vector description of ${}^W\mathbf{p} = {}^W\mathbf{R}_B {}^B\mathbf{p}$. Noting that frame $\{B\}$ and frame $\{W\}$ have coincident origins, and ${}^B\dot{\mathbf{p}} = \mathbf{0}$ (the position vector is constant in frame $\{B\}$), taking the time derivative on both sides of ${}^W\mathbf{p} = {}^W\mathbf{R}_B {}^B\mathbf{p}$ yields,

$${}^W\dot{\mathbf{p}} = {}^W\dot{\mathbf{R}}_B {}^B\mathbf{p}. \tag{A.15}$$

It is important to point out that the derivative of an orthonormal rotation matrix is not an orthonormal matrix. ${}^W\dot{\mathbf{p}}$ can also be determined using elementary mechanics. Specifically, the linear velocity of ${}^W\mathbf{p}$ can be constructed from the cross product of the angular velocity, ${}^W\boldsymbol{\Omega}$, and position vector, ${}^W\mathbf{p}$, both described in frame $\{W\}$,

$${}^W\dot{\mathbf{p}} = {}^W\boldsymbol{\Omega} \times {}^W\mathbf{p} = \lfloor {}^W\boldsymbol{\Omega} \rfloor_\times {}^W\mathbf{p}. \tag{A.16}$$

Incorporating ${}^W\mathbf{p} = {}^W\mathbf{R}_B {}^B\mathbf{p}$ into (A.16) results in

$${}^W\dot{\mathbf{p}} = \lfloor {}^W\boldsymbol{\Omega} \rfloor_\times {}^W\mathbf{R}_B {}^B\mathbf{p}. \tag{A.17}$$

Comparing (A.15) and (A.17), an explicit description of the rate of change of the rotation matrix follows as

$${}^W\dot{\mathbf{R}}_B = \lfloor {}^W\boldsymbol{\Omega} \rfloor_\times {}^W\mathbf{R}_B. \tag{A.18}$$

The result above shows that the rate of change of the encoded orientation is a function of both the angular velocity vector and current orientation. It can easily be shown that

$$\lfloor \mathbf{R}\boldsymbol{\Omega} \rfloor_{\times} = \mathbf{R}\lfloor \boldsymbol{\Omega} \rfloor_{\times}\mathbf{R}^{T}. \tag{A.19}$$

The angular velocity of the two frames is related by

$${}^{W}\boldsymbol{\Omega} = {}^{W}\mathbf{R}_{B}{}^{B}\boldsymbol{\Omega}, \tag{A.20}$$

which can be used in (A.18) to obtain

$${}^{W}\dot{\mathbf{R}}_{B} = \lfloor {}^{W}\mathbf{R}_{B}{}^{B}\boldsymbol{\Omega} \rfloor_{\times}{}^{W}\mathbf{R}_{B}. \tag{A.21}$$

Using the identity in (A.19), (A.21) can be reposed as

$${}^{W}\dot{\mathbf{R}}_{B} = {}^{W}\mathbf{R}_{B}\lfloor {}^{B}\boldsymbol{\Omega} \rfloor_{\times}. \tag{A.22}$$

A.2 Quaternions

A.2.1 Mathematical properties

Euler angles are appealing as an introduction to understanding orientation, as they are relatively intuitive and easy to visualise. However, the potential for the rotation matrix becoming degenerate and resulting in kinematic singularities is problematic in robotic applications that require arbitary combinations of rotation angles.

The quaternion is a popular choice of encoding attitude that avoids the issue of gimbal lock by instead describing the orientation as a single rotation about some defined axis that intercepts the relevant reference frame origin. The quaternion [119], **q**, is a 4-element vector (also known as a 4-tuple), and is structured as

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{bmatrix}, \tag{A.23}$$

in which q_0, q_x, q_y, and q_z are real numbers. In the context of describing orientation, only unit quaternions are considered, which obeys the constraint of

$$q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1. \tag{A.24}$$

Explicitly, the unit quaternion follows an angle-axis structure of

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\alpha}{2} \\ \sin\frac{\alpha}{2}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \cos\frac{\alpha}{2} \\ \sin\frac{\alpha}{2}v_x \\ \sin\frac{\alpha}{2}v_y \\ \sin\frac{\alpha}{2}v_z \end{bmatrix}, \tag{A.25}$$

where $\mathbf{v} = [v_x\ v_y\ v_z]^T$ is the unit-length ($|\mathbf{v}| = 1$) rotation axis, and $\alpha \in [-\pi, \pi)$ is the rotation angle. Equation (A.25) is a way of encoding the 3D rotation in a useful format, analogous to how rotation matrices encode attitude. The angle-axis information can be extracted from the quaternion using

$$\alpha = 2\cos^{-1}(q_0), \tag{A.26}$$

and

$$\mathbf{v} = \frac{1}{\sin\frac{\alpha}{2}} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \frac{1}{\sqrt{1 - q_0^2}} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}. \tag{A.27}$$

When $q_0 = 1$, (which implies that $\alpha = 0$), the relationship in (A.27) breaks, making the rotation axis become undefined, but this is not problematic, as this corresponds to the null rotation. Specifically, this is referred to as the the identity (or null) quaternion, with the form $\mathbf{q_I} = [1\ 0\ 0\ 0]^T$.

The quaternion is related to the rotation matrix by

$$\mathbf{R} = \begin{bmatrix} 2(q_0^2 + q_x^2) - 1 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & 2(q_0^2 + q_y^2) - 1 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & 2(q_0^2 + q_z^2) - 1 \end{bmatrix}. \tag{A.28}$$

Given two unit quaternions, $\mathbf{q}$ and $\mathbf{r}$, the quaternion product (or Hamilton Product), indicated using the symbol $\otimes$, is defined as

$$\mathbf{q} \otimes \mathbf{r} = \begin{bmatrix} q_0 r_0 - q_x r_x - q_y r_y - q_z r_z \\ q_0 r_x + r_0 q_x + q_y r_z - r_y q_z \\ q_0 r_y + r_0 q_y + r_x q_z - q_x r_z \\ q_0 r_z + r_0 q_z + q_x r_y - r_x q_y \end{bmatrix}. \tag{A.29}$$

The quaternion inverse is defined as

$$\mathbf{q}^{-1} = \begin{bmatrix} q_0 \\ -q_x \\ -q_y \\ -q_z \end{bmatrix}, \tag{A.30}$$

which can be though of in two ways: either (i) the rotation vector changes direction and points in the opposite direction, or (ii) the rotation angle changes direction. Analogous to the rotation matrix, the quaternion that describes the orientation of frame $\{B\}$ with respect to frame $\{W\}$ is ${}^{W}\mathbf{q}_{B}$. Using the definition for the quaternion inverse, it follows that

$$ {}^{W}\mathbf{q}_{B}^{-1} = {}^{B}\mathbf{q}_{W}. \tag{A.31} $$

The unit quaternion does not uniquely define an orientation. Specifically, the orientation derived using a rotation about some axis is equivalent to a negative rotation about that same axis after it has been inverted. This results in a double cover phenomenon where $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. As a result of this double covering, every unique orientation will map to two quaternions: one that describes the minimum angle rotation, and the other describing the maximum angle rotation (unless the rotation angle is 180°, making both rotations equivalent in terms of angular distance). This becomes important if a robot is attempting to follow a path. In practice, this minor caveat is avoided by actively selecting the minimum angle rotation when required.

The relationship between angular velocity and the quaternion follows a similar pattern to that of the rotation matrix counterpart. Analogous to (A.18), the quaternion derivative [48] can be written as

$$ {}^{W}\dot{\mathbf{q}}_{B} = \frac{1}{2}{}^{W}\mathbf{q}_{B} \otimes \begin{bmatrix} 0 \\ {}^{B}\boldsymbol{\Omega} \end{bmatrix}. \tag{A.32} $$

The result in (A.32) can be used to update a quaternion estimate using angular rate information obtained in the body-frame, which takes the form of

$$ \begin{aligned} {}^{W}\mathbf{q}_{B}(t+\Delta t) &= {}^{W}\mathbf{q}_{B}(t) + \Delta t\,{}^{W}\dot{\mathbf{q}}_{B}(t), \\ &= {}^{W}\mathbf{q}_{B}(t) + \frac{\Delta t}{2}{}^{W}\mathbf{q}_{B}(t) \otimes {}^{B}\boldsymbol{\Omega}(t), \\ &= {}^{W}\mathbf{q}_{B}(t) \otimes \left(\mathbf{q}_{\mathbf{I}} + \frac{\Delta t}{2}{}^{B}\boldsymbol{\Omega}(t)\right). \end{aligned} \tag{A.33} $$

This result requires less computations than rotation matrix update equations. Note that adding a nonzero vector to a unit quaternion will result in a non-unit quaternion. However, if the angular velocity and/or sample time is sufficiently small, the approximation becomes reasonable. As with the rotation matrix update, we require normalisation of the quaternion in order to maintain the unitary structure. In the case of the quaternion, normalisation simply requires dividing all elements by the quaternion norm

$$ {}^{W}\mathbf{q}_{B} = \frac{{}^{W}\mathbf{q}_{B}}{|{}^{W}\mathbf{q}_{B}|}. \tag{A.34} $$

This type of process is commonly found in state estimation routines, such as an extended Kalman filter, that incorporates quaternions to describe orientation [22].

A.2.2 Quaternion error formulation

The differential equation of a quaternion, described in the inertial-frame, is

$$ {}^{W}\dot{\mathbf{q}}_{B} = \frac{1}{2} {}^{W}\mathbf{q}_{B} \otimes {}^{B}\mathbf{\Omega}, \tag{A.35}$$

where ${}^{B}\tilde{\mathbf{\Omega}} = [0 \;\; {}^{B}\mathbf{\Omega}^{T}]^{T}$ is the measured-body-frame rate from the gyroscope, resized as a 4-vector. Similarly, the differential equation of the desired quaternion, described in the inertial-frame, is

$$ {}^{W}\dot{\mathbf{q}}^{*}_{B} = \frac{1}{2} {}^{W}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*}, \tag{A.36}$$

where ${}^{B}\tilde{\mathbf{\Omega}}^{*} = [0 \;\; {}^{B}\mathbf{\Omega}^{*T}]^{T}$ is the desired angular rate 4-vector, described in the body-frame. The desired quaternion in the body-frame is described by

$$\begin{aligned} {}^{B}\mathbf{q}^{*}_{B} &= ({}^{W}\mathbf{q}_{B})^{-1} \otimes {}^{W}\mathbf{q}^{*}_{B}, \\ &= {}^{B}\mathbf{q}_{W} \otimes {}^{W}\mathbf{q}^{*}_{B}. \end{aligned} \tag{A.37}$$

The time derivative of the above error quaternion is

$$ {}^{B}\dot{\mathbf{q}}_{B} = {}^{B}\dot{\mathbf{q}}_{W} \otimes {}^{W}\mathbf{q}^{*}_{B} + {}^{W}\mathbf{q}_{B} \otimes {}^{W}\dot{\mathbf{q}}^{*}_{B}. \tag{A.38}$$

After some simplification, (A.38) can be expressed as

$$ {}^{B}\dot{\mathbf{q}}^{*}_{B} = \frac{1}{2} {}^{B}\mathbf{\Omega}^{-1} \otimes {}^{W}\mathbf{q}_{B}{}^{-1} \otimes {}^{W}\mathbf{q}^{*}_{B} + \frac{1}{2} {}^{W}\mathbf{q}^{*}_{B}{-1} \otimes {}^{W}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*}. \tag{A.39}$$

The above equation simplifies to

$$ {}^{B}\dot{\mathbf{q}}^{*}_{B} = -\frac{1}{2} {}^{B}\mathbf{\Omega} \otimes {}^{B}\mathbf{q}^{*}_{B} + \frac{1}{2} {}^{B}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*}. \tag{A.40}$$

Post-multiplying the second term in the above equation by the identity quaternion, ${}^{B}\mathbf{q}^{*}_{B}{}^{-1} \otimes {}^{B}\mathbf{q}^{*}_{B} = [1\,0\,0\,0] = 1$, gives

$$ {}^{B}\dot{\mathbf{q}}^{*}_{B} = -\frac{1}{2} {}^{B}\mathbf{\Omega} \otimes {}^{B}\mathbf{q}^{*}_{B} + \frac{1}{2} {}^{B}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*} \otimes ({}^{B}\mathbf{q}^{*}_{B})^{-1} \otimes {}^{B}\mathbf{q}^{*}_{B}. \tag{A.41}$$

Analysing ${}^{B}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*} \otimes ({}^{B}\mathbf{q}^{*}_{B})^{-1}$ from the second term of (A.41) and expanding yields,

$$ {}^{B}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*} \otimes ({}^{B}\mathbf{q}^{*}_{B})^{-1} = {}^{W}\mathbf{q}_{B}{}^{-1} \otimes \left[{}^{W}\mathbf{q}^{*}_{B} \otimes {}^{B}\mathbf{\Omega}^{*} \otimes ({}^{W}\mathbf{q}^{*}_{B})^{-1}\right] \otimes {}^{W}\mathbf{q}_{B}. \tag{A.42}$$

$$ = {}^{W}\mathbf{q}_{B}{}^{-1} \otimes \left[{}^{W}\mathbf{q}^{*}_{B} \otimes [0 \;\; ({}^{B}\vec{\Omega}^{*})^{T}]^{T} \otimes ({}^{W}\mathbf{q}^{*}_{B})^{-1}\right] \otimes {}^{W}\mathbf{q}_{B}. \tag{A.43}$$

Equation (A.43) describes a rotation of the vector ${}^B\vec{\Omega}^*$ from the desired-body-frame to the inertial-frame, followed by a rotation from the inertial-frame into the measured-body-frame. Equation (A.43) therefore corresponds to the desired rate described in the body-frame,

$$ {}^B\mathbf{q}_B^* \otimes {}^B\mathbf{\Omega}^* \otimes ({}^B\mathbf{q}_B^*)^{-1} = {}^B\mathbf{\Omega}^*. \tag{A.44} $$

Following from this, (A.41) can be rewritten as

$$ {}^B\dot{\mathbf{q}}_B^* = -\frac{1}{2}{}^B\mathbf{\Omega} \otimes {}^B\mathbf{q}_B^* + \frac{1}{2}{}^B\mathbf{\Omega}^* \otimes {}^B\mathbf{q}_B^* = \frac{1}{2}\left({}^B\mathbf{\Omega}^* - {}^B\mathbf{\Omega}\right) \otimes {}^B\mathbf{q}_B^* = \frac{1}{2}{}^B\mathbf{\Omega}^* \otimes {}^B\mathbf{q}_B^*, \tag{A.45} $$

where ${}^B\mathbf{\Omega}^*$ is the body-frame rate error. Post-multiplying both sides of the above equation by twice the quaternion error conjugate yields

$$ {}^B\mathbf{\Omega}^* = 2{}^B\dot{\mathbf{q}}_B^* \otimes ({}^B\mathbf{q}_B^*)^{-1}. \tag{A.46} $$

The vector portion of ${}^B\mathbf{\Omega}^*$ can be described as

$$ {}^B\mathbf{\Omega}^* = 2({}^Bq_B^*)_0{}^B\dot{q}_B^* - 2({}^B\dot{q}_B^*)_0{}^Bq_B^* - 2{}^B\dot{q}_B^* \times {}^Bq_B^*. \tag{A.47} $$

Rewriting the above equation in the equivalent angle-vector form gives

$$ {}^B\mathbf{\Omega}^* = 2\cos(\frac{\alpha^e}{2})[\dot{\vec{v}}^e\sin(\frac{\alpha^e}{2})+\vec{v}^e\cos(\frac{\alpha^e}{2})\frac{\dot{\alpha}^e}{2}]+2\vec{v}^e\sin^2(\frac{\alpha^e}{2})\frac{\dot{\alpha}^e}{2}-2[\dot{\vec{v}}^e\sin(\frac{\alpha^e}{2})+\vec{v}^e\cos(\frac{\alpha^e}{2})\frac{\dot{\alpha}^e}{2}]\times\vec{v}^e\sin(\frac{\alpha^e}{2}). \tag{A.48} $$

Making use of the appropriate trigonometric identities, and noting that $\vec{v}^e \times \vec{v}^e = 0$, the above equation simplifies to

$$ {}^B\mathbf{\Omega}^* = \sin(\alpha^e)\dot{\vec{v}}^e + \dot{\alpha}^e\vec{v}^e + [\cos(\alpha^e) - 1][\dot{\vec{v}}^e \times \vec{v}^e]. \tag{A.49} $$

For small angle errors the measured-body-frame rate error reduces to

$$ {}^B\mathbf{\Omega}^* = \alpha^e\dot{\vec{v}}^e + \dot{\alpha}^e\vec{v}^e. \tag{A.50} $$

Finally, integrating the above equation with initial conditions set to zero derives a usable measured-body-frame angle error extracted from the quaternion error

$$ {}^B\Theta^* = \int {}^B\mathbf{\Omega}^* dt = \alpha^e\vec{v}^e. \tag{A.51} $$

It is worth noting that (A.51) is also valid when the Euler axis changes sufficiently slowly, $\dot{\vec{v}}^e \rightarrow \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, regardless of the magnitude of the Euler error.

Appendix B

Trajectory generation

B.1 Cubic polynomial interpolation

Cubic (3^{rd}-order) polynomials are commonly used to interpolate between two points, and can be written as

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \tag{B.1}$$

where a_0, a_1, a_2, and a_3 are constants. Equation (B.1) can be repeatedly differentiated to find the corresponding kinematic profiles, namely the quadratic *velocity* profile

$$\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2, \tag{B.2}$$

the linear *acceleration* profile

$$\ddot{s}(t) = 2a_2 + 6a_3 t, \tag{B.3}$$

and the constant *jerk* profile

$$\dddot{s}(t) = 6a_3. \tag{B.4}$$

The velocity profile in (B.2) notably exhibit a continuous behaviour, and this is a necessary condition for a *smooth* position trajectory. However, the acceleration and jerk profile do not have a continuous behaviour. If we were interested in ensuring there was continuity at the acceleration and jerk level, we could use a higher order polynomial, such a quartic (4^{th}-order) or quintic (5^{th}-order).

Given that there are four undefined constants, we need to impose four constraints on the various profiles. This can be done by assigning initial and final position and velocity values,

$$a_0 = s(0),$$

$$a_1 = \dot{s}(0),$$

$$a_0 + a_1 T + a_2 T^2 + a_3 T^3 = s(T),$$

$$a_1 + 2a_2 T + 3a_3 T^2 = \dot{s}(T).$$

We can also write this in matrix form as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} s(0) \\ \dot{s}(0) \\ s(T) \\ \dot{s}(T) \end{bmatrix}, \tag{B.5}$$

and then invert our matrix to solve for our constant vector

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T & T^2 & T^3 \\ 0 & 1 & 2T & 3T^2 \end{bmatrix}^{-1} \begin{bmatrix} s(0) \\ \dot{s}(0) \\ s(T) \\ \dot{s}(T) \end{bmatrix}. \tag{B.6}$$

Note that we need to select our terminal time, T. Once our constants and initial conditions are defined, we can populate (B.1) to (B.4) without issue. We did not have to start and stop in a stationary condition, but we notably have no control over how the acceleration profile evolves. If we wanted to have a say about the initial and final acceleration conditions, we would require a quintic polynomial. Given that our velocity profile from (B.2) is quadratic, we can find the time point when the maximum value occurs (at the turning point) as

$$t_{max} = -\frac{a_2}{3a_3}. \tag{B.7}$$

The corresponding maximum value then follows as

$$\dot{s}_{max} = \dot{s}(t_{max}) = a_1 + 2a_2 t_{max} + 3a_3 t_{max}^2 = a_1 - \frac{a_2^2}{3a_3}. \tag{B.8}$$

B.2 Polynomial splines

Polynomial splines, such as a cubic spline, is a method to find a smooth trajectory from a set of trajectory via points. Conceptually, if we were given a start point, and end point, and some finite amount of intermediate via points, then we can use polynomial interpolation between each set of adjacent points. These piece-wise polynomials are then blended together to form a single trajectory that is made of a set of polynomials. The result is a continuous trajectory that moves through each specified point.

In order to ensure that the full trajectory (from the start point to the end point) has some level of smoothness, we need to constrain the velocity such that it remains continuous at each via point (also known as *knots*). The trajectory generation problem would then require a sufficient amount of waypoint information, such as the required position and velocity at each waypoint, in order to construct a problem that is solvable.

This type of trajectory generation method is useful in applications that require surveying or task-specific motion. For example, if a quadrotor is required to inspect different specific location on a map, we are not too concerned with how it behaves between these specified locations (our waypoints), aside from the quadrotor to be energy efficient. So we can define a polynomial spline that intersects all the waypoints with a piece-wise set of polynomials.

We start by considering a generalised trajectory with N waypoints, and a start time and end time of $T_1 = 0$ and $T_N = T$, respectively. T_1 then corresponds to the starting point, whereas T_2 is the time requirement at the second point. The polynomial is individually assessed between each moving pair of via points. For example, we will define a polynomial between point 1 and point 2, followed by a different polynomial that joins point 2 and point 3, and so on. At each point $i \in \{1, ...N\}$, the user specifies both the desired position, $s(T_i)$ and velocity, $\dot{s}(T_i)$. The N waypoints results in $N-1$ polynomials to be determined, with a segment duration equal to $\Delta T_j = T_{j+1} - T_j$, where $j \in \{1, ...N-1\}$.

In the case of cubic polynomials, four parameters need to be determined in order to fully specify a segment trajectory, as seen in (B.1). This means that a path with N points will require $4(N-1)$ parameters to be determined in order to find all segment polynomials. However, as with Section B.1, we can solve for each segment as long as we adhere to the via point position and velocity constraints of

$$
\begin{aligned}
s(T_j) &= s_j, \\
s(T_j + \Delta T_j) &= s_{j+1}, \\
\dot{s}(T_j) &= \dot{s}_j, \\
\dot{s}(T_j + \Delta T_j) &= \dot{s}_{j+1},
\end{aligned}
\tag{B.9}
$$

where s_j and $\dot{s}_j$ are the user specified position and velocity requirements at point j. In other words, the end position and velocity of polynomial j is required to match the starting position and velocity of polynomial $j+1$. The trajectory during segment j follows as

$$
s(T_j + \Delta t) = a_{j_0} + a_{j_1}\Delta t + a_{j_2}\Delta t^2 + a_{j_3}\Delta t^3, \tag{B.10}
$$

where $0 \leq \Delta t \leq \Delta T_j$. Note that there is no loss of generality by expressing the independent variable as incremental time, Δt, instead of t, as in (B.1). The only difference is that the polynomial coefficients will change accordingly.

As in the previous Section, we can pose the constraints in (B.9) as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & \Delta T_j & \Delta T_j^2 & \Delta T_j^3 \\ 0 & 1 & 2\Delta T_j & 3\Delta T_j^2 \end{bmatrix} \begin{bmatrix} a_{j_0} \\ a_{j_1} \\ a_{j_2} \\ a_{j_3} \end{bmatrix} = \begin{bmatrix} s_j \\ \dot{s}_j \\ s_{j+1} \\ \dot{s}_{j+1} \end{bmatrix}, \tag{B.11}$$

and solve for $\{a_{j_0}, a_{j_1}, a_{j_2}, a_{j_3}\}$, which yields

$$\begin{aligned} a_{j_0} &= s_j, \\ a_{j_1} &= \dot{s}_j, \\ a_{j_2} &= \frac{3s_{j+1} - 3s_j - 2\dot{s}_j\Delta T_j - \dot{s}_{j+1}\Delta T_j}{\Delta T_j^2}, \\ a_{j_3} &= \frac{2s_j + (\dot{s}_j + \dot{s}_{j+1})\Delta T_j - 2s_{j+1}}{\Delta T_j^3}. \end{aligned} \tag{B.12}$$

The "efficiency" of the resulting blended trajectory will be dependent on how the via-point times and velocities are selected. Additionally, specifying velocity for intermediate points is not always obvious or intuitive. One can instead only specify the position and time for the via points, and then use heuristic methods to separately find an appropriate velocity that rationally links sequential via point positions (e.g. using simple linear interpolation).

Using (B.9), cubic splines are effective in ensuring continuous position and velocity trajectories, but there is no constraint on how acceleration behaves at each via point. If the trajectory requires a higher degree of smoothness (that reaches the jerk level for example) or if acceleration is an important quantity at each via point, then a quintic (fifth-order) polynomial can be used. This follows the same procedure as in the case of cubics, but we would then require six constraints to fully determine the six unknown polynomial parameters. This can be addressed by also imposing via point acceleration specifications.

B.3 Trapezoidal velocity profiles

While the cubic polynomial trajectories are appealing from an energy efficient point of view, they are not time-optimal. This is exemplified by the fact that any imposed velocity constraint can at best only momentarily be met at equality momentarily (at the turning point). This means that for almost the entire trajectory, the robot is not being driven at its velocity limit. Similarly, the acceleration will only be maximised at either the start point or end point of the trajectory (or both simultaneously), implying that the acceleration is below its theoretical maximum for every other point on the trajectory. This obviously means that we are not dealing with the time-optimal trajectory given the robot's dynamic constraints.

A *trapezoidal velocity profile* is an alternative approach that blends different polynomial types together during a point to point manoeuvre. The trajectory is split up into three phases:

- Phase 1: Constant acceleration (*spool up*).
- Phase 2: Constant velocity (*cruise*).
- Phase 3: Constant deceleration (*spool down*).

The resulting position trajectory is made up of a linear segment (as a result of phase 2) that is connected on either side by quadratic segments (as a result of phase 1 and 3). Both the initial and final velocities are set to be zero, and the time duration of Phase 1 and Phase 3 are equal.

To ensure the transition from s_0 to s_T in the timespan of T, the trajectory needs to satisfy certain constraints. The first constraint must enforce that the velocity achieved at the end of Phase 1 be equal to that of Phase 2:

$$\ddot{s}_{max} t_c = \dot{s}_{max}(t_c) = \frac{s_m - s_c}{t_m - t_c}, \tag{B.13}$$

where s_c is the position value on the trajectory at the point where Phase 1 and Phase 2 meet (the end of the parabolic segment at t_c), and $s_m = \frac{s_0 + s_T}{2}$. Double integrating the acceleration profile during Phase 1, s_c can be expressed as

$$s_c = s_0 + \frac{1}{2}\ddot{s}_{max} t_c^2. \tag{B.14}$$

Combining (B.13) and (B.14) results in

$$\ddot{s}_{max} t_c^2 - \ddot{s}_{max} T t_c + s_T - s_0 = 0. \tag{B.15}$$

The Phase 1 time duration can then be found by solving the quadratic equation in (B.15)

$$t_c = \frac{T}{2} - \frac{1}{2}\sqrt{\frac{T^2 \ddot{s}_{max} - 4(s_T - s_0)}{\ddot{s}_{max}}}, \tag{B.16}$$

where $t_c \leq T/2$ is required to facilitate the symmetric profile. Note that based on the structure of (B.16), this will always be the case. If $\ddot{s}_{max}$ is additionally constrained such that $\text{sgn}(\ddot{s}_{max}) = \text{sgn}(s_T - s_0)$, then a real-valued t_c can be determined if

$$|\ddot{s}_{max}| \geq \frac{4|s_T - s_0|}{T^2}. \tag{B.17}$$

If (B.17) is satisfied at equality, the resulting trajectory will not feature a cruise stage. More specifically, there will be a single time value, $t_c = T/2$, that corresponds to the aforementioned velocity constraint, and this results in the trajectory only having a *spool up* stage and a *spool down* stage. Visually, this is shown as a triangular profile. On the other hand, if (B.17) is satisfied in some arbitrary sense, then two solutions for t_c will exist, which will define the start and end points of the Phase 2 - the *cruise* stage.

If we specify s_0, s_T, and T, and by extension, $\dot{s}_{max}$, the constraint in (B.17) can be used to limit the acceleration command based on the physical limitation of the platform. In other words, selecting a valid $\ddot{s}_{max}$ can then be used to populate (B.16). The resulting piece-wise position trajectory follows as

$$s(t) = \begin{cases} s_0 + \frac{1}{2}\ddot{s}_{max}t^2 & 0 \leq t \leq t_c \\ s_0 + \ddot{s}_{max}t_c(t - t_c/2) & t_c < t \leq T - t_c \\ s_T - \frac{1}{2}\ddot{s}_{max}(T - t)^2 & T - t_c \leq t \leq T \end{cases}. \tag{B.18}$$

We can therefore define a trajectory that demands the maximum actuator acceleration. We can similarly repose the setup of the problem to focus on the maximum velocity. Taking into account that $\dot{s}_{max} = \ddot{s}_{max}t_c$, (B.15) can be reposed as

$$\dot{s}_{max}t_c - \dot{s}_{max}T + s_T - s_0 = 0. \tag{B.19}$$

As in the case of (B.16), we solve for t_c, which gives us

$$t_c = \frac{s_0 - s_T + \dot{s}_{max}T}{\dot{s}_{max}}. \tag{B.20}$$

The constraint that ensures t_c is a positive-real follows as

$$\frac{s_T - s_0}{T} \leq \dot{s}_{max}. \tag{B.21}$$

We additionally require $t_c \leq T/2$, for sake of symmetry, which leads us to the other constraint of

$$\frac{s_0 - s_T + \dot{s}_{max}T}{\dot{s}_{max}} \leq \frac{T}{2}, \tag{B.22}$$

or more concisely,

$$\dot{s}_{max} \leq 2\frac{s_T - s_0}{T}. \tag{B.23}$$

Collating the results from (B.21) and (B.23), the constraint on $\dot{s}_{max}$ can be written as

$$\frac{s_T - s_0}{T} \leq \dot{s}_{max} \leq 2\frac{s_T - s_0}{T}. \tag{B.24}$$

As in the case of the acceleration limit, we can select a valid $\dot{s}_{max}$ using (B.24), which will then inform the selection of t_c in (B.20). Lastly, we can calculate the corresponding maximum acceleration from (B.19) as

$$\ddot{s}_{max} = \frac{s_0 - s_T + \dot{s}_{max}T}{t_c^2}, \tag{B.25}$$

which can then be used to populate our piece-wise position trajectory in (B.18).

In summary, depending on the particular constraint, be it velocity or acceleration, this limit can be directly imposed on the trajectory, and this will allow the system to theoretically be driven at its maximum capability for as long as possible (within the kinematically feasible task space).

Bibliography

[1] TED, “The astounding athletic power of quadcopters - raffaello d’andrea: Ted talks,” https://www.youtube.com/watch?v=w2itwFJCgFQ, [Online; accessed 6-February-2016].

[2] M. Cutler and J. P. How, “Actuator constrained trajectory generation and control for variable-pitch quadrotors,” in *AIAA Guidance, Navigation, and Control Conference*, 2012, pp. 1–15.

[3] M. Cutler, N. K. Ure, B. Michini, and J. P. How, “Comparison of fixed and variable pitch actuators for agile quadrotors,” in *AIAA Conf. on Guidance, Navigation and Control, Portland, OR*, 2011.

[4] J. Elso, M. Gil-Martínez, and M. García-Sanz, “Quantitative feedback–feedforward control for model matching and disturbance rejection,” *IET Control Theory & Applications*, vol. 7, no. 6, pp. 894–900, 2013.

[5] J. Elso, M. Gil-Martinez, and M. Garcia-Sanz, “A quantitative feedback solution to the multivariable tracking error problem,” *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2331–2346, 2014.

[6] D. Limebeer, “The specification and purpose of a controller design case study,” in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE, 1991, pp. 1579–1580.

[7] G. Hoffmann, S. Waslander, and C. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering,” *2009 IEEE International Conference on Robotics and Automation*, pp. 3277–3282, May 2009.

[8] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, May 2011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980409

[9] M. Cutler and J. How, “Actuator Constrained Trajectory Generation and Control for Variable-Pitch Quadrotors,” *AIAA Guidance, Navigation, and Control Conference*, pp. 1–13, Aug. 2012.

[10] M. Cutler, Ure, B. Michini, and J. P. How, “Comparison of Fixed and Variable Pitch Actuators for Agile Quadrotors,” *In AIAA Guidance, Navigation, and Control Conference (GNC)*, pp. 1–17, 2012.

[11] I. Horowitz, “Invited paper survey of quantitative feedback theory (qft),” *International Journal of control*, vol. 53, no. 2, pp. 255–291, 1991.

[12] ——, “Quantitative feedback theory,” in *IEE Proceedings D (Control Theory and Applications)*, vol. 129, no. 6. IET, 1982, pp. 215–226.

[13] F. Bailey and C.-H. Hui, “Cacsd tools for loop gain-phase shaping design of siso robust controllers,” in *IEEE Control Systems Society Workshop on Computer-Aided Control System Design*. IEEE, 1989, pp. 151–157.

[14] J. C. Moreno, A. Baños, and M. Berenguel, “The design of qft robust compensators with magnitude and phase specifications,” *Mathematical Problems in Engineering*, vol. 2010, 2010.

[15] E. Eitelberg, “Quantitative feedback design for tracking error tolerance,” *Automatica*, vol. 36, no. 2, pp. 319–326, 2000.

[16] W. Wu and S. Jayasuriya, “A new qft design methodology for feedback systems under input saturation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 2, pp. 225–232, 2001.

[17] C. J. Pritchard and B. Wigdorowitz, “On the determination of time-domain signal levels at the specification stage in quantitative feedback theory controller synthesis,” *International Journal of Control*, vol. 66, no. 2, pp. 329–348, 1997.

[18] E. Boje, “Pre-filter design for tracking error specifications in qft,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 13, no. 7, pp. 637–642, 2003.

[19] ——, “Quantitative digital design of crossfeed and feedback controllers for the uh-60 black hawk helicopter,” in *International Symposium on Quantitative Feedback Theory and Robust Frequency Domain Methods*. University of Natal Durban, 1999, pp. 161–178.

[20] A. Pretorius and E. Boje, “A complementary quantitative feedback theory solution to the 2× 2 tracking error problem,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 16, pp. 6569–6584, 2020.

[21] ——, “Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 12 208–12 213, 2014.

[22] ——, “Shutter delay estimation for a low-cost stand-alone visual tracking system,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 441–11 446, 2017.

[23] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, “A survey of quadrotor unmanned aerial vehicles,” in *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012, pp. 1–6.

[24] S. Bouabdallah and R. Siegwart, “Design and control of a miniature quadrotor,” in *Advances in unmanned aerial vehicles*. Springer, 2007, pp. 171–210.

[25] Y. Bai and S. Gururajan, “Evaluation of a baseline controller for autonomous “figure-8” flights of a morphing geometry quadcopter: Flight performance,” *Drones*, vol. 3, no. 3, p. 70, 2019.

[26] A. Tayebi and S. McGilvray, “Attitude stabilization of a vtol quadrotor aircraft,” *IEEE Transactions on control systems technology*, vol. 14, no. 3, pp. 562–571, 2006.

[27] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment,” in *AIAA guidance, navigation and control conference and exhibit*, 2007, p. 6461.

[28] B. Michini, J. Redding, N. K. Ure, M. Cutler, and J. P. How, “Design and flight testing of an autonomous variable-pitch quadrotor,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2978–2979.

[29] A. Abhishek, A. Duhoon, M. Kothari, S. Kadukar, L. Rane, and G. Suryavanshi, “Design, development, and closed-loop flight-testing of a single power plant variable pitch quadrotor unmanned air vehicle,” in *Proceedings of 73rd American Helicopter Society Annual Forum*, 2017, pp. 9–11.

[30] T. Pang, K. Peng, F. Lin, and B. M. Chen, “Towards long-endurance flight: Design and implementation of a variable-pitch gasoline-engine quadrotor,” in *2016 12th IEEE International Conference on Control and Automation (ICCA)*. IEEE, 2016, pp. 767–772.

[31] CharmedLabs. (2016) Pixycam product page. [Online]. Available: https://charmedlabs.com/default/pixy-cmucam5/

[32] J.-Y. Bouguet. (2015) Camera calibration toolbox for matlab. [Online]. Available: https://www.vision.caltech.edu/bouguetj/

[33] C. Sanderson and R. Curtin, “Armadillo: a template-based c++ library for linear algebra,” *Journal of Open Source Software*, vol. 1, no. 2, p. 26, 2016.

[34] B. Erginer and E. Altuğ, “Modeling and pd control of a quadrotor vtol vehicle,” in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 894–899.

[35] T. Luukkonen, “Modelling and control of quadcopter,” *Independent research project in applied mathematics, Espoo*, 2011.

[36] R. Beard, “Quadrotor dynamics and control rev 0.1,” 2008.

[37] N. L. Johnson and K. K. Leang, “Enhanced proportional-derivative control of a micro quadcopter,” in *ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013, pp. V001T01A005–V001T01A005.

[38] J. Kim, M.-S. Kang, and S. Park, “Accurate modeling and robust hovering control for a quad-rotor vtol aircraft,” in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer, 2009, pp. 9–26.

[39] Z. Zuo, “Trajectory tracking control design with command-filtered compensation for a quadrotor,” *Control Theory & Applications, IET*, vol. 4, no. 11, pp. 2343–2355, 2010.

[40] I. Sa and P. Corke, “System identification, estimation and control for a cost effective open-source quadcopter,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2202–2209.

[41] P.-J. Bristeau, P. Martin, E. Salaun, and N. Petit, “The role of propeller aerodynamics in the model of a quadrotor uav,” in *Control Conference (ECC), 2009 European*. IEEE, 2009, pp. 683–688.

[42] M. Bangura, R. Mahony *et al.*, “Nonlinear dynamic modeling for high performance control of a quadrotor,” 2012.

[43] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, “Nonlinear feedback control of quadrotors exploiting first-order drag effects,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189–8195, 2017.

[44] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain, “Quadrotors and accelerometers: State estimation with an improved dynamic model,” *IEEE Control Systems*, vol. 34, no. 1, pp. 28–41, 2014.

[45] M. R. Gharib and M. Moavenian, “Full dynamics and control of a quadrotor using quantitative feedback theory,” *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 2015.

[46] H. Liu, X. Wang, and Y. Zhong, “Quaternion-based robust attitude control for uncertain robotic quadrotors,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 406–415, 2015.

[47] H. Liu, D. Li, Z. Zuo, and Y. Zhong, “Robust three-loop trajectory tracking control for quadrotors with multiple uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2263–2274, 2016.

[48] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[49] K. Chaney and A. Eggers Jr, “Expanding wake induction effects on thrust distribution on a rotor disc,” *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 5, no. 2-3, pp. 213–226, 2002.

[50] J. Anderson, *EBOOK: Fundamentals of Aerodynamics (SI units)*. McGraw hill, 2011.

[51] G. D. Padfield, *Helicopter Flight Dynamics*, G. D. Padfield, Ed. Oxford, UK: Blackwell Publishing Ltd, Jan. 2007.

[52] W. Lord and J. H. Hwang, "Dc servomotors-modeling and parameter determination," *IEEE Transactions on Industry Applications*, no. 3, pp. 234–243, 1977.

[53] N. S. Nise, *Control systems engineering*. John Wiley & Sons, 2020.

[54] L. Ljung and R. Singh, "Version 8 of the matlab system identification toolbox," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 1826–1831, 2012.

[55] A. Nemra and N. Aouf, "Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering," 2009.

[56] C. Fu, A. Carrio, and P. Campoy, "Efficient visual odometry and mapping for unmanned aerial vehicle using arm-based stereo vision pre-processing system," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 957–962.

[57] F. Gao and S. Shen, "Online quadrotor trajectory generation and autonomous navigation on point clouds," in *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 139–146.

[58] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an rgb-d camera," in *Robotics Research*. Springer, 2017, pp. 235–252.

[59] S. Omari and G. Ducard, "Metric visual-inertial navigation system using single optical flow feature," in *Control conference (ECC)*, 2013, pp. 1310–1316.

[60] J. G. Richards, "The measurement of human motion: a comparison of commercially available systems," *Human movement science*, vol. 18, no. 5, pp. 589–602, 1999.

[61] M. Windolf, N. Götzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems,exemplified on the vicon-460 system," *Journal of biomechanics*, vol. 41, no. 12, pp. 2776–2780, 2008.

[62] ViconMotionCaptureSystemsLtd. (2016) Vicon vantage product page. [Online]. Available: https://www.vicon.com/products/camera-systems/

[63] M. Achtelik, T. Zhang, K. Kuhnlenz, and M. Buss, "Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors," in *2009 International Conference on Mechatronics and Automation*. IEEE, 2009, pp. 2863–2869.

[64] Y. Reddi and E. Boje, "System identification for low-cost small-scale helicopters," in *World Congress*, vol. 19, no. 1, 2014, pp. 8831–8836.

[65] A. Schmitz, M. Ye, R. Shapiro, R. Yang, and B. Noehren, "Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system," *Journal of biomechanics*, vol. 47, no. 2, pp. 587–591, 2014.

[66] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[67] R. G. Brown and P. Y. Hwang, "Introduction to random signals and applied kalman filtering: with matlab exercises and solutions," *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, vol. 1, 1997.

[68] P. I. Corke and O. Khatib, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.

[69] E. EITELBERG, "Sampling rate design based on (1- st/2)," *International Journal of Control*, vol. 48, no. 4, pp. 1423–1432, 1988.

[70] C. Borghesani, Y. Chait, and O. Yaniv, "The qft frequency domain control design toolbox," *See also*, 2003.

[71] E. Boje, "Quantitative design of gain scheduled controllers for plants with measurable transport rates," *IFAC Proceedings Volumes*, vol. 23, no. 8, pp. 173–176, 1990.

[72] F. G. Shinskey, *Feedback controllers for the process industries*. McGraw-Hill Professional, 1994.

[73] A. Pretorius and E. Boje, "Robust plant by plant control design using model-error tracking sets," *International Journal of Robust and Nonlinear Control*, 2019.

[74] ——, "A refinement approach to the multivariable tracking error problem," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 12, pp. 7016–7038, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.6183

[75] I. M. Horowitz and M. Sidi, "Synthesis of feedback systems with large plant ignorance for prescribed time-domain tolerances," *International Journal of Control*, vol. 16, no. 2, pp. 287–309, 1972.

[76] O. Yaniv and I. M. Horowitz, "A quantitative design method for mimo linear feedback systems having uncertain plants," *International Journal of Control*, vol. 43, no. 2, pp. 401–421, 1986.

[77] I. Horowitz, "Survey of quantitative feedback theory (qft)," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 11, no. 10, pp. 887–921, 2001.

[78] O. Yaniv, *Quantitative feedback design of linear and nonlinear control systems*. Springer Science & Business Media, 2013, vol. 509.

[79] C. H. Houpis, *Quantitative feedback theory: fundamentals and applications*. CRC press, 1999.

[80] O. Yaniv, *Quantitative feedback design of linear and nonlinear control systems*. Springer Science & Business Media, 1999.

[81] J. Elso, M. Gil-Martinez, and M. Garcia-Sanz, "Quantitative feedback control for multivariable model matching and disturbance rejection," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 1, pp. 121–134, 2017.

[82] I. Horowitz, "Quantitative synthesis of uncertain multiple input-output feedback system," *International Journal of Control*, vol. 30, no. 1, pp. 81–106, 1979.

[83] H. Rosenbrock, "Design of multivariable control systems using the inverse nyquist array," in *Proceedings of the Institution of Electrical Engineers*, vol. 116, no. 11. IET, 1969, pp. 1929–1936.

[84] J. M. Maciejowski, "Multivariable feedback design," *Electronic systems engineering series*, 1989.

[85] A. M. Ostrowski, "Note on bounds for determinants with dominant principal diagonal," *Proceedings of the American Mathematical Society*, vol. 3, no. 1, pp. 26–30, 1952.

[86] H. H. Rosenbrock, "State-space and multivariable theory," *(No Title)*, 1970.

[87] D. Q. Mayne, "The design of linear multivariable systems," *Automatica*, vol. 9, no. 2, pp. 201–207, 1973.

[88] A. Mees, "Achieving diagonal dominance," *Systems & Control Letters*, vol. 1, no. 3, pp. 155–158, 1981.

[89] D. Hawkins, "'pseudodiagonalisation'and the inverse-nyquist array method," in *Proceedings of the Institution of Electrical Engineers*, vol. 119, no. 3. IET, 1972, pp. 337–342.

[90] A. MacFarlane and J. Belletrutti, "The characteristic locus design method," *Automatica*, vol. 9, no. 5, pp. 575–588, 1973.

[91] A. MacFarlane and B. Kouvaritakis, "A design technique for linear multivariable feedback systems," *International Journal of Control*, vol. 25, no. 6, pp. 837–874, 1977.

[92] B. Kouvaritakis, "Characteristic locus methods for multivariable feedback systems design." Ph.D. dissertation, University of Manchester Institute of Science and Technology, 1974.

[93] J. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 4–16, 1981.

[94] E. Boje, “Non-diagonal controllers in mimo quantitative feedback design,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 12, no. 4, pp. 303–320, 2002.

[95] G. Zames, “Optimal sensitivity and feedback: weighted seminorms, approximate inverses, and plant invariant schemes,” 1979.

[96] J. Doyle *et al.*, “Lecture notes for onr/honeywell workshop on advances in multivariable control,” *Minneapolis, MN*, 1984.

[97] G. Balas, R. Chiang, A. Packard, and M. Safonov, “Robust control toolbox user’s guide,” *The Math Works, Inc., Tech. Rep*, 2007.

[98] D. J. Limebeer, E. M. Kasenally, and J. D. Perkins, “On the design of robust two degree of freedom controllers,” *Automatica*, vol. 29, no. 1, pp. 157–168, 1993.

[99] C. Foias, B. Francis, J. W. Helton, H. Kwakernaak, J. B. Pearson, and H. Kwakernaak, “The polynomial approach to h-optimal regulation,” in *H-Control Theory: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (CIME) held in Como, Italy, June 18–26, 1990.* Springer, 1991, pp. 141–221.

[100] M. Green and D. J. Limebeer, *Linear robust control.* Courier Corporation, 2012.

[101] C. A. Desoer and M. Vidyasagar, *Feedback systems: input-output properties.* SIAM, 2009.

[102] D. F. Thompson and O. D. Nwokah, “Frequency response specifications and sensitivity functions in quantitative feedback theory,” in *1991 American Control Conference.* IEEE, 1991, pp. 2015–2020.

[103] J. Elso, M. Gil-Martínez, and M. Garcia-Sanz, “Nonconservative qft bounds for tracking error specifications,” *International journal of robust and nonlinear control*, vol. 22, no. 18, pp. 2014–2025, 2012.

[104] M. A. Franchek and P. A. Herman, “Direct connection between time-domain performance and frequency-domain characteristics,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 8, no. 12, pp. 1021–1042, 1998.

[105] I. Horowitz, “A synthesis theory for a class of saturating systems,” *International Journal of Control*, vol. 38, no. 1, pp. 169–187, 1983.

[106] J. C. Moreno, J. L. Guzmán, A. Baños, and M. Berenguel, “The input amplitude saturation problem in qft: A survey,” *Annual Reviews in Control*, vol. 35, no. 1, pp. 34–55, 2011.

[107] P. Hippe and H. Stahl, “Note on the relation between pole-zero location and input-output behaviour of siso systems and its influence on controller design,” *International Journal of Control - INT J CONTR*, vol. 45, pp. 1469–1477, 04 1987.

[108] "Matlab optimization toolbox," 2015.

[109] E. Boje, "Algorithm for calculating mimo qft tracking bounds," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, pp. 697–699, 2004.

[110] Y. Chait, C. Borghesani, and Y. Zheng, "Single-loop qft design for robust performance in the presence of non-parametric uncertainties," *Journal of dynamic systems, measurement, and control*, vol. 117, no. 3, pp. 420–425, 1995.

[111] I. M. Horowitz, *Synthesis of feedback systems*. Elsevier, 2013.

[112] E. Boje, "Multivariable quantitative feedback design for tracking error specifications," *Automatica*, vol. 38, no. 1, pp. 131–138, 2002.

[113] S. M. Alavi, A. Khaki-Sedigh, B. Labibi, and M. Hayes, "Improved multivariable quantitative feedback design for tracking error specifications," *IET Control Theory & Applications*, vol. 1, no. 4, pp. 1046–1053, 2007.

[114] M. Garcia-Sanz, *Robust control engineering: practical QFT solutions*. CRC Press, 2017.

[115] O. Yaniv and Y. Chait, "A simplified multi-input multi-output formulation for the quantitative feedback theory," 1992.

[116] J. Rico and M. Gil-Martinez, "Multivariable qft robust control of a heat exchanger," in *2011 19th Mediterranean Conference on Control & Automation (MED)*. IEEE, 2011, pp. 588–593.

[117] U. Shaked, I. Horowitz, and S. Golde, "Synthesis of multivariable, basically non-interacting systems with significant plant uncertainty," *Automatica*, vol. 12, no. 1, pp. 61–71, 1976.

[118] J. Elso and J. X. Ostolaza, "Automatic synthesis of feedforward elements in quantitative feedback theory," *International Journal of Robust and Nonlinear Control*, 2021.

[119] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, interpolation and animation*. Citeseer, 1998, vol. 2.