Quadrotor Control in the Presence of Unknown Mass Properties

by

Rikky Ricardo Petrus Rufino Duivenvoorden

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

Quadrotor Control in the Presence of Unknown Mass Properties

Rikky Ricardo Petrus Rufino Duivenvoorden
Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto
2016

Quadrotor UAVs are popular due to their mechanical simplicity, as well as their capability to hover and vertically take-off and land. As applications diversify, quadrotors are increasingly required to operate under unknown mass properties, for example as a multirole sensor platform or for package delivery operations. The work presented here consists of the derivation of a generalized quadrotor dynamic model without the typical simplifying assumptions on the first and second moments of mass. The maximum payload capacity of a quadrotor in hover, and the observability of the unknown mass properties are discussed. A brief introduction of $\mathcal{L}_1$ adaptive control is provided, and three different $\mathcal{L}_1$ adaptive controllers were designed for the Parrot AR.Drone quadrotor. Their tracking and disturbance rejection performance was compared to the baseline nonlinear controller in experiments. Finally, the results of the combination of $\mathcal{L}_1$ adaptive control with iterative learning control are presented, showing high performance trajectory tracking under uncertainty.

Acknowledgements

The work presented in this thesis would not have been possible without the guidance and support of other people. I would foremost like to thank Prof. Angela Schoellig for giving me the opportunity to explore this area of research and for her guidance and valuable advice along the way. She is readily able to provide insight to problems from an alternative perspective, and her eye for detail leads to constructive and comprehensive feedback. Her ambition for novel results has been a great motivation.

I would also like to take this opportunity to thank the additional two members of my Research Assessment Committee, Prof. Jonathan Kelly and Prof. Timothy Barfoot, for their insightful questions and feedback. I am also grateful to Prof. Gabriele D'Eleuterio for reviewing this thesis. Our discussions were beneficial to the derivation of the quadrotor dynamics model.

The graduate students in the student room from the STARS and DSL laboratories have also had their contribution. Thank you to Valentin, Lee, Jacob, Andreas (Andi), Thomas, Kaizad, Felix and Thanh for enriching my overall experience at the University of Toronto Institute for Aerospace Studies (UTIAS). I would like to specifically acknowledge Karime Pereida with whom I have spent numerous hours integrating Iterative Learning Control with $\mathcal{L}_1$ adaptive control. I would also like to thank Mohamed Helwa for his discussions and feedback that have inspired some of the theoretical results in this thesis, as well as Chris McKinnon for his assistance whenever I needed help solving issues with ROS.

My extracurricular activities have greatly shaped the person that I have now become. Without my involvement in the University of Toronto Aerospace Team (UTAT) since my first year of undergraduate studies in Engineering Science, I would not have generated an interest in unmanned aerial vehicles that forms the basis and motivation behind this research. The countless hours consumed by troubleshooting and learning from failures, as well as the time spent at the drawing board, in fabrication rooms, at test flights regardless of Canadian temperature, and at competitions have formed vivid memories that will stay with me for the rest of my life. I had a remarkable experience, met and worked with exceptional people, and learned, invented and tried many new things in the fields of aircraft design and unmanned aerial systems. This thesis marks an end to my enduring commitment as I seek and pursue new endeavours.

Finally, I would like thank my family. My parents Jos and Lilet Duivenvoorden and my sister Rizell Duivenvoorden have supported me throughout my journey. They have challenged, encouraged and motivated me on the path I chose to embark; the path that I continue to traverse and reshape to this day. Thank you.

Contents

1 Introduction **1**

1.1 Overview and Motivation 1

1.2 Project Scope and Objectives 2

2 Related Work **3**

2.1 Quadrotor Dynamics Model 3

2.2 Quadrotor Control Techniques Under Uncertainty 4

2.2.1 Standard Control Techniques 4

2.2.2 System Identification for UAVs 5

2.2.3 Online Parameter Estimation 6

2.2.4 Robust Control Techniques 6

2.2.5 Adaptive Control For Quadrotor UAVs 7

2.3 Payload Carrying Methods 8

3 Contributions **9**

3.1 Novel Contributions of This Work 9

3.2 Outline 9

4 General Quadrotor Dynamics Model **11**

4.1 Preliminaries 11

4.1.1 Time Rate of Change of Vectors 12

4.1.2 Referential Form and Rotation of Vectors 12

4.1.3 Euler Angle Rates 13

4.1.4 Cross Product 14

4.1.5 Referential Form of Twain Vectors and the Cross Operator 14

4.2 Quadrotor Dynamic Model 15

4.2.1 Quadrotor Kinematics 15

4.2.2 Linear and Angular Momentum 15

4.2.3 Translational Equations of Motion . . . 17
4.2.4 Rotational Equations of Motion . . . 18
4.2.5 Combined Translational and Rotational Dynamics . . . 19
4.2.6 External Forces and Torques . . . 21
4.2.7 Quadrotor Nonlinear State Equation in Control Affine Form . . . 23
4.3 Analysis of the Dynamics Model . . . 24
4.3.1 Equilibria of the Equations of Motion . . . 24
4.3.2 Quadrotor Load Limits Under Design Constraints . . . 25

5 Nonlinear Observability Analysis 27
5.1 Preliminaries . . . 27
5.2 Augmented Control-Affine State Equation . . . 28
5.3 Nonlinear Observability of the Augmented State Equation . . . 30

6 $\mathcal{L}_1$ Adaptive Control 31
6.1 Preliminaries . . . 31
6.1.1 State or Output Predictor . . . 32
6.1.2 Projection Based Adaptation Law . . . 33
6.1.3 Piecewise Constant Parameter Adaptation . . . 34
6.1.4 Control Law . . . 34
6.1.5 Systems Theory and Norms . . . 35
6.2 Quadrotor $\mathcal{L}_1$ Adaptive Control Architectures . . . 36
6.2.1 Translational Velocity Control Approach . . . 37
6.2.2 Position Control Approach . . . 38
6.2.3 Attitude Control Approach . . . 39
6.3 Proof of the Stability and Transient Behaviour of the Closed-Loop $\mathcal{L}_1$ Adaptive Output Feedback Controller . . . 40
6.3.1 Problem Formulation . . . 40
6.3.2 $\mathcal{L}_1$ Adaptive Controller Definitions and $\mathcal{L}_1$-Norm Condition . . . 41
6.3.3 Revised Closed-loop $\mathcal{L}_1$ Adaptive Control Architecture . . . 43
6.3.4 Transient and Steady-State Performance . . . 45
6.4 Guideline for Tuning an $\mathcal{L}_1$ Adaptive Controller . . . 50

7 Experimental Results 53
7.1 Experimental Setup . . . 53
7.1.1 Baseline Nonlinear Controller . . . 53
7.1.2 Indoor Robotics Lab . . . 55

7.2 Comparison of $\mathcal{L}_1$ Adaptive Controllers 56
7.2.1 Step Input 56
7.2.2 Compensation of Mass Disturbance in Hover 59
7.3 Package Carrying Load Trajectory 61
7.4 Closed-Loop $\mathcal{L}_1$ Output Feedback Combined Iterative Learning Control . 62
7.4.1 Learning Convergence 63
7.4.2 Repeatability 64
7.4.3 Performance under Changing Conditions 65
7.5 Continuity of the $\mathcal{L}_1$ Adaptive Control Tuning Parameters 67

8 Concluding Remarks and Recommendations **69**
8.1 Conclusion 69
8.2 Future Work 70

Bibliography **71**

A Nonlinear Observability Analysis Script **76**

List of Tables

7.1 Average position error for the three $\mathcal{L}_1$ adaptive control and baseline nonlinear control architectures. The errors for both the step input and the disturbance under hover are shown here. 58

7.2 For the two controllers when the mass disturbance is applied after iteration ten: errors at iteration 10 and iteration 11 for a single set and average error at iteration 11 and maximum standard deviation of the five ten-iteration sets. 67

List of Figures

1.1 Project overview. 2

2.1 Coordinate system and reference frames from [1]. 3

4.1 Inertial and body-fixed 3D frames and points of reference. 12
4.2 Mass limits due to actuator limits. 26

6.1 General $\mathcal{L}_1$ adaptive output feedback controller with input disturbances. 32
6.2 $\mathcal{L}_1$ adaptive output feedback controller in velocity. 38
6.3 Roll, pitch and climb rate based $\mathcal{L}_1$ adaptive output feedback controller. 39
6.4 Closed-loop $\mathcal{L}_1$ adaptive output feedback controller. 43

7.1 UTIAS Indoor Robotics Lab setup showing the components that constitute the complete control loop. 55
7.2 Parrot AR.Drone 2.0 showing extruded polypropylene hull and spherical Vicon reflective markers. A clamp is also shown attached to the aft-right landing leg to simulate a mass disturbance. 56
7.3 Tracking performance comparison of the position-based $\mathcal{L}_1$ (blue), velocity-based $\mathcal{L}_1$ (purple), attitude-based $\mathcal{L}_1$ (yellow) and nonlinear controller (green) in response to a step input (red). 57
7.4 Disturbance rejection performance comparison of the position-based $\mathcal{L}_1$ (blue), velocity-based $\mathcal{L}_1$ (purple), attitude-based $\mathcal{L}_1$ (yellow) and nonlinear controller (green) in response to a sudden change in mass properties at time $t = 0$. The quadrotor is required to maintain hover so the desired trajectory (red) is a constant. 60
7.5 Horizontal package delivery trajectory tracking performance of the nonlinear controller(green), and the velocity based $\mathcal{L}_1$ adaptive controller (purple) carrying an unknown payload. The desired feasible trajectory is indicated by the red curve. 61
7.6 Closed-loop $\mathcal{L}_1$ adaptive output feedback controller. 62

7.7 Repeatability of the learned trajectory at iteration 10 after a mass disturbance has been applied to the system. The error in the PD-ILC approach increases dramatically while the error in the $\mathcal{L}_1$-ILC approach remains close to the error before the disturbance. 64
7.8 Mean of the error across five 10-iteration sets testing repeatability of the learned trajectory after a mass disturbance has been applied to the system. The PD-ILC approach displays a significantly larger standard deviation than the $\mathcal{L}_1$-ILC approach. 65
7.9 Learning behavior of the PD-ILC and $\mathcal{L}_1$-ILC frameworks after a mass disturbance is applied to the system at the end of iteration ten. 66
7.10 Contour plot of the cost function J($\mathbf{y}$, $\mathbf{u}$), evaluated for different values of $\omega_{x,y}$ and $m_{x,y}$. 68

Chapter 1

Introduction

1.1 Overview and Motivation

The number of applications for Unmanned Aerial Vehicles (UAVs) has grown tremendously over the last decade. Historically, these were used primarily by the military for aerial surveillance and reconnaissance, but today the UAV industry also targets civil and commercial markets including disaster management, search and rescue, agriculture and film/photography [2]. Nonmilitary applications of UAV technology is the fastest growing sector in the global aerospace industry and is expected to grow by 700% between 2012 and 2018 [3]. Within the next 10 years, [4] anticipates US$90 billion to be spent on the UAV industry worldwide, and [5] expects the United States to spend US$82 billion with the creation of an additional 100,000 jobs by 2025.

The quadrotor UAV has gained popularity due to its Vertical Takeoff and Landing (VTOL) and hovering capabilities. Its simple mechanical design is comprised of four motors for lift and control, mounted on the ends of four arms meeting at the center of the airframe. Lightweight construction and recent developments in battery technology allow for range and endurance of up to 50 minutes [6], sufficient for a broad range of applications.

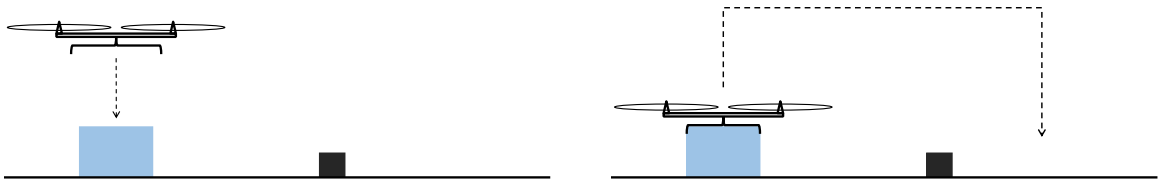
The applications of the quadrotor UAVs are primarily determined by the wide variety of payloads that can be carried; from cameras for surveillance and photography, to gripping mechanisms for carrying packages or other objects. Particularly interesting applications include parcel package delivery (e.g., from Amazon.com and DHL), and UAVs with modular sensor payload capabilities (e.g., from Aeryon Labs Inc.). The challenge associated with these applications is that the UAV mass, center of gravity (CG) and inertia change between or during flights, which imposes requirements on the safety and stability of the UAV. The controller performance will be dependent on how well the un-

known dynamics can be compensated for. Typical classical linear control techniques such as proportional-derivative-integral (PID) control can be inadequate to control the vehicle in these cases [7], so a safer more flexible approach in dynamic modelling and control is required for these applications.

1.2 Project Scope and Objectives

The scope of the project includes the derivation of a general dynamic model of the quadrotor without simplifying assumptions on the center of gravity location and inertia. Using this model, the dynamic behaviour of a quadrotor in atypical mass configurations can be analyzed, such as the payload mass limitations in hover. The model can also be used to determine the observability of the unknown mass properties, which will indicate whether controller design is feasible for a system with ten unknown mass properties. The project scope also includes the design of an adaptive controller for the control of a quadrotor UAV. After this development, the controller will be implemented in experiment on a Parrot AR.Drone 2.0 quadrotor and a quantitative comparison will be made with the baseline nonadaptive nonlinear controller previously used in the lab. Finally, the scope includes a successful demonstration of unknown payload transportation using this quadrotor.

The final high-level objective of this project is summarized in Figure 1.1.



(a) Obtain unknown package/payload. (b) Fly, and adapt the controller online.

Figure 1.1: Project overview.

Chapter 2

Related Work

2.1 Quadrotor Dynamics Model

The dynamics used to model a quadrotor typically use a rigid-body assumption. The coordinate system and frames of reference used in deriving the dynamic equations of motion are shown in Figure 2.1. For the standard quadrotor model [2], the following assumptions are usually made:

- The quadrotor frame is rigid and rotors are rigid, meaning there are no aeroelastic effects on the propellers. Gyroscopic effects are also omitted.
- The quadrotor center of gravity (CG) coincides with the geometric center of the quadrotor frame.
- The quadrotor frame is aligned with the principal axis frame, meaning that the off-diagonal terms in the second moment of inertia matrix are zero.

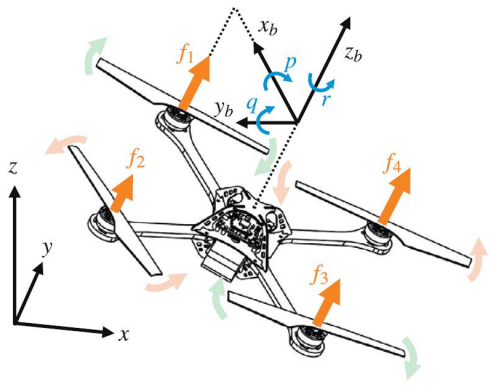


Figure 2.1: Coordinate system and reference frames from [1].

The second assumption has been omitted when uncertainties in CG offsets were used as unknown parameters [7, 8, 9]. The general full-inertia matrix is less widely used, where only [10], [8] and [9] assume the off-diagonal terms in the inertia matrix to be non-zero. Even among research that attempted to disturb the CG location by adding payload, the inertia matrix is assumed to be diagonal in the quadrotor aligned reference frame [11, 12]. The dynamic model of quadrotors to which a robotic arm was mounted has been treated differently. Since a moving robotic arm introduces changes in inertia and CG location, [13] and [14] make use of the Euler-Lagrange formulation using the kinetic and potential energy in both the quadrotor and each of the elements in the robotic arm. Other approaches have used a sequence of rotation matrices and application of the parallel-axis theorem to obtain the inertia matrix in the quadrotor frame taking into account each robot arm [15].

For this project the rigid-body assumption will be retained, so the dynamic model can be derived using the Newton-Euler formulation based on the time rate of change of linear and angular momenta. CG offsets can be taken into account as an additional vector from the geometric center of the quadrotor to the actual CG location. For second moment of inertia, the full matrix can be treated with a set of six unknown parameters.

2.2 Quadrotor Control Techniques Under Uncertainty

2.2.1 Standard Control Techniques

One standard method in quadrotor control is to linearize the nonlinear quadrotor model, and control it using linear controller design techniques such as linear quadratic regulator optimal control, as done by [2]. In experiment, this technique achieved good tracking in altitude, but not in roll and pitch. In [16], a PID controller was implemented to track desired altitude and horizontal position, while a proportional-derivative (PD) controller was used to track attitude angles of the quadrotor UAV. The mass of the UAV was assumed to be unknown and estimated for the altitude control loop, and this resulted in adequate altitude tracking. A similar PD controller was implemented by [7] for attitude control and found to diverge when CG location was perturbed. Feedback linearization, where the system dynamics are linearized through a transformation, also produced unstable results after perturbation of the CG location [7]. From these results, it is clear that a standard control technique such as PD or PID control applied to the nonlinear or linearized model does not satisfy stability requirements under mass, CG or inertia uncertainty. In [11] the uncompensated PID controller had approximately five times larger tracking errors in

altitude compared to the compensated controller using the mass estimate. This explains why adequate control was achieved by [16], since all mass properties except total mass were known before flight, and the controller was allowed to update according to the mass estimate.

For this project where mass, CG location and inertia are not known, a different control design must be implemented.

2.2.2 System Identification for UAVs

System identification is used to determine the mathematical model of unknown or unmodelled dynamics given a set of data. A particular model structure is typically assumed and the model parameters are chosen to best fit the data of the system [17]. System identification is relevant to this project since one method of achieving the objectives is to identify the dynamics of the vehicle model with unknown mass, inertia and offsets must be identified, after which the model can be used to design a controller. The system identification techniques used by [18] and [19] use batch Prediction Error Method (PEM) with the System Identification toolbox in MathWorks MATLAB, while [20] uses CIFER Flight Control Software in the frequency domain for system identification. In [18], the performance of different model structures was evaluated by comparing the identified heave, roll, pitch and yaw dynamics with the actual dynamic model of a quadrotor. Data used in system identification was simulated using the same dynamic model. It was found that the Auto Regressive Moving Average model with eXogenous inputs (ARMAX) model structure produced the closest matching results. On the other hand, [20] found the Output Error (OE) model structure more appropriate using fixed-wing UAV experimental data in identifying the roll and pitch dynamics. In addition, [19] performed off-line, time-domain system identification of a Helicopter UAV around hover, and used the identified yaw and heave dynamics in the design of a controller. Unfortunately, the system identification techniques used in [18, 19, 20] are all offline methods meaning that these techniques are based on data obtained beforehand.

For this project, this is not ideal because this time consuming process requires previous experimental data, which would need to be obtained for each payload or package. Furthermore, it is not practical to obtain experimental data and expect the offline hardware required for system identification to be present at each location the UAV is commanded to go.

2.2.3 Online Parameter Estimation

To overcome the disadvantages of offline batch estimation techniques discussed above, recursive parameter estimation algorithms can be used online and are discussed in [21]. Although [18] and [19] use Prediction Error Methods (PEM) offline in MATLAB, PEM can also be used recursively to estimate parameters online. A large number of online parameter estimation techniques used in literature are discussed in detail in [21]. The gradient-based method is one of the online PEM parameter estimation techniques that seeks to minimize a quadratic error based cost function by using the gradient to find the minimum. The gradient-based method is most widely used in mass and inertia estimation [16, 7], and in some cases in wind parameter estimation as well [22, 23]. The gradient-based parameter estimation is the simplest of the methods to implement and works well with constantly fluctuating parameters [21]. A least squares algorithm with exponential forgetting ensures that old parameter estimates are used with less weight in the estimation of current parameters, making this method appropriate for time-varying parameters as well [21]. This method is implemented by [11] where it is used to estimate mass, CG offset and inertias of a quadrotor UAV fitted with a grasping mechanism.

Although an online parameter estimation scheme can be used for the control of a quadrotor, it is limited to the parameters included within the design of the controller. Any disturbances and unmodelled dynamics influence the estimation of these parameters.

2.2.4 Robust Control Techniques

The limitations introduced above can be accounted for in robust control where stability is guaranteed in a specific parameter uncertainty envelope. Sliding mode control is a robust controller technique implemented in quadrotor control by [24], and is tested in simulation with 30% state and mass properties parameter deviations. Although a stable response is obtained, roll and pitch angles only stabilize after 20 seconds, resulting in large position errors. In [25], a robust controller is initially implemented to take into account the unknown total mass of the quadrotor. A small tracking error was present and slowly converged to zero, but much better performance was obtained when an adaptive controller with a mass estimation algorithm was used in the control loop. This shows the stability of the system under uncertainty using a robust controller, but also the limitations in performance. This is because the robust controller seeks to optimize performance over a relatively large range of unknown parameters. The work in [25] shows that a robust control technique augmented with an adaptive controller can successfully control a quadrotor under mass uncertainty. The main disadvantage of this approach is that it is

much more complicated to implement compared to either an adaptive or robust controller only.

In the application of quadrotor package delivery, robust control is able to compensate for the associated mass uncertainties, but in exchange for compromised tracking performance. This is not feasible for eventual full-scale implementation due to the extended endurance requirements which would not be necessary if high performance is achieved.

2.2.5 Adaptive Control For Quadrotor UAVs

For the limitations discussed in the previous two sections, the majority of previous work with unknown parameters have implemented an adaptive controller [8, 9, 11, 12, 14, 15]. The two main adaptive control techniques used are the model reference adaptive control (MRAC) and self-tuning control (STC) designs. MRAC compares the actual output with the output of a reference model, and from this the unknown parameters are estimated. On the other hand, STC estimates the parameters based on the control input and output of the plant [21]. The estimates of the unknown parameters are then used in the adaptation of the controller. MRAC is used in the motion control of a quadrotor equipped with a moving robotic arm [14, 15] but also under mass and inertia uncertainties [12], where the dynamic quadrotor model is used as the reference model. STC is successfully implemented in the control of a quadrotor UAV with an unknown payload mass [11] and with an unknown CG location in [9]. Both cases were validated in experiment.

Owing to the broad range in operating conditions of large-scale aircraft throughout their operating envelope, adaptive control techniques have been developed to maximize control performance under all operating circumstances [21]. The $\mathcal{L}_1$ adaptive controller was designed to improve the performance of classical adaptive control techniques, and has been successfully implemented in high-performance small-scale UAVs such as the NASA AirSTAR flight test vehicle [26]. This control architecture has recently also been implemented on manned aircraft including the variable stability Learjet [27] and F-16 Fighting Falcon [28]. The advantage of the $\mathcal{L}_1$ adaptive control is that to the pilot flying the aircraft, the inputs are always predictable and produce the same response, regardless of operating condition. As a result, implementing an $\mathcal{L}_1$ adaptive controller removes the variability of the system resulting in consistent and repeatable performance.

$\mathcal{L}_1$ adaptive control is based on the MRAC architecture with the addition of a low pass filter that decouples robustness from adaptation [21]. This allows arbitrarily high adaptation gains to be chosen for fast adaptation. The development of improved $\mathcal{L}_1$

adaptive control schemes in theory and verified in simulation for a quadrotor show its promising potential [29, 30]. This relatively new algorithm has already shown promising results showing compensation of partial motor failure in [31] in a higher level translational velocity $\mathcal{L}_1$ control loop. Improved tracking performance and successful disturbance rejection of offset center of gravity location has been demonstrated by [32] using an $\mathcal{L}_1$ adaptive attitude controller to augment a baseline controller tested on a quadrotor, hexacopter and octocopter. Compensation of quadrotor mass uncertainty with an $\mathcal{L}_1$ adaptive controller was shown in [33] on a quadrotor in experiment.

As a result of its emerging use in quadrotor control and the successful control performance achieved in experiments, an $\mathcal{L}_1$ adaptive controller seems to be a logical choice for control under uncertainty in mass properties.

2.3 Payload Carrying Methods

While the majority of related research work that has looked into mass property parameter identification has mainly achieved this by adding masses at different locations on the quadrotor frame [9, 34, 12], some interesting designs have been implemented to carry objects with a quadrotor. A simple friction contact mechanism was implemented in [11] and [35, 36] consisting of two arms that would squeeze the object to be carried. Additionally, [11] used an ingressive grasping mechanism using four hooks that imbedded into objects made of soft wood. In [15] and [14], a more complicated mechanism to perturb CG and inertia is achieved with a robotic arm which was mounted to the underside of the quadrotor frame, which has been implemented in experiment by [37].

An attachment mechanism must be implemented in order to demonstrate package delivery in this project. However, the design should be simple and easy to implement in order to analyze the controller performance, which is the focus of this project.

Chapter 3

Contributions

3.1 Novel Contributions of This Work

As a result of its relatively recent development, there is a limited variety of $\mathcal{L}_1$ adaptive control architectures implemented on quadrotors. The aim of previous work was primarily to show the capability of $\mathcal{L}_1$ adaptive control in quadrotors. It has not been implemented in experiments under complete uncertainty in the mass properties. Applications of the $\mathcal{L}_1$ adaptive control framework in quadrotors has also been limited, including the specific objectives of package delivery. The work presented here, compares the tracking performance and disturbance rejection of several different $\mathcal{L}_1$ adaptive control architectures implemented on a quadrotor. Knowledge of the performance characteristics of different architectures will serve as a design guideline for practical $\mathcal{L}_1$ adaptive control design. In addition, a preliminary $\mathcal{L}_1$ adaptive control tuning guideline is presented based on experimental tuning of a quadrotor. Furthermore, the results of experiments combining $\mathcal{L}_1$ adaptive control within an Iterative Learning Control (ILC) framework in quadrotors show it is suitable for integration into frameworks with higher level control objectives. Finally, a cost function to quantify tracking error or control input shows continuity in the $\mathcal{L}_1$ tuning parameters. This confirms the compatibility of $\mathcal{L}_1$ adaptive control with other learning schemes that aim to optimize performance.

3.2 Outline

The remainder of this thesis is broadly divided according to the logical progression of the project objectives as discussed above in Section 1.2. First, a three-dimensional dynamic model is presented to understand how the quadrotor dynamics are affected by uncon-

ventional center-of-mass location and a full second-moment-of-mass matrix. This section also contains an analysis of the maximum payload limitations dependent on quadrotor design constraints. Using this dynamic model, a nonlinear observability analysis of the unknown mass properties is conducted to ensure that typical quadrotor measurements allow for the estimation of these parameters. This is followed by an overview of the existing nonlinear controller used by the Dynamic Systems Lab for the control of an undisturbed Parrot AR.Drone quadrotor UAV. Next, the $\mathcal{L}_1$ adaptive control theory is introduced as well as the different architectures used in experiments, and a general guideline of tuning an $\mathcal{L}_1$ adaptive controller. An analysis of the convergence, transient and steady-state behaviours is also provided for the closed-loop $\mathcal{L}_1$ adaptive output feedback controller implemented on a quadrotor. Finally, the experimental setup, results and discussion of the performance comparison are presented. This section also shows the demonstration of a payload transporting quadrotor meeting the objectives as outlined above in Section 1.2.

Chapter 4

General Quadrotor Dynamics Model

This section derives the full three-dimensional dynamics model of the quadrotor with offset center of gravity in the body-fixed frame $\mathcal{F}_B$ with the reference point O_B chosen to coincide with the geometric center. This frame and reference point rotates and translates with respect to an inertial reference frame $\mathcal{F}_I$ with an inertial reference origin O_I.

For the purposes of observability analysis, the equations of motion are required to be in control-affine form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{f}_i(\mathbf{x}) u_i \ , \tag{4.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ are the control inputs, and $\mathbf{f}_i : \mathbb{R}^n \to \mathbb{R}^n$ for $i = 0, \ldots, m$ are nonlinear vector fields. This is achieved by treating the translational and rotational dynamics together.

4.1 Preliminaries

The notation used in this chapter is identical to [38]. The frames and points of reference are shown in Figure 4.1. The center of gravity is offset from the geometric center by $\vec{r}_{\text{off}}$. The vector between the inertial reference origin point O_I to the quadrotor geometric center, O_B is $\vec{r}_b$. The center of mass location of the quadrotor assumed to be a rigid-body is indicated by $\oplus$ and the position vector from the inertial origin is defined to be $\vec{r}_{\oplus}$.

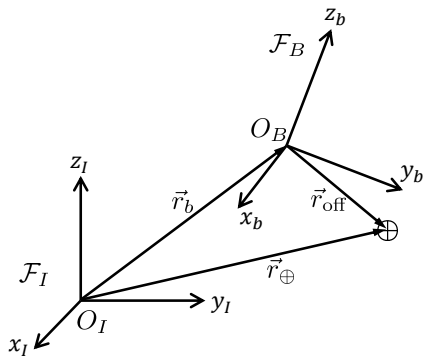


Figure 4.1: Inertial and body-fixed 3D frames and points of reference.

4.1.1 Time Rate of Change of Vectors

Differentiation in time in $\mathcal{F}_I$ and $\mathcal{F}_B$ are defined as follows:

$$(\cdot)^{\cdot} \triangleq \frac{d}{dt}(\cdot)\bigg|_{\mathcal{F}_I} \qquad \text{and} \qquad (\cdot)^{\circ} \triangleq \frac{d}{dt}(\cdot)\bigg|_{\mathcal{F}_B} . \tag{4.2}$$

Let $\vec{v}_{BI}$ be the relative translational velocity of O_B to O_I measured in $\mathcal{F}_I$. In other words, $\vec{v}_{BI} \triangleq \vec{r}_b^{\,\cdot}$. Additionally, let $\vec{\omega}_{BI}$ be the relative angular velocity of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$.

Recall that time differentiation of a vector in one frame of reference is different as expressed in a different frame of reference depending on the relative angular velocity between the two frames of reference [38]. Consider a vector $\vec{r}$, then the time rate of change of $\vec{r}$ in the rotating frame $\mathcal{F}_B$ is:

$$\vec{r}^{\,\cdot} = \vec{r}^{\,\circ} + \vec{\omega}_{BI} \times \vec{r} \, . \tag{4.3}$$

4.1.2 Referential Form and Rotation of Vectors

Vectors can be expressed in referential form by making use of the Hughes vectrix, $\vec{\mathcal{F}}$ which is a 3×1 column matrix of coordinate vectors defining the respective reference frame [39]. The vector $\vec{r}$ can be expressed in both frame $\mathcal{F}_I$ and $\mathcal{F}_B$ as follows:

$$\vec{r} = \vec{\mathcal{F}}_I^{\,T} \mathbf{r}_I = \vec{\mathcal{F}}_B^{\,T} \mathbf{r}_B \, . \tag{4.4}$$

The rotation from $\mathcal{F}_I$ to $\mathcal{F}_B$ is given by the 3×3 rotation matrix $\mathbf{C}_{BI}$:

$$\vec{\mathcal{F}}_I^T = \vec{\mathcal{F}}_B^T \mathbf{C}_{BI} \qquad \text{such that (4.4) can be expressed in } \mathcal{F}_B\text{:} \qquad \mathbf{C}_{BI}\mathbf{r}_I = \mathbf{r}_B \ . \tag{4.5}$$

Rotation matrices are by definition orthogonal such that $\mathbf{C}_{IB}^{-1} = \mathbf{C}_{IB}^T = \mathbf{C}_{BI}$. Rotations between reference frames are chosen to be represented by Euler angles. Although there always exists a singularity with a 3-dimensional representation of a rotation matrix, only three parameters are associated with the rotation instead of four. The 3-2-1 (Z-Y-X) Euler rotation sequence from the body frame to the inertial frame is defined by the following sequence of principal rotations:

$$\mathbf{C}_{IB}(\phi,\theta,\psi) = \mathbf{C}_x(\phi)\mathbf{C}_y(\theta)\mathbf{C}_z(\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \ , \tag{4.6}$$

where ϕ, θ, ψ are the roll, pitch and yaw angles, respectively, and $c_b = \cos(b)$ and $s_b = \sin(b)$.

A 3-2-1 (Z-Y-X) Euler rotation sequence is chosen since the singularity occurs at a pitch angle of $\theta = \pm 90^\circ$ [38], which is not expected to occur during normal, non-agressive operation.

4.1.3 Euler Angle Rates

The relationship between the attitude described by the Euler angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$ and angular velocity $\boldsymbol{\omega}_{BI_B}$ of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$ expressed in $\mathcal{F}_B$ is given by:

$$\boldsymbol{\omega}_{BI_B} = \mathbf{S}(\phi,\theta)\dot{\boldsymbol{\theta}} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \ , \tag{4.7}$$

where the angular velocity is expressed in terms of Euler angle rates [2, 7]. Derivation of this equation is provided in [38] and [39]. The inverse of $\mathbf{S}(\phi,\theta)$ is:

$$\mathbf{S}^{-1}(\phi,\theta) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \ , \tag{4.8}$$

which is singular at the pitch angle singularities, but invertible everywhere else.

4.1.4 Cross Product

The cross operator for any matrix $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T \in \mathbb{R}^{3\times 1}$ is defined as follows:

$$\mathbf{a}^\times \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} . \text{ Note that } \mathbf{a}^\times \text{ is antisymmetric: } {\mathbf{a}^\times}^T = -\mathbf{a}^\times. \tag{4.9}$$

From [39, 38], the cross product between two vectors can be expressed in reference frame $\mathcal{F}_B$ as follows:

$$\vec{a} \times \vec{b} = \vec{a} \times \vec{\mathcal{F}}_B^T \mathbf{b}_B = \vec{\mathcal{F}}_B^T \, \mathbf{a}_B^\times \mathbf{b}_B \, . \tag{4.10}$$

4.1.5 Referential Form of Twain Vectors and the Cross Operator

Twain vectors and their respective referential form will become useful to compactly express the translational and rotational dynamics. A detailed discussion of twain vector algebra is provided in [39] *A Chain of Bodies*. For the purposes of deriving the dynamic model, it is sufficient to only discuss the referential form of a dextor and its cross operator.

Consider the referential form of dextor $\underset{\rightarrow}{\boldsymbol{a}}$ associated with a reference frame and reference point attached in this frame:

$$\boldsymbol{a} \triangleq \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{bmatrix} \qquad \text{where } \mathbf{a} \text{ is defined above and } \boldsymbol{\alpha} \triangleq \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix}^T \in \mathbb{R}^{3\times 1} \, . \tag{4.11}$$

The twain cross operator $\boldsymbol{a}^\times$ of a twain vector in referential form $\boldsymbol{a}$ is defined as:

$$\boldsymbol{a}^\times \triangleq \begin{bmatrix} \boldsymbol{\alpha}^\times & \mathbf{a}^\times \\ \mathbf{0}_{3\times 3} & \boldsymbol{\alpha}^\times \end{bmatrix} \, , \tag{4.12}$$

where the cross operator on the 3×1 matrices $\mathbf{a}$ and $\boldsymbol{\alpha}$ are defined in (4.10).

4.2 Quadrotor Dynamic Model

4.2.1 Quadrotor Kinematics

The quadrotor position in space is given by the vector $\vec{r}_b$ from the inertial origin to the quadrotor geometric center. The translational velocity of the quadrotor is defined by the vector $\vec{v}_{BI} \triangleq \vec{r}_b^{\,\cdot}$ as mentioned above in Section 4.1.1. This establishes the following relationship in referential form:

$$\begin{aligned} \vec{\mathcal{F}}_I^T \dot{\mathbf{r}}_{b_I} &= \vec{\mathcal{F}}_B^T \mathbf{v}_{BI_B} = \vec{\mathcal{F}}_I^T \mathbf{C}_{IB} \mathbf{v}_{BI_B} \\ \dot{\mathbf{r}}_{b_B} &= \mathbf{C}_{IB} \mathbf{v}_{BI_B} \ . \end{aligned} \tag{4.13}$$

Similarly, for the Euler angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$ in (4.7):

$$\dot{\boldsymbol{\theta}} = \mathbf{S}^{-1}(\phi, \theta) \boldsymbol{\omega}_{BI_B} \ . \tag{4.14}$$

4.2.2 Linear and Angular Momentum

The linear momentum $\vec{p}$ and angular momentum $\vec{h}$ for an arbitrary reference point O_B within a rigid body is derived by treating the rigid body as a collection of infinitesimal mass elements dm. The expressions derived here for the quadrotor are based on the extension of the derivation for a system of particles in [39]. Let $\vec{s}$ be a vector to an arbitrary mass element within the rigid body from the reference point O_B. Since the body is rigid, $\vec{s}^{\,\circ} = \vec{0}$, and:

$$\vec{s}^{\,\cdot} = \vec{\omega}_{BI} \times \vec{s} \ , \tag{4.15}$$

where $\vec{\omega}_{BI}$ is the angular velocity between the body-fixed and inertial reference frames. Also let the translational velocity of the reference point O_B with respect to an inertial reference point O_I be $\vec{v}_{BI} = \vec{r}_b^{\,\cdot}$ where $\vec{r}_b$ is the vector from O_I to O_B. Finally, let $\vec{r}$ be a vector from the inertial origin O_I to an arbitrary mass element, such that $\vec{r} = \vec{r}_b + \vec{s}$. This means that:

$$\vec{r}^{\,\cdot} = \vec{v}_{BI} + \vec{s}^{\,\cdot} \tag{4.16}$$

$$= \vec{v}_{BI} + \vec{\omega}_{BI} \times \vec{s} \ . \tag{4.17}$$

The linear momentum of the rigid-body is:

$$\begin{aligned}\vec{p} &\triangleq \int_B \vec{r}\,\dot{}\,dm \qquad (4.18)\\ &= \int_B \vec{v}_{BI} + \vec{\omega}_{BI} \times \vec{s}\; dm \qquad \text{after substitution of (4.17),}\\ &= \int_B dm\; \vec{v}_{BI} - \left(\int_B \vec{s}\; dm \right) \times \vec{\omega}_{BI}\\ &= m\vec{v}_{BI} - \vec{c} \times \vec{\omega}_{BI}\ , \qquad (4.19)\end{aligned}$$

where $m \triangleq \int_B dm$ and $\vec{c} \triangleq \int_B \vec{s}\; dm$ are the definitions of the zeroth and first moments of mass about O_B respectively [39]. In the case of the quadrotor, the first moment of mass is $\vec{c} = m\vec{r}_{\text{off}}$, where $\vec{r}_{\text{off}}$ is defined in Section 4.1. This means that the linear momentum is:

$$\vec{p} = m\vec{v}_{BI} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI}\ . \qquad (4.20)$$

The angular momentum of the rigid body about the body-fixed reference point O_B is:

$$\begin{aligned}\vec{h} &\triangleq \int_B \vec{s} \times d\vec{p} \qquad (4.21)\\ &= \int_B \vec{s} \times \vec{r}\,\dot{}\,dm\\ &= \int_B \vec{s} \times \vec{v}_{BI}\; dm + \int_B \vec{s} \times (\vec{\omega}_{BI} \times \vec{s})\; dm \qquad \text{after substitution of (4.17),}\\ &= \int_B \vec{s}\; dm \times \vec{v}_{BI} - \int_B \vec{s} \times (\vec{s} \times \vec{\omega}_{BI})\; dm\ .\end{aligned}$$

Substituting in the definition of the first moment of mass, and making use of the vector triple product identity [39] $\big(\ (\vec{a} \times \vec{b}) \times \vec{c} = -\vec{c} \times (\vec{a} \times \vec{b}) = (\vec{c} \cdot \vec{a})\vec{b} - (\vec{c} \cdot \vec{b})\vec{a}\ \big)$:

$$\begin{aligned}&= \vec{c} \times \vec{v}_{BI} - \int_B (\vec{s} \cdot \vec{s})\vec{\omega}_{BI} - (\vec{s} \cdot \vec{\omega}_{BI})\vec{s}\; dm\\ &= \vec{c} \times \vec{v}_{BI} + \int_B [(\vec{s} \cdot \vec{s})\underline{\underline{1}} - \vec{s}\vec{s}] \cdot \vec{\omega}_{BI}\; dm\ ,\end{aligned}$$

where $\underset{\rightarrow}{1}$ is the unit dyadic, and $\vec{s}\vec{s}$ indicates the dyadic product.

$$\begin{aligned} &= \vec{c} \times \vec{v}_{BI} + \left(\int_B (\vec{s} \cdot \vec{s}) \underset{\rightarrow}{1} - \vec{s}\vec{s} \, dm \right) \cdot \vec{\omega}_{BI} \\ &= \vec{c} \times \vec{v}_{BI} + \underset{\rightarrow}{J} \cdot \vec{\omega}_{BI} \ , \end{aligned} \tag{4.22}$$

where $\underset{\rightarrow}{J} \triangleq \int_B (\vec{s} \cdot \vec{s}) \underset{\rightarrow}{1} - \vec{s}\vec{s} \, dm$ is the definition of the second moment of mass or *inertia tensor* about O_B (a second-order tensor) [39]. In the quadrotor case, no further restricting assumptions are made on the inertia tensor due to the center of gravity offset. As in the linear momentum above, $\vec{c} = m\vec{r}_{\text{off}}$ meaning that the angular momentum about O_B is:

$$\vec{h} = m\vec{r}_{\text{off}} \times \vec{v}_{BI} + \underset{\rightarrow}{J} \cdot \vec{\omega}_{BI} \ . \tag{4.23}$$

The translational and rotational equations of motion of the quadrotor are derived next. Note that the moments of mass m, $\vec{c}$ and $\underset{\rightarrow}{J}$ are constants in the body-fixed frame, and that the first and second moments are dependent on the choice of reference point O_B. As discussed in the previous section, O_B is chosen to be coincident with the geometric center of the quadrotor. However, since different payloads or packages will be carried by the quadrotor, all three moments of mass will change with a different quadrotor-payload configuration.

4.2.3 Translational Equations of Motion

The translational dynamics of a rigid body in three dimensions are governed by Newton's law of motion:

$$\vec{p}^{\cdot} = \vec{f}_{\text{ext}} \tag{4.24}$$

where $\vec{p}$ is the linear momentum, and $\vec{f}_{\text{ext}}$ is the externally applied force to the rigid body. Making use of (4.3) the time rate of change of $\vec{p}$ in the body frame is:

$$\begin{aligned} \vec{p}^{\cdot} &= \vec{p}^{\circ} + \vec{\omega}_{BI} \times \vec{p} \\ &= \vec{p}^{\circ} - \vec{p} \times \vec{\omega}_{BI} \ . \end{aligned}$$

Substitution of (4.20) and recalling that $m^{\circ} = 0$ and $\vec{r}_{\text{off}}^{\,\circ} = \vec{0}$ results in:

$$\begin{aligned}\vec{p}^{\,\cdot} &= (m\vec{v}_{BI} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI})^{\circ} - (m\vec{v}_{BI} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI}) \times \vec{\omega}_{BI} \\ &= m\vec{v}_{BI}^{\,\circ} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI}^{\,\circ} + m\vec{\omega}_{BI} \times \vec{v}_{BI} - m\vec{\omega}_{BI} \times (\vec{r}_{\text{off}} \times \vec{\omega}_{BI}) \ .\end{aligned}$$

Therefore, the translational dynamics are described by the following equation:

$$m\vec{v}_{BI}^{\,\circ} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI}^{\,\circ} + m\vec{\omega}_{BI} \times \vec{v}_{BI} - m\vec{\omega}_{BI} \times (\vec{r}_{\text{off}} \times \vec{\omega}_{BI}) = \vec{f}_{\text{ext}} \ . \tag{4.25}$$

Alternatively, (4.25) can be expressed in referential form in the body-fixed frame:

$$\begin{aligned}m\vec{\mathcal{F}}_B^{\,T}\dot{\mathbf{v}}_{BI_B} - m\vec{\mathcal{F}}_B^{\,T}\mathbf{r}_{\text{off}_B}^{\times}\dot{\boldsymbol{\omega}}_{BI_B} + m\vec{\mathcal{F}}_B^{\,T}\boldsymbol{\omega}_{BI_B}^{\times}\mathbf{v}_{BI_B} - m\vec{\omega}_{BI} \times \vec{\mathcal{F}}_B^{\,T}\mathbf{r}_{\text{off}_B}^{\times}\boldsymbol{\omega}_{BI_B} &= \vec{\mathcal{F}}_B^{\,T}\mathbf{f}_{\text{ext}_B} \\ \vec{\mathcal{F}}_B^{\,T}\Big(m\dot{\mathbf{v}}_{BI_B} - m\mathbf{r}_{\text{off}_B}^{\times}\dot{\boldsymbol{\omega}}_{BI_B} + m\boldsymbol{\omega}_{BI_B}^{\times}\mathbf{v}_{BI_B} - m\boldsymbol{\omega}_{BI_B}^{\times}\mathbf{r}_{\text{off}_B}^{\times}\boldsymbol{\omega}_{BI_B}\Big) &= \vec{\mathcal{F}}_B^{\,T}\mathbf{f}_{\text{ext}_B} \\ m\dot{\mathbf{v}}_{BI_B} - m\mathbf{r}_{\text{off}_B}^{\times}\dot{\boldsymbol{\omega}}_{BI_B} + m\boldsymbol{\omega}_{BI_B}^{\times}\mathbf{v}_{BI_B} - m\boldsymbol{\omega}_{BI_B}^{\times}\mathbf{r}_{\text{off}_B}^{\times}\boldsymbol{\omega}_{BI_B} &= \mathbf{f}_{\text{ext}_B} \ .\end{aligned} \tag{4.26}$$

4.2.4 Rotational Equations of Motion

Consider the rigid body from the previous section. The rotational dynamics are derived by differentiating the definition of the angular momentum of a rigid body (4.21) about the body-fixed reference point O_B in the inertial reference frame $\mathcal{F}_I$:

$$\begin{aligned}\vec{h}^{\,\cdot} &= \left(\int_B \vec{s} \times d\vec{p}\right)^{\cdot} \\ &= \int_B \vec{s}^{\,\cdot} \times d\vec{p} + \int_B \vec{s} \times d\vec{p}^{\,\cdot} \ .\end{aligned} \tag{4.27}$$

From (4.16) and (4.24):

$$\begin{aligned}&= \int_B (\vec{r}^{\,\cdot} - \vec{v}_{BI}) \times d\vec{p} + \int_B \vec{s} \times d\vec{f} \\ &= \int_B \vec{r}^{\,\cdot} \times \vec{r}^{\,\cdot} dm - \vec{v}_{BI} \times \int_B \vec{r}^{\,\cdot} dm + \int_B \vec{s} \times d\vec{f} \ .\end{aligned}$$

Since $\vec{r}^{\,\cdot} \times \vec{r}^{\,\cdot} = \vec{0}$, $\vec{\tau}_{\text{ext}} \triangleq \int_B \vec{s} \times d\vec{f}$ is the total external torque about the reference point O_B, and $\vec{p} \triangleq \int_B \vec{r}^{\,\cdot} dm$ from definition (4.18):

$$\begin{aligned}
&\vec{h}^{\,\cdot} = -\vec{v}_{BI} \times \vec{p} + \vec{\tau}_{\text{ext}} \\
&\vec{h}^{\,\cdot} + \vec{v}_{BI} \times \vec{p} = \vec{\tau}_{\text{ext}} \text{ , which results in:} \\
&\vec{h}^{\,\circ} + \vec{\omega}_{BI} \times \vec{h} + \vec{v}_{BI} \times \vec{p} = \vec{\tau}_{\text{ext}} \ ,
\end{aligned} \tag{4.28}$$

where the derivative is taken in the body frame $\mathcal{F}_B$. An expression for the third term $\vec{v}_{BI} \times \vec{p}$ can be obtained from (4.20):

$$\begin{aligned}
\vec{v}_{BI} \times \vec{p} &= \vec{v}_{BI} \times (m\vec{v}_{BI} - m\vec{r}_{\text{off}} \times \vec{\omega}_{BI}) \\
&= m\vec{v}_{BI} \times \vec{v}_{BI} - m\vec{v}_{BI} \times (\vec{r}_{\text{off}} \times \vec{\omega}_{BI}) \\
&= m\vec{v}_{BI} \times (\vec{\omega}_{BI} \times \vec{r}_{\text{off}}) \ .
\end{aligned} \tag{4.29}$$

The rotational dynamics are obtained by substituting (4.23) and (4.29) into (4.28):

$$\begin{aligned}
&(m\vec{r}_{\text{off}} \times \vec{v}_{BI} + \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI})^{\circ} + \vec{\omega}_{BI} \times (m\vec{r}_{\text{off}} \times \vec{v}_{BI} + \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI}) + (m\vec{v}_{BI} \times (\vec{\omega}_{BI} \times \vec{r}_{\text{off}})) = \vec{\tau}_{\text{ext}} \\
&m\vec{r}_{\text{off}} \times \vec{v}_{BI}^{\,\circ} + \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI}^{\,\circ} + m\big(\vec{\omega}_{BI} \times (\vec{r}_{\text{off}} \times \vec{v}_{BI}) + \vec{v}_{BI} \times (\vec{\omega}_{BI} \times \vec{r}_{\text{off}})\big) + \vec{\omega}_{BI} \times \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI} = \vec{\tau}_{\text{ext}} \ .
\end{aligned}$$

By using the Jacobi identity $\big(\ \vec{a} \times (\vec{b} \times \vec{c}) + \vec{b} \times (\vec{c} \times \vec{a}) + \vec{c} \times (\vec{a} \times \vec{b}) = \vec{0} \ \big)$:

$$m\vec{r}_{\text{off}} \times \vec{v}_{BI}^{\,\circ} + \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI}^{\,\circ} - m\vec{r}_{\text{off}} \times (\vec{v}_{BI} \times \vec{\omega}_{BI}) + \vec{\omega}_{BI} \times \underset{\rightharpoonup}{J} \cdot \vec{\omega}_{BI} = \vec{\tau}_{\text{ext}} \ . \tag{4.30}$$

As in the linear momentum case, (4.30) can be expressed in referential form in the body-fixed frame:

$$\begin{aligned}
&\vec{\mathcal{F}}_B^{\,T}\Big(m\mathbf{r}_{\text{off}_B}^{\times}\dot{\mathbf{v}}_{BI_B} + \mathbf{J}\dot{\boldsymbol{\omega}}_{BI_B} - m\mathbf{r}_{\text{off}_B}^{\times}\mathbf{v}_{BI_B}^{\times}\boldsymbol{\omega}_{BI_B} + \boldsymbol{\omega}_{BI_B}^{\times}\mathbf{J}\boldsymbol{\omega}_{BI_B}\Big) = \vec{\mathcal{F}}_B^{\,T}\boldsymbol{\tau}_{\text{ext}_B} \\
&m\mathbf{r}_{\text{off}_B}^{\times}\dot{\mathbf{v}}_{BI_B} + \mathbf{J}\dot{\boldsymbol{\omega}}_{BI_B} - m\mathbf{r}_{\text{off}_B}^{\times}\mathbf{v}_{BI_B}^{\times}\boldsymbol{\omega}_{BI_B} + \boldsymbol{\omega}_{BI_B}^{\times}\mathbf{J}\boldsymbol{\omega}_{BI_B} = \boldsymbol{\tau}_{\text{ext}_B} \ .
\end{aligned} \tag{4.31}$$

4.2.5 Combined Translational and Rotational Dynamics

Both (4.26) and (4.31) describe the dynamics of the quadrotor. To arrive at the control-affine form in (4.1), the translational rotational equations of motion expressed in the body-fixed frame can be stacked to simplify the nonlinear observability analysis. Define

the following matrices:

$$\boldsymbol{v} \triangleq \begin{bmatrix} \mathbf{v}_{BI_B} \\ \boldsymbol{\omega}_{BI_B} \end{bmatrix} , \qquad \boldsymbol{M} \triangleq \begin{bmatrix} m\mathbf{1}_{3\times 3} & -m\mathbf{r}^{\times}_{\mathrm{off}_B} \\ m\mathbf{r}^{\times}_{\mathrm{off}_B} & \mathbf{J} \end{bmatrix} , \qquad \boldsymbol{f} \triangleq \begin{bmatrix} \mathbf{f}_{\mathrm{ext}_B} \\ \boldsymbol{\tau}_{\mathrm{ext}_B} \end{bmatrix} , \tag{4.32}$$

where $\mathbf{1}_{n\times n}$ is the n-by-n identity matrix. Note that $\boldsymbol{M} = \boldsymbol{M}^T$ is symmetric, $\boldsymbol{v}$ is the referential form of a dextor and $\boldsymbol{f}$ is the referential form of a pseudodextor. Since $m > 0$, and the inertia matrix $\mathbf{J}$ is of a rigid-body occupying three-dimensional space, $\mathbf{J} > 0$ is positive-definite, therefore $\det(\boldsymbol{M}) \neq 0$ and $\boldsymbol{M}$ is invertible.

Combining (4.26) and (4.31) and making use of the Jacobi identity for matrices in referential form:

$$\begin{bmatrix} m\dot{\mathbf{v}}_{BI_B} - m\mathbf{r}^{\times}_{\mathrm{off}_B}\dot{\boldsymbol{\omega}}_{BI_B} + m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{v}_{BI_B} - m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}_B}\boldsymbol{\omega}_{BI_B} \\ m\mathbf{r}^{\times}_{\mathrm{off}_B}\dot{\mathbf{v}}_{BI_B} + \mathbf{J}\dot{\boldsymbol{\omega}}_{BI_B} - m\mathbf{r}^{\times}_{\mathrm{off}_B}\mathbf{v}^{\times}_{BI_B}\boldsymbol{\omega}_{BI_B} + \boldsymbol{\omega}^{\times}_{BI_B}\mathbf{J}\boldsymbol{\omega}_{BI_B} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{ext}_B} \\ \boldsymbol{\tau}_{\mathrm{ext}_B} \end{bmatrix}$$

$$\begin{bmatrix} m\mathbf{1}_{3\times 3} & -m\mathbf{r}^{\times}_{\mathrm{off}_B} \\ m\mathbf{r}^{\times}_{\mathrm{off}_B} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{BI_B} \\ \dot{\boldsymbol{\omega}}_{BI_B} \end{bmatrix} + \begin{bmatrix} m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{v}_{BI_B} - m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}_B}\boldsymbol{\omega}_{BI_B} \\ m(\mathbf{v}^{\times}_{BI_B}\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}_{\mathrm{off}_B} + \boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}_B}\mathbf{v}_{BI_B}) + \boldsymbol{\omega}^{\times}_{BI_B}\mathbf{J}\boldsymbol{\omega}_{BI_B} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathrm{ext}_B} \\ \boldsymbol{\tau}_{\mathrm{ext}_B} \end{bmatrix}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \begin{bmatrix} m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{v}_{BI_B} - m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}_B}\boldsymbol{\omega}_{BI_B} \\ -m\mathbf{v}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}}\boldsymbol{\omega}_{BI_B} + m\boldsymbol{\omega}^{\times}_{BI_B}\mathbf{r}^{\times}_{\mathrm{off}_B}\mathbf{v}_{BI_B} + \boldsymbol{\omega}^{\times}_{BI_B}\mathbf{J}\boldsymbol{\omega}_{BI_B} \end{bmatrix} = \boldsymbol{f}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \begin{bmatrix} \boldsymbol{\omega}^{\times}_{BI_B} & \mathbf{0}_{3\times 3} \\ \mathbf{v}^{\times}_{BI_B} & \boldsymbol{\omega}^{\times}_{BI_B} \end{bmatrix} \begin{bmatrix} m\mathbf{v}_{BI_B} - m\mathbf{r}^{\times}_{\mathrm{off}_B}\boldsymbol{\omega}_{BI_B} \\ -m\mathbf{r}^{\times}_{\mathrm{off}_B}\mathbf{v}_{BI_B} + \mathbf{J}\boldsymbol{\omega}_{BI_B} \end{bmatrix} = \boldsymbol{f}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} - \begin{bmatrix} -\boldsymbol{\omega}^{\times}_{BI_B} & \mathbf{0}_{3\times 3} \\ -\mathbf{v}^{\times}_{BI_B} & -\boldsymbol{\omega}^{\times}_{BI_B} \end{bmatrix} \begin{bmatrix} m\mathbf{1}_{3\times 3} & -m\mathbf{r}^{\times}_{\mathrm{off}_B} \\ m\mathbf{r}^{\times}_{\mathrm{off}_B} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{BI_B} \\ \boldsymbol{\omega}_{BI_B} \end{bmatrix} = \boldsymbol{f}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} - \boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} = \boldsymbol{f} \ . \tag{4.33}$$

This expression describes the combined translational and rotational dynamics of the quadrotor in the body-fixed frame $\mathcal{F}_B$ about the geometric center O_B of the quadrotor frame for the general case, i.e. without the simplifying assumptions on center of gravity location and body-frame axes alignment. Note that with the simplifying assumptions of a center of gravity location coincident with the geometric center, and $\mathcal{F}_B$ aligned with the principal axis frame, the first moment of mass about O_B becomes zero ($\mathbf{r}_{\mathrm{off}_B} = \mathbf{0}$), and the second moment of mass about O_B reduces to a diagonal matrix. The resulting mass matrix $\boldsymbol{M}$ also reduces to a diagonal matrix recovering the typical dynamic quadrotor

model in [2].

4.2.6 External Forces and Torques

As defined in (4.32), the external forces and torques applied to the quadrotor are combined into $\boldsymbol{f}$. To obtain the control-affine form in (4.1), it is beneficial to decompose $\boldsymbol{f}$ into the forces and torques from the controller, and those applied to the quadrotor by Earth's gravity separately:

$$\boldsymbol{f} = \boldsymbol{f}_g + \boldsymbol{f}_c = \begin{bmatrix} \mathbf{F}_{g_B} \\ \boldsymbol{\tau}_{g_B} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_{c_B} \\ \boldsymbol{\tau}_{c_B} \end{bmatrix} \ .$$

The external forces and torques arising from gravity acting on the quadrotor geometric center $\boldsymbol{f}_g$ can be obtained by assuming gravity acts as a uniform vector field. This assumption means that gravity acts as a force on the quadrotor center of gravity in the inertial reference frame, which can be expressed in the body frame as follows:

$$\vec{f}_g = \vec{\mathcal{F}}_I^T \mathbf{F}_{g_I} = \vec{\mathcal{F}}_I^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \vec{\mathcal{F}}_B^T \mathbf{C}_{BI} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \vec{\mathcal{F}}_B^T \mathbf{F}_{g_B} \ , \tag{4.34}$$

where the first relation in (4.5) has been used to rotate the force of gravity in the body-fixed frame. The external torque about O_B is simply the moment of $\vec{f}_g$ acting $\vec{r}_{\text{off}}$ from O_B:

$$\vec{\tau}_g = \vec{r}_{\text{off}} \times \vec{f}_g = \vec{r}_{\text{off}} \times \vec{\mathcal{F}}_B^T \mathbf{F}_{g_B} = \vec{\mathcal{F}}_B^T \mathbf{r}_{\text{off}_B}^{\times} \mathbf{F}_{g_B} = \vec{\mathcal{F}}_B^T \boldsymbol{\tau}_{g_B} \ . \tag{4.35}$$

The forces and torques from the motors acting on the quadrotor about the geometric center are independent of the assumptions on center of gravity location and inertia, so the typical quadrotor control forces and torques are derived here. The motor locations with respect to the body-fixed reference frame and geometric center are shown in Figure 2.1. The quadrotor has a motor driving a propeller on each end of its four arms with length l from the quadrotor geometric center. These control the quadrotor by applying forces and moments to the quadrotor. In the body-fixed frame, the control inputs are defined

as:

$$\mathbf{F}_{c_B} = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\tau}_{c_B} = \begin{bmatrix} l(F_2 - F_4) \\ l(-F_1 + F_3) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} . \tag{4.36}$$

The thrust, F_i, and moment, M_i, generated by each propeller are modelled to be proportional to the square of the rotational speed ω_i of the propeller blade [2], such that $F_i = k_f \omega_i^2$ and $M_i = k_m \omega_i^2$. This results in the following relationship between the thrust and moment: $M_i = \gamma F_i$, where $\gamma = \frac{k_m}{k_f}$.

Let the control inputs, be defined as the individual motor forces $u_i \triangleq F_i$, then the control inputs can be expressed as follows:

$$\boldsymbol{f}_c = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \\ l(F_2 - F_4) \\ l(-F_1 + F_3) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -l \\ \gamma \end{bmatrix} F_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ l \\ 0 \\ -\gamma \end{bmatrix} F_2 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ l \\ \gamma \end{bmatrix} F_3 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -l \\ 0 \\ -\gamma \end{bmatrix} F_4 .$$

Define the following 6×1 matrices $\mathbf{g}_i$:

$$\mathbf{g}_1 \triangleq \begin{bmatrix} 0 & 0 & 1 & 0 & -l & \gamma \end{bmatrix}^T \qquad \mathbf{g}_2 \triangleq \begin{bmatrix} 0 & 0 & 1 & l & 0 & -\gamma \end{bmatrix}^T$$
$$\mathbf{g}_3 \triangleq \begin{bmatrix} 0 & 0 & 1 & 0 & l & \gamma \end{bmatrix}^T \qquad \mathbf{g}_4 \triangleq \begin{bmatrix} 0 & 0 & 1 & -l & 0 & -\gamma \end{bmatrix}^T$$

This means that the all the external forces and torques acting on the quadrotor can be summarized as follows:

$$\boldsymbol{f} = \boldsymbol{f}_g + \mathbf{g}_1 u_1 + \mathbf{g}_2 u_2 + \mathbf{g}_3 u_3 + \mathbf{g}_4 u_4 \qquad \text{where} \qquad \boldsymbol{f}_g = \begin{bmatrix} \mathbf{F}_{g_B} \\ \mathbf{r}^{\times}_{\text{off}_B} \mathbf{F}_{g_B} \end{bmatrix} . \tag{4.37}$$

No distortion in the control inputs arise as discussed in [8] where the reference point was changed to the offset CG location, the reference point O_B for the moments is kept at the geometric center of the quadrotor. Substitution of (4.37) into (4.33) results in the

following control-affine form:

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{v}} - \boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} &= \boldsymbol{f}_g + \mathbf{g}_1 u_1 + \mathbf{g}_2 u_2 + \mathbf{g}_3 u_3 + \mathbf{g}_4 u_4 \\ \boldsymbol{M}\dot{\boldsymbol{v}} &= \boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} + \boldsymbol{f}_g + \mathbf{g}_1 u_1 + \mathbf{g}_2 u_2 + \mathbf{g}_3 u_3 + \mathbf{g}_4 u_4 \\ \dot{\boldsymbol{v}} &= \boldsymbol{M}^{-1}(\boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} + \boldsymbol{f}_g) + \boldsymbol{M}^{-1}(\mathbf{g}_1 u_1 + \mathbf{g}_2 u_2 + \mathbf{g}_3 u_3 + \mathbf{g}_4 u_4)\ . \end{aligned} \quad (4.38)$$

4.2.7 Quadrotor Nonlinear State Equation in Control Affine Form

Define the state $\mathbf{x}$ to be a 12-dimensional column matrix constructed from the position $\mathbf{r}_{b_I} = (x_I, y_I, z_I)$ expressed in the inertial frame, the attitude described by the Euler angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$, translational velocity $\mathbf{v}_{BI_B} = (\dot{x}_B, \dot{y}_B, \dot{z}_B)$ expressed in the body frame, and the relative angular velocity of the body frame with respect to the inertial frame expressed in the body frame $\boldsymbol{\omega}_{BI_B} = (p, q, r)$ [2, 7]:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{b_I}^T & \boldsymbol{\theta}^T & \mathbf{v}_{BI_B}^T & \boldsymbol{\omega}_{BI_B}^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{r}_{b_I}^T & \boldsymbol{\theta}^T & \boldsymbol{v}^T \end{bmatrix}^T\ . \quad (4.39)$$

The control-affine state equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ in (4.1) can be obtained by combining the quadrotor kinematics and dynamics in (4.13), (4.14) and (4.38) respectively as follows:

$$\begin{aligned} \dot{\mathbf{x}} = &\begin{bmatrix} \mathbf{C}_{IB}(\boldsymbol{\theta})\mathbf{v}_{BI_B} \\ \mathbf{S}^{-1}(\boldsymbol{\theta})\boldsymbol{\omega}_{BI_B} \\ \boldsymbol{M}^{-1}(\boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} + \boldsymbol{f}_g) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{M}^{-1}\mathbf{g}_1 \end{bmatrix} u_1 + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{M}^{-1}\mathbf{g}_2 \end{bmatrix} u_2 \\ &+ \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{M}^{-1}\mathbf{g}_3 \end{bmatrix} u_3 + \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{M}^{-1}\mathbf{g}_4 \end{bmatrix} u_4\ , \end{aligned} \quad (4.40)$$

which is in control-affine form with four inputs and a 12-dimensional state $\mathbf{x}$, where $\mathbf{f}_0(\mathbf{x})$ and $\mathbf{f}_i(\mathbf{x})$ are defined as follows:

$$\mathbf{f}_0(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{C}_{IB}(\boldsymbol{\theta})\mathbf{v}_{BI_B} \\ \mathbf{S}^{-1}(\boldsymbol{\theta})\boldsymbol{\omega}_{BI_B} \\ \boldsymbol{M}^{-1}(\boldsymbol{v}^{\times^T}\boldsymbol{M}\boldsymbol{v} + \boldsymbol{f}_g) \end{bmatrix}\ , \quad \text{and} \quad \mathbf{f}_i(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{M}^{-1}\mathbf{g}_i \end{bmatrix} \quad \text{for } i = 1, \ldots, 4. \quad (4.41)$$

4.3 Analysis of the Dynamics Model

4.3.1 Equilibria of the Equations of Motion

To gain more insight into the equilibria of the dynamics model, consider the following equivalent form of (4.40):

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{C}_{IB}(\boldsymbol{\theta})\mathbf{v}_{BI_B} \\ \mathbf{S}^{-1}(\boldsymbol{\theta})\boldsymbol{\omega}_{BI_B} \\ \boldsymbol{M}^{-1}(\boldsymbol{v}^{\times T}\boldsymbol{M}\boldsymbol{v} + \boldsymbol{f}_g) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \\ \boldsymbol{M}^{-1}\boldsymbol{f}_c \end{bmatrix}, \quad \text{where} \quad \boldsymbol{f}_c = \begin{bmatrix} \mathbf{F}_{c_B} \\ \boldsymbol{\tau}_{c_B} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{2\times 1} \\ u_F \end{bmatrix} \\ \mathbf{u}_\tau \end{bmatrix}, \tag{4.42}$$

where $u_F \in \mathbb{R}$ is the sum of the vertical thrust applied by the motors in the body frame, and $\mathbf{u}_\tau \in \mathbb{R}^{3\times 1}$ is the control torque applied by the four motors about the quadrotor geometric center, as both defined in (4.36).

Definition. $(\overline{\mathbf{x}}, \overline{\mathbf{u}})$ is an equilibrium pair if $\dot{\mathbf{x}} = \mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) = \mathbf{0}$.

Solving (4.42) for $\mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) = \mathbf{0}$, start with the first equation:

$$\mathbf{C}_{IB}(\boldsymbol{\theta})\mathbf{v}_{BI_B} = \mathbf{0}_{3\times 1} \tag{4.43}$$

implies that $\mathbf{v}_{BI_B} = \mathbf{0}$ since $\mathbf{C}_{IB}{}^{-1} = \mathbf{C}_{IB}^T$ is invertible, and that the equilibrium position $\mathbf{r}_{b_I} = \begin{bmatrix} \bar{x}_I & \bar{y}_I & \bar{z}_I \end{bmatrix}$ is arbitrary. Also:

$$\mathbf{S}^{-1}(\boldsymbol{\theta})\boldsymbol{\omega}_{BI_B} = \mathbf{0}_{3\times 1}\ , \tag{4.44}$$

means that $\boldsymbol{\omega}_{BI_B} = \mathbf{0}$ if $\theta \neq \pm\frac{\pi}{2}$, since $(\mathbf{S}^{-1}(\boldsymbol{\theta}))^{-1} = \mathbf{S}(\boldsymbol{\theta})$ is nonsingular. It further follows from (4.43) and (4.44) that $\boldsymbol{v} = \mathbf{0}$. The final equations in (4.42) are: $-\boldsymbol{f}_g = \boldsymbol{f}_c$, which can be expressed as:

$$-\mathbf{r}_{\text{off}_B}^{\times}\mathbf{C}_{BI}\mathbf{F}_{g_I} = \boldsymbol{\tau}_{c_B} = \mathbf{u}_\tau\ , \qquad \text{and} \qquad \mathbf{F}_{g_I} = -\mathbf{C}_{IB}\mathbf{F}_{c_B}\ . \tag{4.45}$$

The second equation can be further expanded as follows:

$$\begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = -\begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ u_F \end{bmatrix}, \tag{4.46}$$

where the first and second rows lead to: $\sin(\theta) = 0$, and $-\sin(\phi)\cos(\theta) = 0$ under the assumption of positive thrust control inputs $F_i > 0$. This results in $\phi = 0 \pm n\pi$ and $\theta = 0 \pm n\pi$ for $n \in \mathbb{Z}$. Assuming only upright attitude, the equilibrium Euler angles $\boldsymbol{\theta} = \begin{bmatrix} 0 & 0 & \bar{\psi} \end{bmatrix}^T$ where $\bar{\psi}$ is an arbitrary yaw angle. This also means that the rotation matrix $\mathbf{C}_{IB} = \mathbf{C}_z(\bar{\psi})$ the principal rotation matrix about the vertical axis. As a result the equilibrium state is:

$$\overline{\mathbf{x}} = \begin{bmatrix} \overline{x} & \overline{y} & \overline{z} & 0 & 0 & \overline{\psi} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T , \tag{4.47}$$

which describes the quadrotor in hover at an arbitrary position in space and at an arbitrary yaw attitude.

In terms of the control inputs at the equilibrium state, the expressions in (4.45) reduce to the following:

$$\overline{\mathbf{u}} = \begin{bmatrix} \overline{u}_F \\ \overline{\mathbf{u}}_\tau \end{bmatrix} = \begin{bmatrix} mg \\ -\mathbf{r}^{\times}_{\text{off}_B} \mathbf{C}_{BI} \mathbf{F}_{g_I} \end{bmatrix} , \tag{4.48}$$

which implies that the net lift from all four rotors $F_1 + F_2 + F_3 + F_4 \triangleq u_F = mg$ is equal to the weight of the quadrotor and the payload it is carrying. It also shows that the torque applied by the motors about the quadrotor geometric center exactly balances out the torque that the offset center of gravity applies to the quadrotor. Both of these results are logical for a quadrotor in hover with an offset center of gravity position.

4.3.2 Quadrotor Load Limits Under Design Constraints

Ultimately, the theoretical limitations on the payload mass properties that can be carried by a quadrotor are a direct result of the design parameters of the airframe. In practice, the design of the controller dictates how close this limit will be approached.

The control inputs discussed in Section 4.2.6 have limits in the amount of thrust that can be produced. The thrust envelope and the length of the quadrotor arms limit the total net force and torque that can be applied to the airframe. This translates into a maximum total mass of the quadrotor, and maximum CG offset to maintain hover. These limitations are design dependent and will be different for different quadrotors. From (4.48), the parameters that influence the maximum mass and CG offset are the thrust constant k_f, the quadrotor arm length from the geometric center l and the minimum and maximum propeller rotational speeds ω discussed in Section 4.2.6. To illustrate the maximum mass and CG offset, the parameters from a typical quadrotor found experimentally in [2] are

used.

The parameters of the Hummingbird Quadrotor [2] are:

$$1200 < \omega < 7800 \quad [\text{rpm}] \quad , \quad l = 0.175 \quad [m] \quad \text{and} \quad k_f = 6.11 \times 10^{-8} \quad \left[\frac{N}{\text{rpm}^2}\right] .$$

Figure 4.2 shows the contour plot of the maximum mass that can be supported at each horizontal CG offset location. As expected, the maximum mass decreases the further the center of mass is located from the geometric center, since a larger thrust is required to produce a counter moment. The contours are square because the motor pairs produce moments independently in the pitch and roll axes, which are perpendicular, and the quadrotor frame is symmetric.

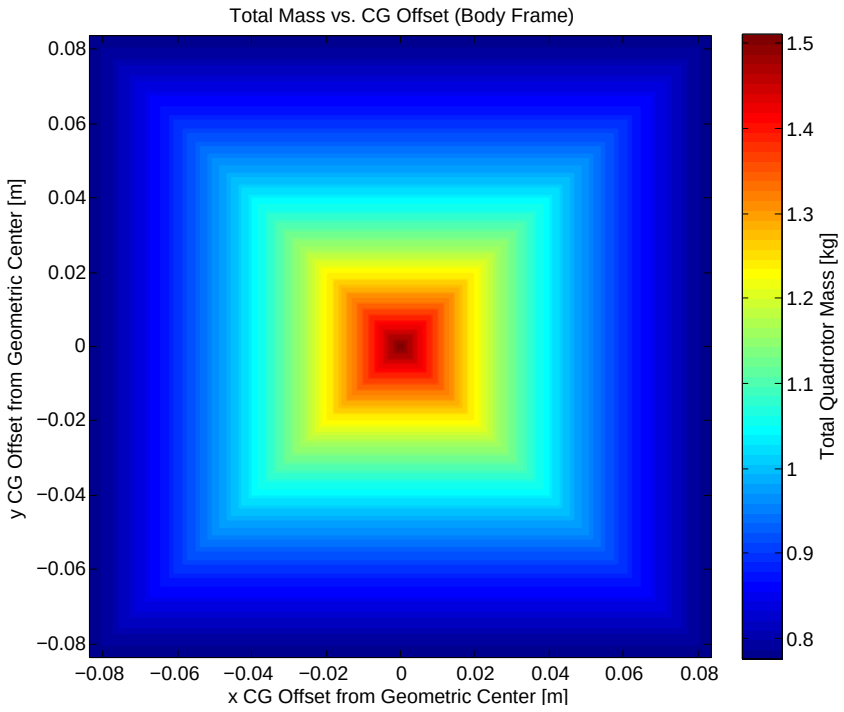


Figure 4.2: Mass limits due to actuator limits.

Load limits under maneuvering flight can be calculated from the dynamic equations of motion in (4.40). Further design constraints on the desired translational and angular accelerations are necessary to solve the equations of motion. As a result of the quadrotor symmetry, the contours will always be square in plots similar to Figure 4.2 of mass properties given performance constraints. Rectangular contours will be obtained for asymmetric performance specifications.

Chapter 5

Nonlinear Observability Analysis

This section discusses the observability of the unknown mass properties in the augmented full nonlinear general model extended from the dynamic model in (4.40) derived previously. First, the definitions for nonlinear observability are presented. This is followed by the identification of the unknown parameters involved. The output model is chosen based on the realistic selection of sensor measurement types for a typical quadrotor configuration consisting of interoceptive and exteroceptive sensor measurements. Finally, the results of the observability analysis are presented.

The objective is to determine whether the unknown mass properties can be estimated from typical quadrotor output measurements.

5.1 Preliminaries

The notation used in this chapter is identical to [40]. Let $h : \mathbb{R}^n \to \mathbb{R}$ be a smooth scalar function, and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ be a smooth vector field on $\mathbb{R}^n$.

Definition (Lie Derivative)**.** The Lie derivative [21] of h with respect to $\mathbf{f}$ is a scalar function defined by:

$$L_{\mathbf{f}}h \triangleq \nabla h\mathbf{f} = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \cdots & \frac{\partial h}{\partial x_n} \end{bmatrix} \mathbf{f} \,, \tag{5.1}$$

where $\nabla(\cdot)$ is the gradient operator that maps a scalar function dependent on n variables into a $1 \times n$ matrix of its partial derivatives. Repeated Lie derivatives are defined recursively as follows:

$$L_{\mathbf{f}}^{i} h \triangleq L_{\mathbf{f}}(L_{\mathbf{f}}^{i-1} h) = \nabla(L_{\mathbf{f}}^{i-1} h)\mathbf{f} \qquad \text{for } i = 1, 2, \ldots, \text{ where:} \tag{5.2}$$

$$L_{\mathbf{f}}^{0} h \triangleq h \tag{5.3}$$

Consider the following control-affine system $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{y} \in \mathbb{R}^p$, where:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{f}_i(\mathbf{x}) u_i \ , \tag{5.4}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) & \cdots & h_p(\mathbf{x}) \end{bmatrix}^T \tag{5.5}$$

Definition (Observability Matrix)**.** [40] Given the system in (5.4) and (5.5), the observability matrix, $\mathcal{O}$ is defined as:

$$\mathcal{O} \triangleq \begin{bmatrix} (\nabla L_{\mathbf{f}}^{0} h_1)^T & \cdots & (\nabla L_{\mathbf{f}}^{0} h_p)^T & \cdots & (\nabla L_{\mathbf{f}}^{n-1} h_1)^T & \cdots & (\nabla L_{\mathbf{f}}^{n-1} h_p)^T \end{bmatrix}^T \tag{5.6}$$

Definition (Weak Local Observability)**.** [40] The system defined by (5.4) and (5.5) is weakly locally observable at $\mathbf{x}_0$ if $\text{rank}(\mathcal{O}) = n$ at $\mathbf{x}_0$.

5.2 Augmented Control-Affine State Equation

Unknown Mass Properties

The quadrotor carrying an unknown payload has ten unknown parameters associated with the unknown mass properties. These are:

1. Mass, m
2. Center of gravity location, $\mathbf{r}_{\text{off}_B} = \begin{bmatrix} x_{CG} & y_{CG} & z_{CG} \end{bmatrix}^T$
3. Second moment of inertia matrix, $\mathbf{J} = \mathbf{J}^T = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix}$

Output Model

Typical sensors onboard a quadrotor include an Inertial Measurement Unit (IMU) containing a 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer, a GPS sensor or other positioning system such as a Vicon motion capture system, and an altimeter. These data provide information on the vehicle's position and attitude, from which all states are observed directly with noise. For the purpose of analyzing the observability of the unknown mass properties, it is assumed that the full state in (4.39) can be directly measured.

The output equation $\mathbf{y} = \mathbf{h}(\mathbf{x}) \in \mathbb{R}^p$ can therefore be written as follows:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathbf{r}_{b_I}^T & \boldsymbol{\theta}^T & \boldsymbol{v}^T \end{bmatrix}^T \quad . \tag{5.7}$$

Augmented State Equation

To analyze the observability of the unknown parameters, the quadrotor control-affine state equation in (4.40) is augmented with the unknown parameters discussed in Section 5.2. These parameters are assumed to be constant in time such that the additional rows in $\mathbf{f}_0$ to $\mathbf{f}_4$ contain only zeros. The augmented state equation is:

$$\begin{aligned}
&\dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{4} \mathbf{f}_i(\mathbf{x}) u_i \qquad \in \mathbb{R}^{22\times 1} \text{ , where:} \\
&\mathbf{x} \triangleq \begin{bmatrix} \mathbf{r}_{b_I}^T & \boldsymbol{\theta}^T & \boldsymbol{v}^T & m & \mathbf{r}_{\text{off}_B} & J_{xx} & J_{xy} & J_{xz} & J_{yy} & J_{yz} & J_{zz} \end{bmatrix}^T , \\
&\mathbf{f}_0(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{C}_{IB}(\boldsymbol{\theta}) \mathbf{v}_{BI_B} \\ \mathbf{S}^{-1}(\boldsymbol{\theta}) \boldsymbol{\omega}_{BI_B} \\ \boldsymbol{M}^{-1}(\boldsymbol{v}^{\times^T} \boldsymbol{M} \boldsymbol{v} + \boldsymbol{f}_g) \\ \mathbf{0}_{10\times 1} \end{bmatrix} , \quad \text{and} \quad \mathbf{f}_i(\mathbf{x}) \triangleq \begin{bmatrix} \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \\ \boldsymbol{M}^{-1} \mathbf{g}_i \\ \mathbf{0}_{10\times 1} \end{bmatrix} \quad \text{for } i = 1, \ldots, 4.
\end{aligned} \tag{5.8}$$

Note that the first 12 rows in (5.8) are identical to (4.40). The system described by (5.8) and (5.7) is in a form from which the observability matrix can be constructed and the observability of the unknown mass parameters can be analyzed.

5.3 Nonlinear Observability of the Augmented State Equation

The observability analysis of the system defined by (5.8) and (5.7) is elaborate and therefore omitted for brevity. Instead, a MATLAB script was written to determine the rank of the observability matrix from which the observability of the unknown parameters can be determined. This script is included in Appendix A.

The output of the script returns a rank 22 observability matrix constructed from only the zeroth and first order Lie derivatives of $\mathbf{h}(\mathbf{x})$ with respect to $\mathbf{f}_0(\mathbf{x})$, $\mathbf{f}_1(\mathbf{x})$, $\mathbf{f}_2(\mathbf{x})$, $\mathbf{f}_3(\mathbf{x})$ and $\mathbf{f}_4(\mathbf{x})$. Since this holds for any value of the extended state, the unknown mass properties are weakly observable. This means that having access to the output and the control inputs, the augmented state can be reconstructed using a state estimation algorithm, including the constant unknown mass properties. This result shows that a parametric adaptive controller with estimation of the unknown mass properties can successfully control a quadrotor UAV when these are not known.

Parametric based control has limitations which include the inability to compensate for other disturbances and unmodelled dynamics. These influence the estimates of the parameters over time which in turn affect the controller. For this reason, an $\mathcal{L}_1$ adaptive controller is used to control the quadrotor, which compensates for any disturbances including those arising from changes in the moments of mass. However with this approach, the parameters being updated within the controller become abstract, and any information about the unknown mass properties is lost.

Chapter 6

$\mathcal{L}_1$ Adaptive Control

The $\mathcal{L}_1$ adaptive controller was derived from a reparameterization of the MRAC architecture with the addition of a low-pass filter through which the control signal is passed. This additional attenuation of the high frequency component ensures that only the disturbances and unmodelled dynamics within the bandwidth of the control channel are compensated for. As a result, the low-pass filter decouples adaptation from robustness, because the high-frequency oscillations that can lead to instability are not propagated through the rest of the controller. This means that arbitrarily high adaptation gains can be selected in the adaptation, without resulting in unstable or undesirable transient behaviour. This overcomes the major drawback of the classical MRAC architecture, which is sensitive to the large adaptation gains. The trade-off between performance and robustness still exists and is addressed by the appropriate choice of low-pass filter. The reparameterization of MRAC and the design rationale behind the $\mathcal{L}_1$ adaptive control framework is provided in [41]. After introducing the general structure of the $\mathcal{L}_1$ adaptive controller and its components, the different architectures that were implemented on the AR.Drone quadrotor are presented. This section also includes a transient analysis of the closed-loop $\mathcal{L}_1$ adaptive output feedback architecture, which was not given in any of the literature.

6.1 Preliminaries

The notation used in this chapter is identical to [41], which does not denote vector or matrix entities with a boldface font. A typical $\mathcal{L}_1$ adaptive control loop for uncertain input disturbances is shown in Figure 6.1 and consists of an output predictor, a parameter adaptation law, and a low pass filter. A description of each block and its respective purpose in the adaptive controller is provided in the subsequent sections.

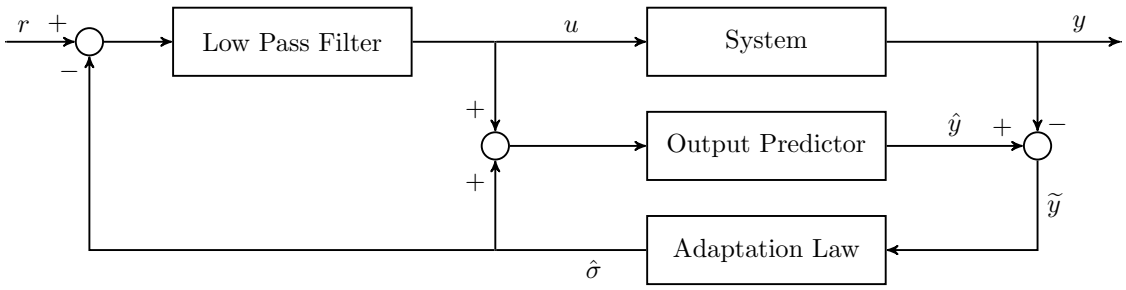


Figure 6.1: General $\mathcal{L}_1$ adaptive output feedback controller with input disturbances.

6.1.1 State or Output Predictor

This block within the architecture uses the current estimate of the unknown parameter, and the control input to the system to predict what the actual output should be. With respect to MRAC, the output predictor is analogous to the reference model. The method of the prediction is up to the designer, since any deviation between the predicted output and the actual output will be compensated for by a change in the estimate of the unknown parameter. This block therefore dictates the user specified desired performance, which the remainder of the $\mathcal{L}_1$ architecture will aim to achieve.

In general, the SISO output predictor with uncertain input disturbances has the following form:

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m\hat{x}(t) + B_m(u(t) + \hat{\sigma}(t)) \ , \qquad \hat{x}(0) = 0 \\ \hat{y}(t) &= C^T\hat{x}(t) \ , \end{aligned} \tag{6.1}$$

where $\hat{\sigma}(t)$ is the adaptive estimate of $\sigma(t)$, and $A_m \in \mathbb{R}^{n\times n}$, $B_m \in \mathbb{R}^{n\times 1}$ and $C \in \mathbb{R}^{n\times 1}$ describe a user specified SISO system with desired closed-loop behaviour. In the Laplace domain, this is equivalent to:

$$\hat{x}(s) = H(s)(u(s) + \hat{\sigma}(s)) \ , \qquad \text{where} \qquad H(s) \triangleq (s\mathbb{I} - A_m)^{-1}B_m \ , \tag{6.2}$$

$$\hat{y}(s) = M(s)(u(s) + \hat{\sigma}(s)) \ , \qquad \text{where} \qquad M(s) \triangleq C^T(s\mathbb{I} - A_m)^{-1}B_m \ , \tag{6.3}$$

the $n \times n$ identity matrix is denoted as $\mathbb{I}$, and $\hat{x}(s)$, $\hat{y}(s)$, $u(s)$ and $\hat{\sigma}(s)$ are the Laplace transforms of the respective time-domain signals.

6.1.2 Projection Based Adaptation Law

The adaptive estimate, $\hat{\sigma}$, is updated according to one of two methods. The first is the projection-based adaptation law:

$$\dot{\hat{\sigma}}(t) = \Gamma \text{Proj}(\hat{\sigma}(t), -mP\tilde{y}(t)) \ , \qquad \hat{\sigma}(0) = 0 \ , \tag{6.4}$$

where $\tilde{y}(t) \triangleq \hat{y}(t) - y(t)$, and $P > 0$ solves the algebraic Lyapunov equation $A_m^T P + P A_m = -Q$ for arbitrary $Q > 0$. In the implementation of this adaptation law, $Q = \frac{1}{2}$ was chosen resulting in $P = \frac{1}{m}$. $\Gamma \in \mathbb{R}^+$ is the adaptation rate subject to the lower bound as specified in [41]. Typically in $\mathcal{L}_1$ adaptive control, Γ is set very large. Experiments described in the next section were carried out with an adaptation rate of at least $\Gamma = 1000$.

The projection operator $\text{Proj}(\cdot, \cdot)$ is defined as:

Definition (Projection Operator)**.** [41] Consider the parameter θ with vector norm bound $\theta_{\max}$. The projection operator is defined as:

$$\text{Proj}(\theta, y) \triangleq \begin{cases} y & \text{if } f(\theta) < 0 \ , \\ y & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^T y \leq 0 \ , \\ y - \frac{\nabla f}{\|\nabla f\|} \left(\frac{\nabla f}{\|\nabla f\|}\right)^T y) f(\theta) & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^T y > 0 \ . \end{cases} \tag{6.5}$$

The scalar function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as follows:

$$f(\theta) \triangleq \frac{(\epsilon_\theta + 1)\theta^T \theta - \theta_{\max}^2}{\epsilon_\theta \theta_{\max}^2} \ , \tag{6.6}$$

where $\epsilon_\theta > 0$ is a user specified tolerance bound on the projection operator.

The operator ensures that the estimation of σ is guaranteed to remain within a specified convex set, determined by $\sigma_{\max}$ and ϵ_σ which are the design parameters of the function $f(\hat{\sigma})$. This is achieved by scaling the component normal to the convex set of the update to $\hat{\sigma}$, only when the estimate is close to the boundary. This occurs when $(\epsilon_{\hat{\sigma}} + 1)\hat{\sigma}^T \hat{\sigma} > \sigma_{\max}^2$, and $\tilde{y}$ has a component normal to the set such that $\nabla f^T \tilde{y} > 0$. At the boundary of the set, the component of $\tilde{y}$ normal to the set is driven to zero, by subtracting the component of $\tilde{y}$ normal to the convex set from it, when $f(\hat{\sigma}) = 1$. This ensures that the update to $\hat{\sigma}$ never updates beyond the boundary of the set. The projection is also used in other adaptive controllers [41]. In the case of SISO $\mathcal{L}_1$ adaptive output feedback under input disturbance, σ is a scalar, meaning that the projection operator linearly scales the update to zero as it approaches its bound defined by $f(\hat{\sigma})$.

6.1.3 Piecewise Constant Parameter Adaptation

The second adaptation law updates the parameter at the sampling rate of the $\mathcal{L}_1$ adaptive controller:

$$\hat{\sigma}(t) = \hat{\sigma}(iT_s) \ , \qquad t \in [iT_s, (i+1)T_s) \ ,$$

where T_s is the sampling time of the $\mathcal{L}_1$ control loop. This relation shows the piecewise constant parameter estimate, which is valid for the duration of each time step. The parameter estimate is calculated as follows:

$$\hat{\sigma}(iT_s) = -B_m^{-1}(e^{A_m T_s} - \mathbb{I})^{-1} A_m e^{A_m T_s} \tilde{y}(iT_s) \ , \qquad \text{for } i = 1, 2, \dots \ , \tag{6.7}$$

where $e^{A_m T_s}$ is the matrix exponential of $A_m T_s$. The advantage of this parameter estimation law is that the adaptation rate is as high as possible for the rate at which the controller is running. Additionally, for small or diagonal matrices A_m, the matrix exponential is not expensive to compute.

6.1.4 Control Law

The control input to the system is computed from the current adaptive parameter estimate, and the desired output that the system is required to track. As a result of the fast adaptation, the parameter estimate signal contains high frequency oscillations which the control law is required to attenuate. This is the key behind decoupling the adaptation from robustness [41].

For the single-input single-output case, the control input signal is the difference between the $\mathcal{L}_1$ desired trajectory signal, r and the adaptive estimate, $\hat{\sigma}$ after being filtered by the strictly-proper low-pass filter $C(s)$, with DC gain $C(0) = 1$:

$$u(s) = C(s)(r(s) + \hat{\sigma}(s)) \ . \tag{6.8}$$

In the case of multiple-input multiple-output (MIMO) $\mathcal{L}_1$ adaptive control, C(s) is a matrix of transfer functions. The trade-off between performance and robustness is dictated by the design of the low-pass filter. A larger cutoff frequency improves performance but simultaneously affects the robustness of the controller. The optimal design of the frequency response of the low-pass filter is still an open area of research [41].

The appropriate choice of $M(s)$ and $C(s)$ is governed by the $\mathcal{L}_1$-norm condition from which the adaptive controller bears its name. For the state feedback $\mathcal{L}_1$ adaptive

controller the $\mathcal{L}_1$-norm condition is defined as:

$$\|G(s)\|_{\mathcal{L}_1} L < 1 \ , \qquad \text{where} \qquad G(s) \triangleq H(s)(1 - C(s)) \ , \tag{6.9}$$

and L is a maximum bound on the absolute value of the unknown parameter. In output feedback, the uncertainty in the system cannot be decoupled from the $\mathcal{L}_1$-norm condition. The $\mathcal{L}_1$-norm condition for this case is presented below in the analysis of the closed-loop $\mathcal{L}_1$ output feedback controller.

6.1.5 Systems Theory and Norms

The following norm definitions are included here for completeness. A complete review of norms on vectors, matrices, and systems are provided in [41], and [42]

Definition (Vector norms)**.** Consider the vector $u = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}^T \in \mathbb{R}^m$, the norms on this vector are defined as follows:

$$\textbf{1-norm:} \qquad \|u\|_1 \triangleq \sum_{i=1}^{m} |u_i| \ , \tag{6.10}$$

$$\textbf{p-norm:} \qquad \|u\|_p \triangleq \left(\sum_{i=1}^{m} |u_i|^p \right)^{\frac{1}{p}} \ , \qquad \text{for } p \in [1, +\infty) \ , \tag{6.11}$$

$$\textbf{∞-norm:} \qquad \|u\|_\infty \triangleq \max_{1 \le i \le m} |u_i| \tag{6.12}$$

Definition ($\mathcal{L}$-spaces and $\mathcal{L}$-norms)**.** Consider the function $f : [0, +\infty) \to \mathbb{R}^n$, the norms on this function are defined as follows:

$$\textbf{$\mathcal{L}_1$-norm:} \qquad \|f\|_{\mathcal{L}_1} \triangleq \int_0^\infty \|f(\tau)\| d\tau < \infty \ , \tag{6.13}$$

$$\textbf{$\mathcal{L}_p$-norm:} \qquad \|f\|_{\mathcal{L}_p} \triangleq \left(\int_0^\infty \|f(\tau)\|^p d\tau \right)^{\frac{1}{p}} < \infty \ , \tag{6.14}$$

$$\textbf{$\mathcal{L}_\infty$-norm:} \qquad \|f\|_{\mathcal{L}_1} \triangleq \max_{1 \le i \le n} \left(\sup_{\tau \ge 0} |f_i(\tau)| \right) < \infty \ . \tag{6.15}$$

The space of piecewise continuous integrable functions with bounded $\mathcal{L}_p$-norm is defined to be the $\mathcal{L}_p^n$ space.

In the above definition of $\mathcal{L}$-norms, any of the vector norms defined above can be used. The $\mathcal{L}_p^n$ space can be extended by truncating the function $f(t)$:

Definition (Extended $\mathcal{L}_{pe}^n$ space)**.** [41] The extended space $\mathcal{L}_{pe}^n$ is defined as the space of functions:

$$\mathcal{L}_{pe}^n \triangleq \{f(t) | f_\tau(t) \in \mathcal{L}_p^n, \forall \tau \in [0, +\infty)\} \ , \tag{6.16}$$

where

$$f_\tau(t) \triangleq \begin{cases} f(t) & 0 \leq t \leq \tau \ , \\ 0 & t > \tau \ . \end{cases} \tag{6.17}$$

The extended $\mathcal{L}_{pe}^n$ space holds for any of the $\mathcal{L}$-norms defined above. The system norm is defined as follows:

Definition (System norm)**.** Consider the following linear time-invariant transfer function matrix $G(s)$, with impulse response $g(t) \in \mathbb{R}^{p \times m}$. The $\mathcal{L}_1$ norm of the system is defined as:

$$\|G(s)\|_{\mathcal{L}_1} \triangleq \max_{i=1,\ldots,p} \left(\sum_{j=1}^{m} \|g_{ij}\|_{\mathcal{L}_1} \right) . \tag{6.18}$$

The result of the following two lemmas from [41] will become useful in proving the bounded-input bounded-output stability of the $\mathcal{L}_1$ reference system.

Lemma 6.1.1. *Assume that $G(s) \in \mathcal{L}_1$. Then for arbitrary $d(t) \in \mathcal{L}_{\infty e}$,*

$$\|e_\tau\|_{\mathcal{L}_\infty} \leq \|G(s)\|_{\mathcal{L}_1} \|d_\tau\|_{\mathcal{L}_\infty} \ , \tag{6.19}$$

and $e(t) \in \mathcal{L}_{\infty e}$

Proof. Provided in [41]. □

Lemma 6.1.2. *A continuous time linear time-invariant proper system, $e(s) = G(s)d(s)$ with impulse response matrix $g(t)$ is bounded-input bounded-output stable if and only if its $\mathcal{L}_1$-norm is bounded, i.e. $G(s) \in \mathcal{L}_1$.*

Proof. Provided in [41]. □

6.2 Quadrotor $\mathcal{L}_1$ Adaptive Control Architectures

The general $\mathcal{L}_1$ adaptive control structure has many different forms and can be combined or nested within other control laws suitable for different systems. A complete overview

and theoretical discussion of the $\mathcal{L}_1$ adaptive controller is provided in [41]. In this section, several different $\mathcal{L}_1$ adaptive control architectures are presented that place the $\mathcal{L}_1$ adaptive output controller at different places within the cascaded control structure of the quadrotor. Each of these architectures has been implemented on the Parrot AR.Drone, and the results of each architecture are presented in Section 7. In each of the architectures introduced below, the controllers are implemented separately in each of the three directions x, y and z or the respective attitude angles that actuate the system in these directions. This means that any unknown coupling or asymmetric dynamics if present, are rejected by the $\mathcal{L}_1$ output feedback controller.

6.2.1 Translational Velocity Control Approach

The architecture presented in [31] is a good starting point for the $\mathcal{L}_1$ adaptive control implementation, because it has shown the feasibility for quadrotors. The main architecture is shown in Figure 6.2, where the $\mathcal{L}_1$ adaptive control was chosen to act on the translational velocity of the quadrotor. The additional proportional feedback controller around the $\mathcal{L}_1$ architecture was added such that the system tracks a position trajectory instead of translational velocity. This produces much more predictable results [31], and in the application to quadrotor control within a confined space such as the UTIAS Indoor Robotics Lab, this architecture also ensures that the quadrotor stays within the boundaries of the space. This is not necessarily the case when tracking a translational velocity profile.

However, there are some theoretical and practical drawbacks of this approach. As a result of the additional feedback loop, the proofs for stability and transient performance found in [41] are no longer valid. Unfortunately, only the architecture was presented in [31]; no proof on transient and steady-state performance was conducted. The implications of the additional feedback loop also means there is one more additional parameter to tune. As shown in Section 6.4, the tuning is further complicated in this architecture since the tuning parameters of the $\mathcal{L}_1$ controller are coupled in the effect they have on the natural frequency and damping ratio of the response.

The following definitions of the intermediate signals as shown in Figure 6.2 are used for the stability and transient proofs conducted in Section 6.3. The desired position trajectory is given by the signal $r_2(t)$. The error between the desired position $r_2(t)$ and the actual position $y_2(t)$ is passed to a proportional controller resulting in a desired translational velocity, $r_2(t)$. This is the input to the $\mathcal{L}_1$ adaptive output feedback controller shown in Figure 6.1. The error between the desired translational velocity r_1 and the

parameter estimate $\hat{\sigma}$, is passed through the low-pass filter obtaining the control input $u(t)$. This is fed into both the output predictor and the system, the latter consisting of the standard nonlinear quadrotor controller and the Parrot AR.Drone quadrotor. Similar to [31], the projection-based parameter adaptation law discussed above is used in this architecture. Measurements from the Vicon motion capture system allow the position y_2 and translational velocity y_1 to be measured for feedback in the controller.

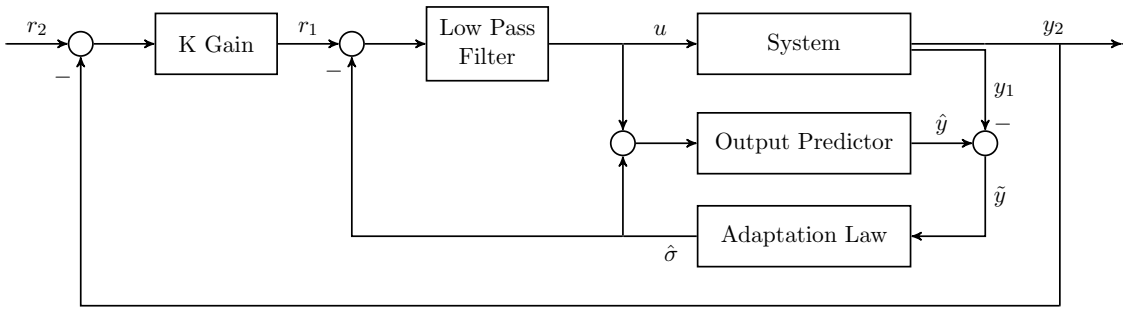


Figure 6.2: $\mathcal{L}_1$ adaptive output feedback controller in velocity.

The system block in Figure 6.2 contains a cascaded control architecture consisting of the nonlinear controller, the AR.Drone onboard controller and the quadrotor dynamic system. The nonlinear controller also uses the estimates of attitude from the Vicon motion capture system to compute its control signals, while the onboard quadrotor controller uses state estimates from its onboard inertial measurement unit.

6.2.2 Position Control Approach

The position based $\mathcal{L}_1$ adaptive controller is a direct implementation of the adaptive output feedback controller from [41]. As a result, the control block diagram is identical to Figure 6.1. The adaptive estimate $\hat{\sigma}$ is subtracted from the desired position trajectory $r(t)$ and is filtered through the low-pass filter. The control signal $u(t)$ representing a desired position is subsequently passed to the system consisting of the nonlinear controller and Parrot AR.Drone. The Vicon motion capture system is used to obtain measurements of the quadrotor position, $y(t)$. The piecewise constant adaptation law is used to update $\hat{\sigma}$ with a sampling rate equal to the rate at which the $\mathcal{L}_1$ control loop is operating. The stability and transient performance analyses are provided in [41]. Surprisingly, no previous work has attempted to implement this controller in the outer position control loop for quadrotor control.

6.2.3 Attitude Control Approach

The attitude based $\mathcal{L}_1$ adaptive controller is placed in the same position in the cascaded control architecture as in [32] of the Euler angle variant, except that desired roll and pitch attitude inputs to the $\mathcal{L}_1$ controller are obtained from the nonlinear controller instead. Therefore in this architecture shown in Figure 6.3, the $\mathcal{L}_1$ adaptive control augments the standard nonlinear controller. For convenience, the MIMO $\mathcal{L}_1$ adaptive control architecture is shown, but the implementation of this architecture involves diagonal low-pass filter and output predictor transfer function matrices resulting in decoupling of the roll, pitch and climb rate dynamics. Hence, Figure 6.3 can be represented as three separate control block diagrams.

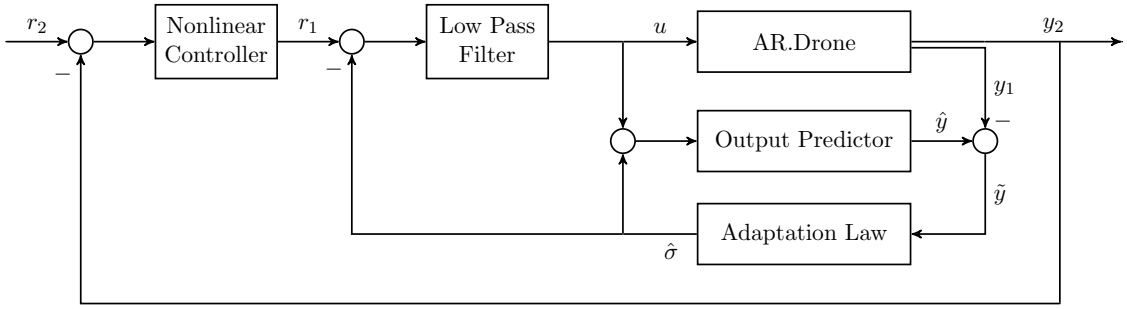


Figure 6.3: Roll, pitch and climb rate based $\mathcal{L}_1$ adaptive output feedback controller.

The definitions of each of the intermediate signals are defined as follows:

$$r_2 \triangleq \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad r_1 \triangleq \begin{bmatrix} \phi_{\text{des}} \\ \theta_{\text{des}} \\ \dot{z}_{\text{des}} \end{bmatrix}, \quad u \triangleq \begin{bmatrix} \phi_{\mathcal{L}_1\text{des}} \\ \theta_{\mathcal{L}_1\text{des}} \\ \dot{z}_{\mathcal{L}_1\text{des}} \end{bmatrix}, \quad \hat{y} \triangleq \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \\ \hat{\dot{z}} \end{bmatrix}, \quad y_1 \triangleq \begin{bmatrix} \phi \\ \theta \\ \dot{z} \end{bmatrix}, \quad y_2 \triangleq \begin{bmatrix} \phi \\ \theta \\ \dot{z} \end{bmatrix}, \tag{6.20}$$

where $r_2(t)$ is a desired position trajectory, $r_1(t)$ is the output of the nonlinear controller specifying a desired roll, pitch climb rate command. Instead of passing this signal into the Parrot AR.Drone onboard controller, r_1 is the input to the $\mathcal{L}_1$ adaptive controller. The control input u consisting of desired roll and pitch angles as well as the climb rate is obtained from the filtering the error between r_2 and the parameter estimates $\hat{\sigma}$. The control input is then passed to the AR.Drone onboard controller as well as the designer specified output predictor. The output of the AR.Drone consists of y_1 containing Vicon motion capture measurements of the attitude and climb rate, and y_2 containing Vicon measurements of the current quadrotor position.

Similar to the position-based approach in Section 6.2.2, the structure of the adaptive output feedback loop is unchanged.

6.3 Proof of the Stability and Transient Behaviour of the Closed-Loop $\mathcal{L}_1$ Adaptive Output Feedback Controller

As mentioned above in Section 6.2.1, the additional feedback loop shown in Figure 6.2 means that the proofs provided in [41] are no longer valid. To prove the stability and transient behaviour of this architecture, a generalized system with uncertainty is chosen similar to [41], and the analysis is conducted on an equivalent block diagram to Figure 6.2.

6.3.1 Problem Formulation

The analysis starts with the SISO system identical to [41] for output feedback:

$$y_1(s) = A(s)(u(s) + d(s)) \ , \tag{6.21}$$

where $A(s)$ is a strictly proper unknown transfer function, $u(s)$ is the Laplace transform of the input signal, and $d(s)$ is the Laplace transform of the disturbance signal defined as $d(t) \triangleq f(t, y_1(t))$, where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is an unknown map subject to the following assumption:

Assumption 6.3.1 (Lipschitz continuity)**.** There exist constants $L > 0$ and $L_0 > 0$, possibly arbitrarily large, such that the following inequalities hold uniformly in t:

$$|f(t, x) - f(t, y)| \leq L|x - y| \ , \text{ and} \tag{6.22}$$

$$|f(t, y)| \leq L|y| + L_0 \ . \tag{6.23}$$

When the $\mathcal{L}_1$ adaptive control is acting on the velocity, the output $y_1(t)$ is the velocity of the system. The additional proportional closed-loop feedback controller around the $\mathcal{L}_1$ controller acts on the position. As a result, the following relation holds:

$$y_2(s) \triangleq \frac{1}{s} y_1(s) \ , \tag{6.24}$$

meaning that the overall closed-loop structure tracks a position instead of translational velocity.

The objective of the closed-loop $\mathcal{L}_1$ adaptive output feedback control is to design a control input $u(t)$ such that $y_2(t)$ tracks $r_2(t)$.

Assumption 6.3.2. The desired trajectory $r_2(t)$ is a bounded piecewise continuous signal.

As discussed above in Section 6.2.1, this can be achieved by nesting the $\mathcal{L}_1$ adaptive output feedback controller within a proportional feedback controller such that $y_1(t)$ tracks $r_1(t)$. Following the same method to simplify the proof as presented in [41], the objective of the $\mathcal{L}_1$ adaptive controller is for $y_1(t)$ to track $r_1(t)$ according to a first-order reference system:

$$M(s) = \frac{m}{s+m}, \quad m > 0 \ . \tag{6.25}$$

6.3.2 $\mathcal{L}_1$ Adaptive Controller Definitions and $\mathcal{L}_1$-Norm Condition

Proceeding with the $\mathcal{L}_1$ formulation in [41], the system in (6.21) can be rewritten in terms of the first-order reference system (6.25):

$$\begin{aligned} y_1(s) &= A(s)u(s) + A(s)d(s) \\ &= A(s)u(s) + M(s)u(s) - M(s)u(s) + A(s)d(s) \\ &= M(s)u(s) + M(s)\left(\frac{(A(s) - M(s))u(s) + A(s)d(s)}{M(s)}\right) \\ &= M(s)(u(s) + \sigma(s)) \ , \end{aligned} \tag{6.26}$$

where the uncertainties of $A(s)$ and $d(s)$ are combined into σ:

$$\sigma(s) = \frac{(A(s) - M(s))u(s) + A(s)d(s)}{M(s)} \ . \tag{6.27}$$

Consider a strictly-proper low-pass filter $C(s)$ with $C(0) = 1$, a proportional gain $K \in \mathbb{R}^+$, as well as the following transfer functions:

$$H(s) \triangleq \frac{A(s)M(s)}{C(s)A(s) + (1 - C(s))M(s)} \ , \tag{6.28}$$

$$F(s) \triangleq \frac{1}{s + H(s)C(s)K} \ . \tag{6.29}$$

Lemma 6.3.1. *There exists a low-pass filter $C(s)$ with $C(0) = 1$ such that the transfer function $H(s)$ is stable if $A(s)$ can be stabilized by a proportional-integral controller.*

Proof. Shown in [41] when $C(s)$ is a first-order low-pass filter. □

Assumption 6.3.3 (Stability of $F(s)$)**.** There exists a proportional gain K such that the transfer function $F(s)$ is stable.

Remark. From Lemma 6.3.1, $H(s)$ is stable given the conditions on $A(s)$. These same conditions on $A(s)$ can be used to determine the stability of $F(s)$ using the root locus with the gain K. A related lemma and detailed proof will be established in future work.

Also, $C(s)$ and K are chosen such that the $\mathcal{L}_1$-norm condition is satisfied:

$$\|G(s)\|_{\mathcal{L}_1} L < 1 \ , \tag{6.30}$$

where

$$G(s) \triangleq H(s)(1 - C(s))F(s) \ . \tag{6.31}$$

For the proof of the bounded response of the closed-loop $\mathcal{L}_1$ adaptive output feedback controller, the following definitions will become useful:

$$H_0(s) \triangleq \frac{A(s)}{C(s)A(s) + (1 - C(s))M(s)} \ , \text{ and} \tag{6.32}$$

$$H_1(s) \triangleq \frac{(A(s) - M(s))C(s)}{C(s)A(s) + (1 - C(s))M(s)} \ . \tag{6.33}$$

From [41], it is shown that both $H_0(s)$ and $H_1(s)$ are strictly-proper stable transfer functions. Furthermore, the following expressions using the definitions of $H_0(s)$ and $H_1(s)$ will also be useful:

$$\begin{aligned} M(s)H_0(s) &= M(s)\bigg(\frac{A(s)}{C(s)A(s)+(1-C(s))M(s)}\bigg) \\ &= H(s)\ , \text{and} \end{aligned} \tag{6.34}$$

$$\begin{aligned} M(s)\Big(C(s)&+H_1(s)(1-C(s))\Big) \\ &= M(s)\Big(C(s)+\Big(\frac{(A(s)-M(s))C(s)}{C(s)A(s)+(1-C(s))M(s)}\Big)(1-C(s))\Big) \\ &= M(s)\Big(1+\Big(\frac{(A(s)-M(s))}{C(s)A(s)+(1-C(s))M(s)}\Big)(1-C(s))\Big)C(s) \\ &= M(s)\bigg(\frac{C(s)A(s)+(1-C(s))M(s)+(A(s)-M(s))(1-C(s))}{C(s)A(s)+(1-C(s))M(s)}\bigg)C(s) \\ &= M(s)\bigg(\frac{A(s)}{C(s)A(s)+(1-C(s))M(s)}\bigg)C(s) \\ &= H(s)C(s)\ . \end{aligned} \tag{6.35}$$

Finally, let ρ_r be defined as follows:

$$\rho_r \triangleq \frac{K\|H(s)C(s)F(s)\|_{\mathcal{L}_1}\|r_2\|_{\mathcal{L}_\infty}+\|G(s)\|_{\mathcal{L}_1}L_0}{1-\|G(s)\|_{\mathcal{L}_1}L}\ . \tag{6.36}$$

6.3.3 Revised Closed-loop $\mathcal{L}_1$ Adaptive Control Architecture

Figure 6.2 can be rearranged into the following SISO closed-loop $\mathcal{L}_1$ adaptive controller architecture shown in Figure 6.4.

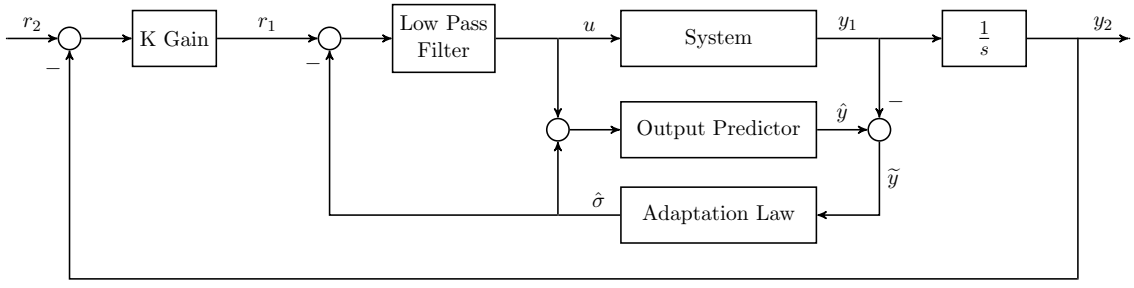


Figure 6.4: Closed-loop $\mathcal{L}_1$ adaptive output feedback controller.

The specific individual components of the $\mathcal{L}_1$ architecture are introduced below upon

substitution of the quantities introduced in Section 6.2.1 into the $\mathcal{L}_1$ adaptive output structure in Section 6.1. The equations for the outer proportional control loop are also shown here.

Output Predictor

The following output predictor is obtained upon substitution of (6.25):

$$\dot{\hat{y}}(t) = -m\hat{y}(t) + m(u(t) + \hat{\sigma}) \ , \qquad \hat{y}(0) = 0 \ , \tag{6.37}$$

where $\hat{\sigma}$ is the adaptive estimate of σ. In the Laplace domain, this is equivalent to:

$$\hat{y}(s) = M(s)(u(s) + \hat{\sigma}(s)) \ . \tag{6.38}$$

Adaptation Law

The adaptive estimate, $\hat{\sigma}$, is updated according to the projection-based adaptation law introduced in Section 6.1.2.

Control Law

The control input signal is the difference between the $\mathcal{L}_1$ desired trajectory signal, r_1 and the adaptive estimate, $\hat{\sigma}$ after being filtered by the low-pass filter $C(s)$:

$$u(s) = C(s)(r_1(s) + \hat{\sigma}(s)) \ . \tag{6.39}$$

Closed-Loop Feedback

The following equations describe the closed-loop feedback loop around the $\mathcal{L}_1$ controller:

$$y_2(s) = \frac{1}{s} y_1(s) \ . \tag{6.40}$$

The negative feedback is defined as follows:

$$r_1(s) = K(r_2(s) - y_2(s)) \ , \tag{6.41}$$

where the objective is for y_2 to track r_2.

The equations describing the closed-loop $\mathcal{L}_1$ output feedback architecture are (6.38), (6.4), (6.39), (6.40) and (6.41).

6.3.4 Transient and Steady-State Performance

Lemma 6.3.2. *Let $C(s)$, $M(s)$ and K satisfy the $\mathcal{L}_1$-norm condition in* (6.30)*. Then the following closed-loop reference system:*

$$\begin{aligned} y_{2,ref}(s) &= F(s)H(s)\big(C(s)Kr_2(s) + (1 - C(s))d_{ref}(s)\big) \\ d_{ref}(t) &\triangleq f(t, y_{2,ref}(t)) \end{aligned} \tag{6.42}$$

is bounded-input bounded-output stable.

Proof. Since $r_2(t)$ is bounded and $H(s)$, $C(s)$ and $F(s)$ are strictly proper stable transfer functions, taking the norm of the reference system and making use of Assumption (6.22) yields the following lower bound:

$$\begin{aligned} \|y_{2,\text{ref}_t}\|_{\mathcal{L}_\infty} &\leq K\|H(s)C(s)F(s)\|_{\mathcal{L}_1}\|r_2\|_{\mathcal{L}_\infty} + \|G(s)\|_{\mathcal{L}_1}\|f(t, y_{2,\text{ref}}(t))\|_{\mathcal{L}_\infty} \\ \|y_{2,\text{ref}_t}\|_{\mathcal{L}_\infty} &\leq K\|H(s)C(s)F(s)\|_{\mathcal{L}_1}\|r_2\|_{\mathcal{L}_\infty} + \|G(s)\|_{\mathcal{L}_1}(L\|y_{2,\text{ref}_t}\|_{\mathcal{L}_\infty} + L_0) \ , \end{aligned} \tag{6.43}$$

where $\|y_{2,\text{ref}_\tau}\|_{\mathcal{L}_\infty}$ is the truncated $\mathcal{L}_\infty$-norm of the signal $y_{2,\text{ref}}(t)$ up to $t = \tau$. From the $\mathcal{L}_1$-norm condition in (6.30) and the definition of ρ_r in (6.36):

$$\|y_{2,\text{ref}_t}\|_{\mathcal{L}_\infty} \leq \rho_r \ . \tag{6.44}$$

This result holds uniformly, so $\|y_{2,\text{ref}}\|_{\mathcal{L}_\infty}$ is bounded. Hence, from Lemma 6.1.2 the closed-loop reference system in (6.42) is bounded-input bounded-output stable. □

Theorem 6.3.1. *Consider the system in* (6.21)*, with a control input from the closed-loop $\mathcal{L}_1$ output feedback adaptive controller defined in* (6.38)*,* (6.4)*,* (6.39)*,* (6.40) *and* (6.41)*. Suppose $C(s)$, $M(s)$ and K satisfy the $\mathcal{L}_1$-norm condition in* (6.30)*. Then the following bounds hold:*

$$\|\tilde{y}\|_{\mathcal{L}_\infty} \leq \gamma_0 \ , \tag{6.45}$$

$$\|y_{2,ref} - y_2\|_{\mathcal{L}_\infty} \leq \gamma_1 \ , \tag{6.46}$$

where $\tilde{y}(t) \triangleq \hat{y}(t) - y_1(t)$, $\gamma_0 \propto \sqrt{\frac{1}{\Gamma}}$ is defined in [41], and

$$\gamma_1 \triangleq \frac{\|\frac{F(s)H(s)C(s)}{M(s)}\|_{\mathcal{L}_1}}{1 - \|G(s)\|_{\mathcal{L}_1}L}\gamma_0 \ . \tag{6.47}$$

Proof. Theorem 4.1.1 in [41] proves the bound in (6.45) for the same conditions, so only the bound in (6.46) remains to be shown.

Let $\tilde{\sigma}(t) \triangleq \hat{\sigma}(t) - \sigma(t)$ where $\hat{\sigma}$ is the adaptive estimate, and σ is defined in (6.27). From the control law in (6.39):

$$\begin{aligned} u(s) &= C(s)(r_1(s) - \hat{\sigma}(s)) \\ &= C(s)r_1(s) - C(s)(\tilde{\sigma}(s) + \sigma(s)) \ . \end{aligned} \tag{6.48}$$

Substituting (6.48) into the system (6.26):

$$\begin{aligned} y_1(s) &= M(s)\big(u(s) + \sigma(s)\big) \\ &= M(s)\big(C(s)r_1(s) - C(s)\sigma(s) + \sigma(s) - C(s)\tilde{\sigma}(s)\big) \\ &= M(s)\big(C(s)r_1(s) + (1 - C(s))\sigma(s) - C(s)\tilde{\sigma}(s)\big) \ , \end{aligned} \tag{6.49}$$

as well as into (6.27) results in the following expression for σ:

$$\begin{aligned} \sigma(s) &= \frac{(A(s) - M(s))\big(C(s)r_1(s) - C(s)(\tilde{\sigma}(s) + \sigma(s))\big) + A(s)d(s)}{M(s)} \\ M(s)\sigma(s) &= (A(s) - M(s))\big(C(s)r_1(s) - C(s)\tilde{\sigma}(s)\big) + A(s)d(s) \\ &\quad - (A(s) - M(s))C(s)\sigma(s) \\ \big(A(s)C(s) + M(s) - M(s)C(s)\big)\sigma(s) &= (A(s) - M(s))C(s)\big(r_1(s) - \tilde{\sigma}(s)\big) + A(s)d(s) \\ \sigma(s) &= \frac{(A(s) - M(s))C(s)}{C(s)A(s) + (1 - C(s))M(s)}(r_1(s) - \tilde{\sigma}(s)) \\ &\quad + \frac{A(s)}{C(s)A(s) + (1 - C(s))M(s)}d(s) \ . \end{aligned}$$

From the definitions of $H_0(s)$ and $H_1(s)$ in (6.32) and (6.33) respectively:

$$\sigma(s) = H_1(s)(r_1(s) - \tilde{\sigma}(s)) + H_0(s)d(s) \ . \tag{6.50}$$

Substituting (6.50) into (6.49) results in:

$$\begin{aligned} y_1(s) &= M(s)\Big(C(s)r_1(s) + (1-C(s))\big(H_1(s)(r_1(s)-\tilde{\sigma}(s)) + H_0(s)d(s)\big) \\ &\qquad - C(s)\tilde{\sigma}(s)\Big) \\ &= M(s)\Big(C(s)(r_1(s)-\tilde{\sigma}(s)) + (1-C(s))\big(H_1(s)(r_1(s)-\tilde{\sigma}(s)) + H_0(s)d(s)\big)\Big) \\ &= M(s)\Big(C(s) + H_1(s)(1-C(s))\Big)\big(r_1(s)-\tilde{\sigma}(s)\big) + M(s)H_0(s)(1-C(s))d(s). \end{aligned}$$

From (6.35) and (6.34), this expression simplifies to:

$$y_1(s) = H(s)C(s)\big(r_1(s)-\tilde{\sigma}(s)\big) + H(s)(1-C(s))d(s)\ . \tag{6.51}$$

An expression for y_2 is obtained by substituting (6.51) and (6.41) into (6.40) as follows:

$$\begin{aligned} y_2(s) &= \frac{1}{s}y_1(s) \\ &= \frac{1}{s}\Big(H(s)C(s)\big(K(r_2(s)-y_2(s)) - \tilde{\sigma}(s)\big) + H(s)(1-C(s))d(s)\Big) \\ &= \frac{H(s)}{s}\Big(C(s)\big(Kr_2(s)-\tilde{\sigma}(s)\big) + (1-C(s))d(s)\Big) - \frac{H(s)C(s)K}{s}y_2(s) \\ &= \frac{1}{s+H(s)C(s)K}H(s)\Big(C(s)\big(Kr_2(s)-\tilde{\sigma}(s)\big) + (1-C(s))d(s)\Big)\ . \end{aligned}$$

Using the definition of $F(s)$ in (6.29) results in:

$$\begin{aligned} &= F(s)H(s)\Big(C(s)\big(Kr_2(s)-\tilde{\sigma}(s)\big) + (1-C(s))d(s)\Big) \\ &= F(s)H(s)\Big(C(s)Kr_2(s) + (1-C(s))d(s)\Big) - F(s)H(s)C(s)\tilde{\sigma}(s)\ . \end{aligned} \tag{6.52}$$

From the definition of $\tilde{y}$ in the adaptation law:

$$\tilde{y}(s) = \hat{y}(s) - y_1(s)\ .$$

Substitution of the output predictor and the output of the actual system in (6.26) results in the following expression for $\tilde{y}(s)$:

$$\begin{aligned}\tilde{y}(s) &= M(s)(u(s) + \sigma\hat{(}s)) - M(s)(u(s) + \sigma(s)) \\ &= M(s)(\hat{\sigma}(s) - \sigma(s)) \\ &= M(s)\tilde{\sigma}(s) \end{aligned} \tag{6.53}$$

Recalling the reference system in (6.42) and using the expression for y_2 in (6.52), the error between reference and actual systems, $y_{2,\text{ref}} - y_2$ is obtained as follows:

$$\begin{aligned} y_{2,\text{ref}}(s) - y_2(s) &= F(s)H(s)\Big(C(s)Kr_2(s) + (1 - C(s))d_{\text{ref}}(s)\Big) \\ &\quad - F(s)H(s)\Big(C(s)Kr_2(s) + (1 - C(s))d(s)\Big) - F(s)H(s)C(s)\tilde{\sigma}(s) \\ &= F(s)H(s)\big(1 - C(s)\big)(d_{\text{ref}}(s) - d(s)) - F(s)H(s)C(s)\tilde{\sigma}(s) \\ &= F(s)H(s)\big(1 - C(s)\big)(d_{\text{ref}}(s) - d(s)) - \frac{F(s)H(s)C(s)}{M(s)}M(s)\tilde{\sigma}(s) \ . \end{aligned}$$

Substituting the expression for $\tilde{y}(s)$ in (6.53) and the definition of G(s) in (6.31):

$$= G(s)(d_{\text{ref}}(s) - d(s)) - \frac{F(s)H(s)C(s)}{M(s)}\tilde{y}(s) \ ,$$

Finally, since the $\mathcal{L}_1$-norm of $G(s)$ exists, and $\frac{F(s)H(s)C(s)}{M(s)}$ is strictly proper and stable, the following bound holds by taking the truncated $\mathcal{L}_\infty$-norm:

$$\big\|y_{2,\text{ref}_t} - y_{2_t}\big\|_{\mathcal{L}_\infty} \leq \big\|G(s)\big\|_{\mathcal{L}_1}\big\|d_{\text{ref}_t} - d_t\big\|_{\mathcal{L}_\infty} + \left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}\big\|\tilde{y}_t\big\|_{\mathcal{L}_\infty} \ .$$

From Assumption (6.23):

$$\begin{aligned}\big\|y_{2,\text{ref}_t} - y_{2_t}\big\|_{\mathcal{L}_\infty} &\leq \big\|G(s)\big\|_{\mathcal{L}_1}L\big\|y_{2,\text{ref}_t} - y_{2_t}\big\|_{\mathcal{L}_\infty} + \left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}\big\|\tilde{y}_t\big\|_{\mathcal{L}_\infty} \\ &\leq \frac{\big\|\frac{F(s)H(s)C(s)}{M(s)}\big\|_{\mathcal{L}_1}}{1 - \big\|G(s)\big\|_{\mathcal{L}_1}L}\big\|\tilde{y}_t\big\|_{\mathcal{L}_\infty} \ . \end{aligned}$$

This bound holds uniformly, so:

$$\left\|y_{2,\text{ref}} - y_2\right\|_{\mathcal{L}_\infty} \leq \frac{\left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}}{1 - \left\|G(s)\right\|_{\mathcal{L}_1} L} \left\|\tilde{y}\right\|_{\mathcal{L}_\infty} .$$

From the bound in (6.45) proven in [41], the following bound also holds:

$$\left\|y_{2,\text{ref}} - y_2\right\|_{\mathcal{L}_\infty} \leq \frac{\left\|\frac{F(s)H(s)C(s)}{M(s)}\right\|_{\mathcal{L}_1}}{1 - \left\|G(s)\right\|_{\mathcal{L}_1} L} \gamma_0 .$$

Using the definition of γ_1 in (6.47):

$$\left\|y_{2,\text{ref}} - y_2\right\|_{\mathcal{L}_\infty} \leq \gamma_1 ,$$

which proves the second bound in (6.46). □

Remark. Similar to the $\mathcal{L}_1$ adaptive output feedback controller without the outer loop as discussed in [41], the performance of the closed-loop $\mathcal{L}_1$ adaptive output controller can be made arbitrarily close to that of the reference system in (6.42), by increasing the adaptive gain Γ. This reduces both the magnitudes of γ_0 and γ_1.

Remark. Both the proofs provided here and in [41] are under the assumption of first-order reference system $M(s)$ and low-pass $C(s)$. As shown in the simulation examples in [41], higher order filters have a larger attenuation for frequencies above the cutoff frequency. Hence, for the same $\mathcal{L}_1$-norm condition, a higher-order low-pass filter allows for a smaller adaptation gain, or smaller cutoff frequency to be chosen for the same level of performance [41]. The following third order low-pass filter was determined experimentally to be the best trade-off between complexity in implementation, and improved high-frequency attenuation:

$$C(s) = \frac{3\omega_c^2 s + \omega_c^3}{(s + \omega_c)^3} . \tag{6.54}$$

To keep the number of tuning parameters low, a first-order reference system of the form in (6.25) is implemented in the experiments. In total, there are three tuning parameters associated with this SISO architecture. These are the low-pass filter cutoff frequency ω_c, the pole m of the reference system, and the proportional gain K of the outer loop.

6.4 Guideline for Tuning an $\mathcal{L}_1$ Adaptive Controller

Tuning of the $\mathcal{L}_1$ adaptive controller is not covered in any of the $\mathcal{L}_1$ adaptive control literature including [41]. Tuning in [31] was conducted offline by optimizing the performance of a quadrotor model obtained through system identification. Finding the parameters of the $\mathcal{L}_1$ controller for optimal tracking performance is not trivial if the parameter adaptation is not fast enough. In the projection-based adaptation, the gain Γ must be set large, or the sampling time of the control loop in the piecewise constant adaptation T_s must be set sufficiently small. A theoretical lower bound for Γ is provided in Chapter 4 of [41], but it is not trivial to calculate. In practice, the piecewise constant parameter adaptation sampling time should be chosen to update at the same rate at which the $\mathcal{L}_1$ adaptive control loop is operating. Furthermore, tuning of the cutoff frequency and the output predictor is challenging as well, as both influence the intermediate signals within the $\mathcal{L}_1$ control loop.

To demonstrate the complexity of tuning the output predictor and the proportional gain in the translational velocity $\mathcal{L}_1$ controller, consider the following closed-loop reference system in (6.42) with perfect cancellation of the unknown parameters, and no high-pass frequency attenuation, i.e., $C(s) = 1$:

$$y_{\text{ref}}(s) = F(s)H(s)Kr_2(s) \ .$$

Since $C(s) = 1$, $H(s)$ reduces to $M(s)$:

$$\begin{aligned} y_{\text{ref}}(s) &= \frac{1}{s + M(s)K} M(s)Kr_2(s) \\ &= \frac{\frac{m}{s+m}K}{s + \frac{m}{s+m}K} r_2(s) \\ &= \frac{mK}{s(s+m) + mK} r_2(s) \\ &= \frac{mK}{s^2 + ms + mK} r_2(s) \ . \end{aligned} \tag{6.55}$$

The denominator can be expressed as $s^2 + 2\zeta\omega_0 s + \omega_0^2$ where ζ is the damping ratio and ω_0 is the natural frequency which specify the time-domain response of the system. For desired performance specifications ζ and ω_0, the controller parameters m and K can be chosen as follows:

$$m = 2\zeta\omega_0 \qquad \text{and} \qquad K = \frac{\omega_0}{2\zeta} \ . \tag{6.56}$$

The coupling shown in (6.56) is further complicated in the actual implementation of the $\mathcal{L}_1$ controller by the inclusion of the low-pass filter, and the fact that the disturbances are not perfectly cancelled out.

The procedure outlined in this section provides a guideline to tuning an $\mathcal{L}_1$ adaptive controller, and the intermediate signal behaviour that is desired for optimal tracking performance.

The main objective in the first stage of tuning is to find a combination of tuning parameters that ensure the system is stable. This is dependent on the system that is being controlled, and how deep the $\mathcal{L}_1$ architecture is embedded within the overall control framework. For a quadrotor, the deeper the $\mathcal{L}_1$ architecture is within the control design, the more difficult it is to achieve this objective. This is because the outside controllers are still able to destabilize the overall system, even if the $\mathcal{L}_1$ is stable but responds slowly. During this stage, it is also important to be able to switch to a safe backup controller, or land in the event the quadrotor diverges. It is also critical to ensure the system as a whole is operating reliably during the tuning process. Any communication timeouts, or any minor failures will affect controller performance even if the optimal tuning parameters have been obtained.

Although there is no recipe for achieving a stable system from which performance can be optimized, the following suggestions have been found to facilitate finding a compatible combination. Start off tuning with a relatively low cutoff frequency, and realistic desired response of the output predictor. If there are additional outer control loops, such as the proportional control loop in the velocity based $\mathcal{L}_1$ controller, these gains should be set low to ensure the time constant is larger than the operation of the $\mathcal{L}_1$ control loop. Only when a good response is obtained, should the gains of the outer loops be increased. If the projection based adaptation is used, a relatively large gain Γ should be chosen, on the order of 500 at a minimum, dependent on the low-pass filter design.

The key to high tracking performance of the $\mathcal{L}_1$ control loop is to have high frequency oscillations in the adaptive estimate channel $\hat{\sigma}(t)$, and have a smooth control signal $u(t)$. The challenge is to find the right balance between high cut-off frequency and a fast output predictor (i.e., pole farther from the origin on the left half plane real axis). A high frequency signal in the $\hat{\sigma}(t)$ channel can be obtained when the output predictor is designed to have a fast response. Since the input to the output predictor is the sum of the smooth control signal $u(t)$ and the parameter estimate $\hat{\sigma}$ containing high frequency oscillations, its output $\hat{y}$ will also contain high frequency oscillations that are then fed back into the adaptation law. In fact, the frequency of the oscillations are dictated by the output predictor as it constantly overshoots the actual output from the system. The

large adaptive gain will then cause the adaptive estimates to oscillate at the overshooting frequency of the output predictor with large amplitude. Note that the low-pass filter should not be used in adjusting the high frequency component in the $\hat{\sigma}(t)$ channel.

The low-pass filter is designed to match the bandwidth of the control channel of the system. The tuning task is to increase the cutoff frequency ω_c while maintaining high frequency oscillations in the adaptive estimate channel $\hat{\sigma}(t)$. Oscillations in the system output typically indicate that the cutoff frequency has been set too high. The reason for maximizing the cutoff frequency is because of the frequency response of a low-pass filter. For low cutoff frequencies, the phase shift is unnecessarily large in the bandwidth of the system control channel. This results in a delay between the desired input $r(t)$ and the corresponding signal in the control input $u(t)$. Besides increasing the cutoff frequency, another method of reducing the phase shift is to increase the order the low-pass filter.

Chapter 7

Experimental Results

7.1 Experimental Setup

This section discusses the nonlinear proportional-derivative controller used in combination with the $\mathcal{L}_1$ adaptive controllers described in the previous chapter. The UTIAS Indoor Robotics Lab is also introduced as well as the Vicon motion capture system used for accurate quadrotor state estimation. The platform used in all experiments is the Parrot AR.Drone 2.0 UAV.

7.1.1 Baseline Nonlinear Controller

The proportional-derivative controller forms a basis for the control of the Parrot AR.Drone quadrotor. It has been developed by the Dynamic Systems Lab prior to this project. In addition to serving as a baseline controller for the purpose of comparison, the $\mathcal{L}_1$ architectures developed in the previous section augment it, or share many of its components. A brief overview of its implementation is provided here. A detailed review of the controller is provided in [2].

The Parrot AR.Drone interface with the Robot Operating System (ROS) involves the following four control inputs: ϕ_{cmd}, θ_{cmd}, $\dot{\psi}_{cmd}$ and $\dot{z}_{cmd}$ in the body frame. The onboard controller then computes the required motor thrusts, F_1 to F_4. The objective of the nonlinear controller is to compute these commands, such that the quadrotor tracks a desired position $\mathbf{r}_{des} = (x_{des}, y_{des}, z_{des})$, horizontal translational velocity $\dot{\mathbf{r}}_{des} = (\dot{x}_{des}, \dot{y}_{des})$ and desired heading, ψ_{des} in the inertial frame. From the Vicon overhead motion capture system, the controller obtains measurements about the current position, $\mathbf{r} = (x, y, z)$, translational velocity, $\dot{\mathbf{r}} = (\dot{x}, \dot{y}, \dot{z})$, and Euler angles $\boldsymbol{\theta} = (\phi, \theta, \psi)$ in the inertial frame. This information can then be used to compute the desired input to the Parrot AR.Drone

controller which calculates the required propeller rotation for each motor to ensure the commanded roll, pitch, yaw rate and climb rate are tracked.

Nonlinear Control Law

From the error in horizontal velocity and position, a desired horizontal acceleration can be calculated. This value is dependent on desired transient performance through the tuning parameters τ_{xy} and ζ which are the time constant and damping ratio respectively:

$$\ddot{x}_{cmd} = \frac{2\zeta}{\tau_{xy}}(\dot{x}_{des} - \dot{x}) + \frac{1}{\tau_{xy}^2}(x_{des} - x) \ , \tag{7.1}$$

$$\ddot{y}_{cmd} = \frac{2\zeta}{\tau_{xy}}(\dot{y}_{des} - \dot{y}) + \frac{1}{\tau_{xy}^2}(y_{des} - y) \ . \tag{7.2}$$

The only method for the quadrotor to accelerate horizontally is by tilting in the desired direction of travel. When tilted, the thrust generated by the four motors contains a nonzero horizontal force component which will result in acceleration of the quadrotor in this direction. Before this operation can be done, the desired accelerations in the inertial frame must be rotated to be in the body-fixed frame, and the respective magnitudes must be less than the maximum available mass-normalized thrust. The mass-normalized thrust, T_m, is the magnitude of the acceleration of the quadrotor including the compensation against the force of gravity:

$$T_m = \left\| \ddot{\mathbf{r}} - \frac{1}{m}\mathbf{F}_{g_I} \right\|_2 = \sqrt{\ddot{x}^2 + \ddot{y}^2 + (g + \ddot{z})^2} \ . \tag{7.3}$$

The desired normalized horizontal accelerations can then be rotated by the yaw angle of the quadrotor as follows:

$$\begin{bmatrix} \ddot{x}_b \\ \ddot{y}_b \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \min(\ddot{x}_{cmd}, T_m)T_m^{-1} \\ \min(\ddot{y}_{cmd}, T_m)T_m^{-1} \end{bmatrix} \ . \tag{7.4}$$

The minimum is taken since the desired accelerations cannot be greater than the total available mass-normalized thrust. The resulting roll and pitch angle commands can then be calculated from trigonometry:

$$\phi_{cmd} = \sin^{-1}(\ddot{y}_b) \ , \qquad \text{and} \qquad \theta_{cmd} = \sin^{-1}(\ddot{x}_b) \ . \tag{7.5}$$

The climb and yaw rate commands are calculated from their respective errors as

follows:

$$\dot{z}_{cmd} = \frac{1}{\tau_z^2}(z_{des} - z) \ , \qquad \text{and} \qquad \dot{\psi}_{cmd} = \frac{1}{\tau_\omega}(\psi_{des} - \psi) \ . \tag{7.6}$$

7.1.2 Indoor Robotics Lab

The UTIAS Indoor Robotics Lab consists of the Vicon motion capture system an interfacing computer, and a flying space in which flying experiments can be conducted. The general system setup is shown in Figure 7.1. The Vicon system estimates the position of multiple reflective markers fixed to the rigid quadrotor body using ten infrared cameras and emitters located at known locations within the space. This system is capable of millimeter level positioning accuracy. The computer interface stores a unique model of the quadrotor markers and estimates quadrotor position and attitude at a nominal rate of 200 Hz. This data is sent via ethernet to another computer on which the $\mathcal{L}_1$ adaptive control algorithms are computing control inputs in the form of desired roll, pitch, yaw rate and climb rate commands. After the control inputs are computed, they are sent over a 2.4 GHz Wi-Fi wireless connection to the AR.Drone onboard controller. Using its onboard IMU sensor measurements the AR.Drone calculates the required motor speeds, completing the feedback loop. For safety, the flying space is surrounded by a net that prevents the quadrotor from exiting the designated flying area.

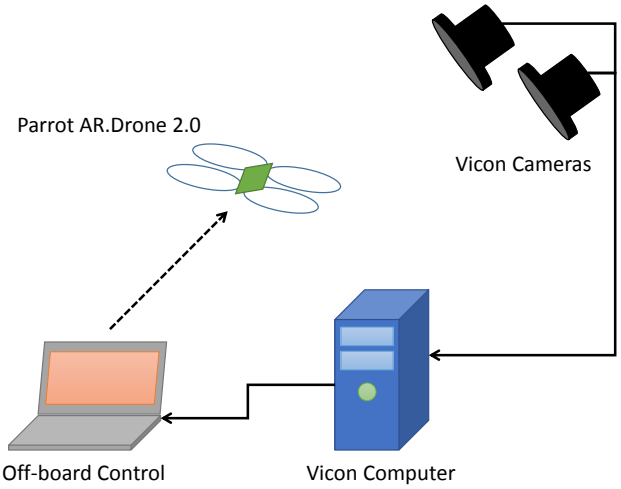


Figure 7.1: UTIAS Indoor Robotics Lab setup showing the components that constitute the complete control loop.

Parrot AR.Drone 2.0 Quadrotor

The Parrot AR.Drone 2.0 is a geared brushless quadrotor shown in Figure 7.2. The quadrotor operates on a 1.5 Ah three cell lithium polymer battery pack. The four rotors are protected by a lightweight extruded polypropylene foam hull that absorbs impact, results in safer operation, and prevents objects from contacting the propellers. The reflective Vicon markers are also mounted to this hull. This platform has been successfully used for research projects in the lab for the previous three years.

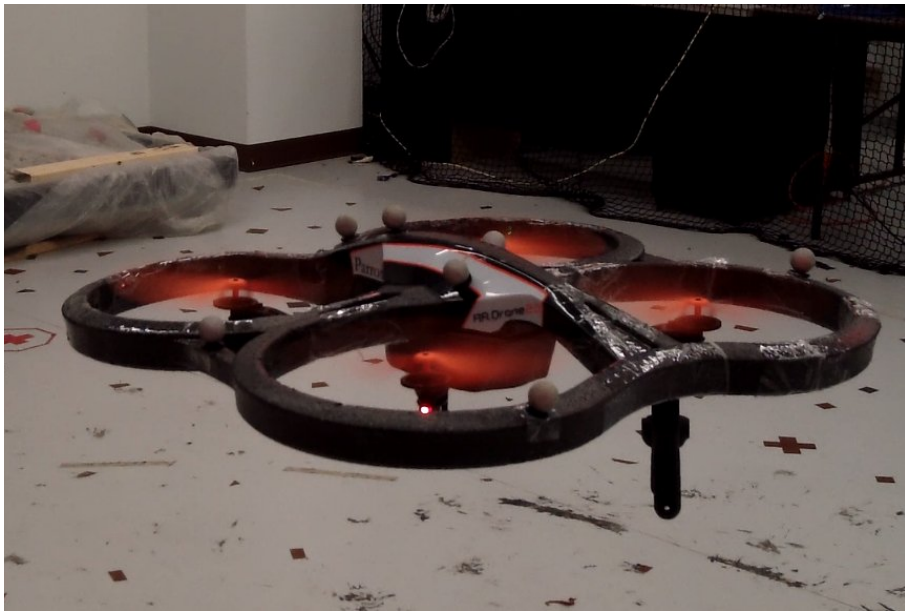


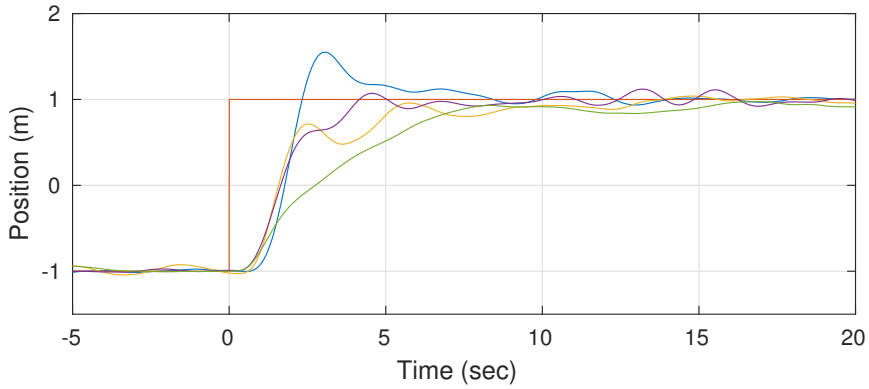
Figure 7.2: Parrot AR.Drone 2.0 showing extruded polypropylene hull and spherical Vicon reflective markers. A clamp is also shown attached to the aft-right landing leg to simulate a mass disturbance.

What follows is a comparison between the different control architectures in trajectory tracking performance and unknown disturbance rejection.

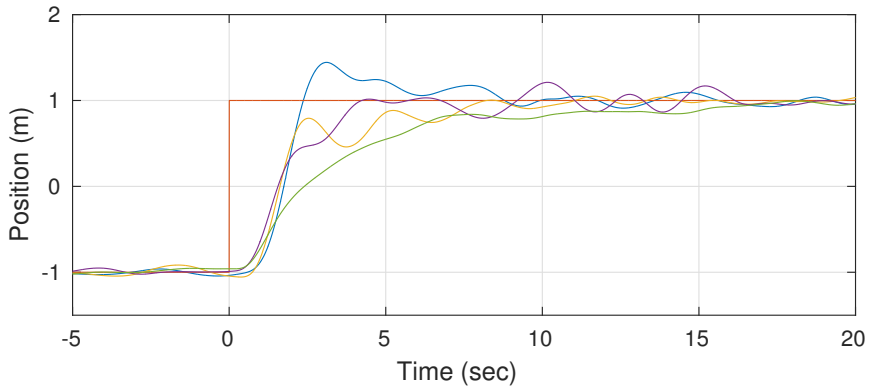
7.2 Comparison of $\mathcal{L}_1$ Adaptive Controllers

7.2.1 Step Input

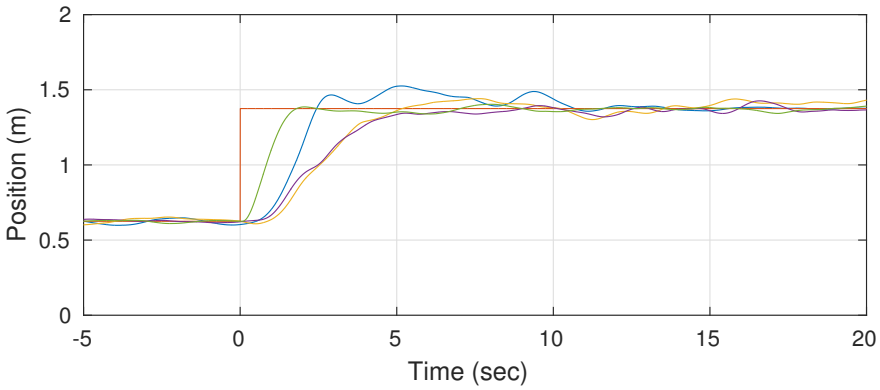
The time domain response to a step input in the x, y and z directions of the closed-loop $\mathcal{L}_1$ adaptive output controllers and the nonlinear controller are shown in Figure 7.3. The step input is 2 m in the x and y directions and 0.75 m in the z direction due to height limitations of the UTIAS Indoor Robotics Lab and ground proximity effects.



(a) Step response comparison in x direction.



(b) Step response comparison in y direction.



(c) Step response comparison in z direction.

Figure 7.3: Tracking performance comparison of the position-based $\mathcal{L}_1$ (blue), velocity-based $\mathcal{L}_1$ (purple), attitude-based $\mathcal{L}_1$ (yellow) and nonlinear controller (green) in response to a step input (red).

Framework	Average error [m] step input	Average error [m] hover disturbance
Position-based $\mathcal{L}_1$ adaptive control	0.380	0.126
Velocity-based $\mathcal{L}_1$ adaptive control	0.364	0.083
Attitude-based $\mathcal{L}_1$ adaptive control	0.403	0.078
Baseline nonlinear controller	0.549	0.233

Table 7.1: Average position error for the three $\mathcal{L}_1$ adaptive control and baseline nonlinear control architectures. The errors for both the step input and the disturbance under hover are shown here.

In order to quantitatively compare the trajectory tracking performance, the following error metric is used based on the average position error for 20 seconds after the step input:

$$J(\mathbf{y}) = \sum_{i=1}^{N} \frac{1}{N} \|\mathbf{r}_{\text{des},i} - \mathbf{y}_i\|_2 \ , \tag{7.7}$$

where $\mathbf{r}_{\text{des},i}$ is the desired position (x_{des}, y_{des}, z_{des}), and $\mathbf{y}_i$ is the actual position of the quadrotor (x, y, z) at each time-step i up to N when 20 seconds of time have elapsed. As shown in Figure 7.3, the 20-second comparison period is well beyond the settling time of any of the responses. The average tracking error as computed in (7.7) for each $\mathcal{L}_1$ control architecture and the baseline nonlinear controller are shown in the middle column of Table 7.1.

As a result of the long rise time in the x and y directions, the average position error of the nonlinear controller is the largest at 0.549 m for this trajectory, despite its superior tracking performance in the z direction. Although the transient behaviour is comparatively slower in the nonlinear control response, there are no oscillations in the x and y directions that are present in all three $\mathcal{L}_1$ adaptive control architectures as shown in Figure 7.3. This is primarily due to the tuning difficulties discussed in Section 6.4, and an improved tuning procedure for $\mathcal{L}_1$ adaptive control architectures has the potential to eliminate these oscillations and further improve tracking performance. For this reason, no conclusion can be made on the dependence of high tracking performance on the location of the adaptation within the control architecture, e.g., in the outside loop as is the case in the position-based approach, or the lowest level as is the case in the attitude approach.

The proportional-derivative control architecture in the attitude-based $\mathcal{L}_1$ adaptive controller increases the coupling between the tuning parameters, which further complicates the tuning process. Any resultant errors in roll and pitch attitudes subsequently

correspond to relatively large errors in position. The outer-loop damping ratio was chosen to dampen any steady-state oscillations, at the expense of poor transient performance. The largest average tracking error among the $\mathcal{L}_1$ architectures of 0.403 m is the result of the tuning difficulties encountered with the attitude-based architecture. As Table 7.1 shows, the position and translational velocity based $\mathcal{L}_1$ adaptive control architectures have the lowest average tracking error at 0.380 m and and 0.364 m respectively. Despite the tuning difficulties, all three $\mathcal{L}_1$ architectures outperform the baseline nonlinear controller in average tracking error.

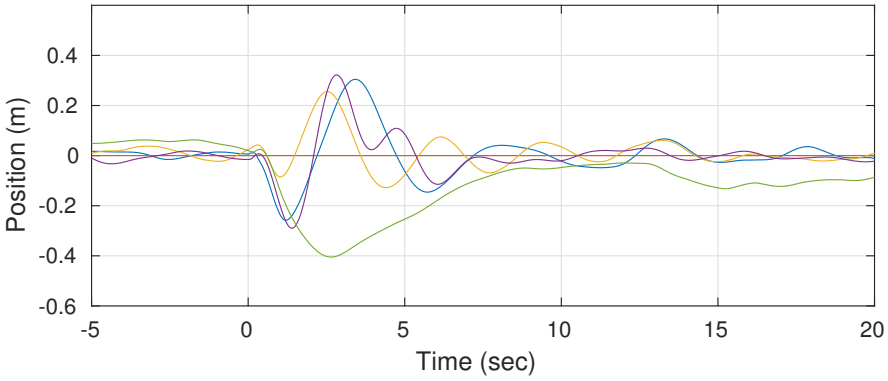
7.2.2 Compensation of Mass Disturbance in Hover

To compare the disturbance rejection capability of each controller, a 63 g mass is added to the aft-right landing gear leg 0.17 m from the geometric center in hover as shown in Figure 7.2. A comparison of the time domain response to this disturbance for each controller is shown in Figure 7.4.

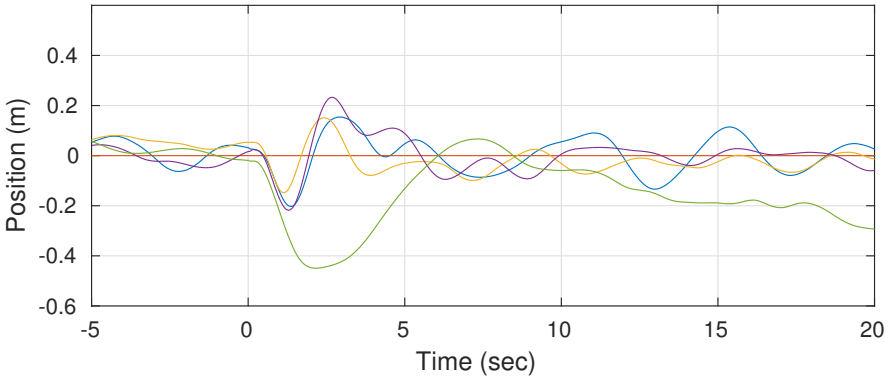
Similar to the step input comparison presented above, the average position error is computed according to (7.7) for 20 seconds after the fixed mass disturbance has been applied. This quantifies the effectiveness of each controller in compensating for the mass disturbance. The average position error of each controller is summarized in Table 7.1.

Among the $\mathcal{L}_1$ adaptive controllers, the position-based adaptive controller has the largest average position error of 0.126 m. Especially in the recovery of the z-direction, which reaches a maximum error of 0.467 m from the desired 1.5 m altitude. This is likely due to the fact that the adaptation law is in the outside control loop. Any phase shift in the low-pass filter and subsequent delayed response of the system contributes to the large errors. The lower the $\mathcal{L}_1$ adaptive controller is situated in the control architecture, the faster it can respond to sudden changes. This can be directly observed by the decreasing position errors of 0.083 m for the translational velocity-based $\mathcal{L}_1$ controller, and 0.078 m for the lowest level attitude based controller.

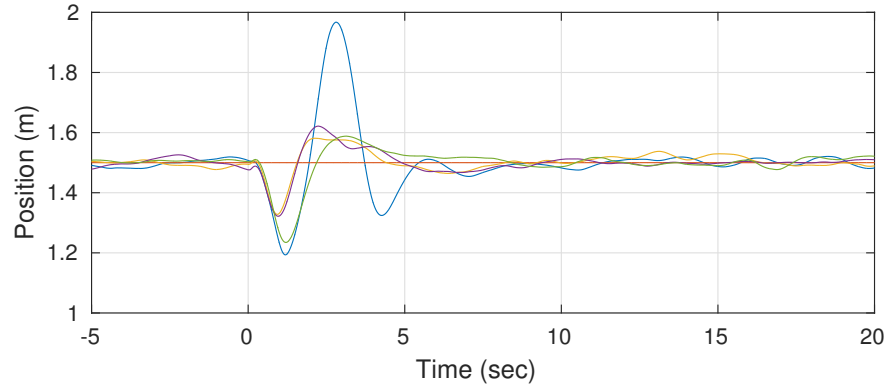
The ability for the baseline controller to compensate partially for the sudden mass change indicates there is an adaptation of some sort occurring on the onboard controller. A typical PID controller can be tuned to drive steady-state errors to zero, which is the result of the integrator term in the control law. However, as indicated by [43], a PID controller performance degrades under changing conditions, whereas the $\mathcal{L}_1$ adaptive controller that was implemented in roll and pitch control only is able to compensate for disturbances without any detrimental effects on performance. As indicated in Figure 7.4, the position error is only able to partially compensate for the increased and offset mass.



(a) Mass disturbance response during hover comparison in x direction.



(b) Mass disturbance response during hover comparison in y direction.



(c) Mass disturbance response during hover comparison in z direction.

Figure 7.4: Disturbance rejection performance comparison of the position-based $\mathcal{L}_1$ (blue), velocity-based $\mathcal{L}_1$ (purple), attitude-based $\mathcal{L}_1$ (yellow) and nonlinear controller (green) in response to a sudden change in mass properties at time $t = 0$. The quadrotor is required to maintain hover so the desired trajectory (red) is a constant.

The adaptation that is present is also comparatively slow, especially in the x and y directions. As a result, the baseline controller has the largest average position error of 0.233 m in the 20 second period after the clamp was attached.

7.3 Package Carrying Load Trajectory

The translational velocity-based $\mathcal{L}_1$ adaptive controller is used in demonstrating a quadrotor flight with an unknown package, because of its good trajectory tracking performance and mass property disturbance rejection. Since the goal of this project is to demonstrate high performance control in the presence of unknown mass properties, the design of an elaborate package carrying mechanism was not undertaken. The package attachment to the quadrotor was achieved using the Dual Lock reclosable fastener from 3M. At the expense of losing automated package retrieval and release capability, the main advantages of this design are mechanical simplicity, low cost and ease of implementation.

The horizontal tracking performance of the quadrotor carrying an unknown package following a simulated delivery trajectory is shown in Figure 7.5.

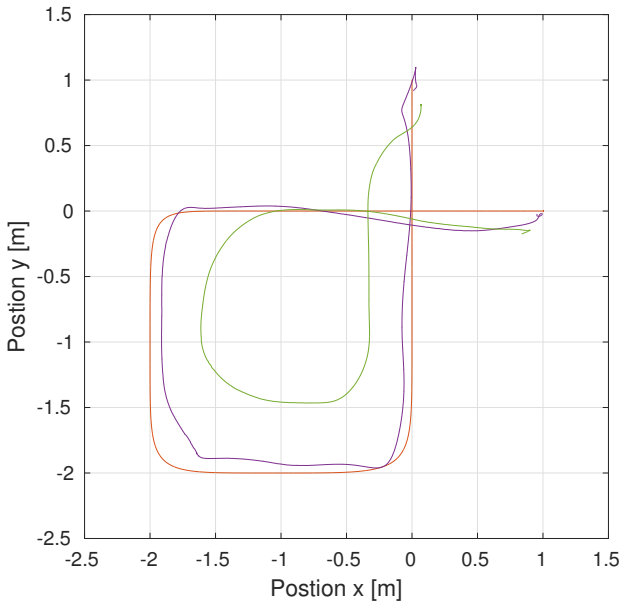


Figure 7.5: Horizontal package delivery trajectory tracking performance of the nonlinear controller(green), and the velocity based $\mathcal{L}_1$ adaptive controller (purple) carrying an unknown payload. The desired feasible trajectory is indicated by the red curve.

This result demonstrates the successful control of the quadrotor under unknown mass properties using the $\mathcal{L}_1$ adaptive controller, and meets the high-level objective specified in Section 1.2.

7.4 Closed-Loop $\mathcal{L}_1$ Output Feedback Combined Iterative Learning Control

The consistency and disturbance rejection benefits of the $\mathcal{L}_1$ adaptive controller means that it is a good candidate to combine it with other learning techniques that are sensitive to inconsistency. The results presented in this section show the performance of the $\mathcal{L}_1$ adaptive output feedback in translational velocity from Section 6.2.1, combined with a model-based iterative learning control (ILC) scheme developed by Karime Pereida of the Dynamic Systems Lab. The combined architecture is shown in Figure 7.6, and details of the ILC can be found in [44]. As in the above cases, the performance of this architecture is compared to the combined proportional-derivative nonlinear controller within the ILC framework.

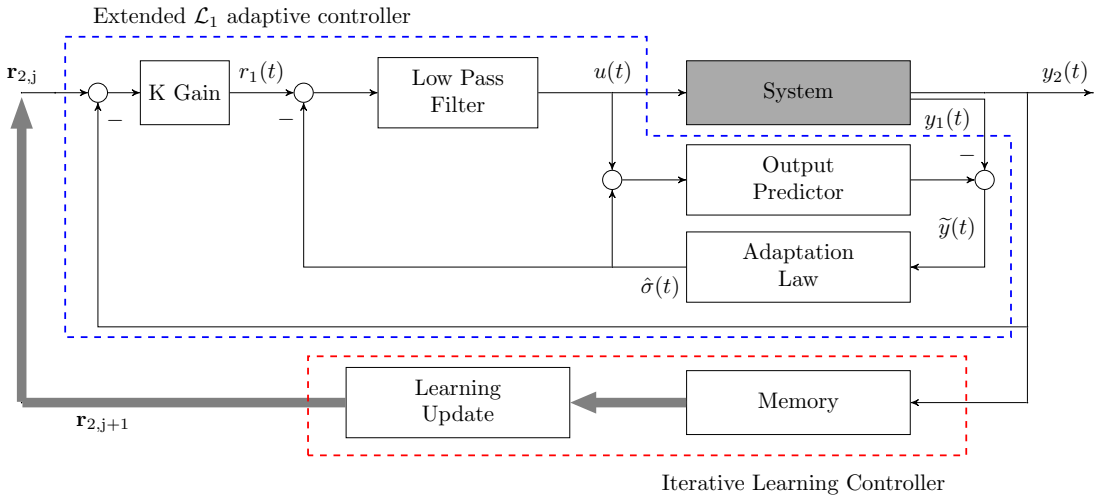


Figure 7.6: Closed-loop $\mathcal{L}_1$ adaptive output feedback controller.

The motivation behind this combined architecture is to improve tracking performance of a quadrotor following a repetitive trajectory under uncertainty and disturbances. The ILC optimizes in the iteration domain allowing the performance to be improved using data from the entire trajectory, as compared to time-domain methods that are only able to learn from past data. For convergence, the ILC requires a system that is repeatable and consistent, so any uncertainty or disturbances will affect the convergence of the ILC. The embedded $\mathcal{L}_1$ adaptive controller ensures that the system from the perspective of the ILC is repeatable and consistent. This means that the combined architecture can achieve high tracking performance in the presence of uncertainty [44]. The results presented here are from [44].

The $\mathcal{L}_1$ adaptive control in an ILC framework ($\mathcal{L}_1$-ILC) is used to minimize the trajectory tracking error of a quadrotor flying a three-dimensional trajectory under different dynamic conditions. The multi-input multi-output implementation is obtained by using 3×3 diagonal transfer function matrices for the low-pass filter and first-order output predictor. Each element of the three-dimensional signals and each diagonal element of the transfer function matrices correspond to the x, y and z inertial directions respectively. Since the transfer function matrices are diagonal, any coupling between the x and y dynamics are compensated for by the controller.

To test the performance of the proposed approach under unknown, changing disturbances, we change the dynamic behavior of the quadrotor by adding a *mass disturbance*. To create the mass disturbance a 50 g mass is suspended 55 cm below the back-left leg, 17 cm from the geometric center of the frame, creating a pendulum.

The performance of the proposed $\mathcal{L}_1$-ILC approach is compared to a pure ILC with an underlying, non-adaptive proportional-derivative controller (PD-ILC). To quantify the controller performance, the error in the system is defined as:

$$e = \frac{\sum_{i=1}^{N} \sqrt{(e_x(i))^2 + (e_y(i))^2 + (e_z(i))^2}}{N} \,, \tag{7.8}$$

where $e_x(i) = r^*_{2,x}(i) - y_{2,x}(i)$, $e_y(i) = r^*_{2,y}(i) - y_{2,y}(i)$ and $e_x(i) = r^*_{2,z}(i) - y_{2,z}(i)$ are the deviations from the desired trajectory in each axis. Three scenarios are considered to compare the performance of the two control frameworks: learning convergence, repeatability and performance under changing conditions. In all three scenarios the $\mathcal{L}_1$-ILC approach shows higher performance compared to the PD-ILC approach. The performance improvement is due to the ability of the $\mathcal{L}_1$ adaptive controller to make the system behave in a repeatable, predefined way, despite changes in the system dynamics. Consequently, the performance achieved with the learned trajectories is generalizable to a system with different dynamic properties (such as the suspended mass).

7.4.1 Learning Convergence

First, the convergence of the proposed $\mathcal{L}_1$-ILC framework is compared with the learning performance of the PD-ILC approach. The quadrotor learns to track a desired trajectory for ten iterations using each of the two frameworks: PD-ILC and $\mathcal{L}_1$-ILC. The errors of this initial learning process are depicted in the first ten iterations of Fig. 7.7. The proposed $\mathcal{L}_1$-ILC framework improves the trajectory tracking performance in only a few iterations. The initial error is larger for the PD-ILC approach and continues to be larger

until iteration eight when both frameworks achieve a similar performance.

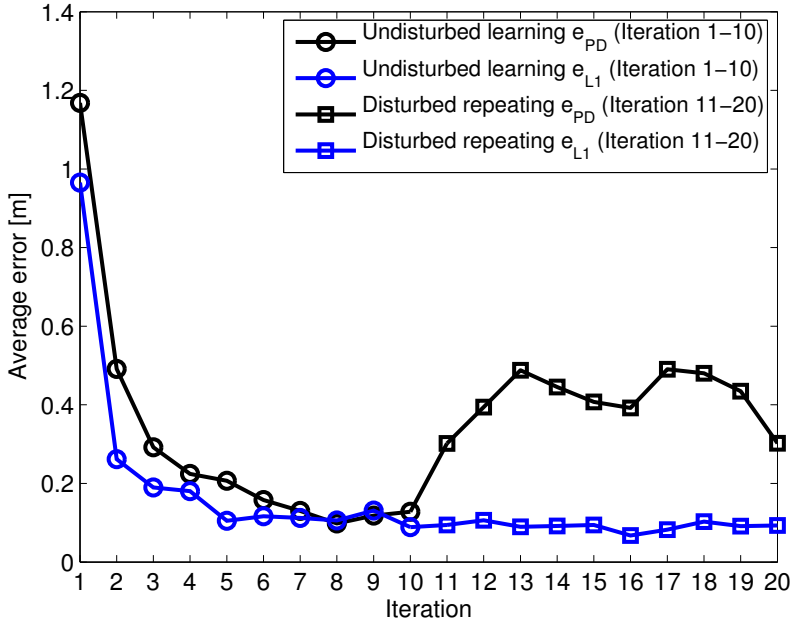


Figure 7.7: Repeatability of the learned trajectory at iteration 10 after a mass disturbance has been applied to the system. The error in the PD-ILC approach increases dramatically while the error in the $\mathcal{L}_1$-ILC approach remains close to the error before the disturbance.

After this initial learning process a mass disturbance is applied to the system. After the disturbance, the learned trajectory (at iteration ten) is repeated for ten more iterations with both $\mathcal{L}_1$-ILC and PD-ILC frameworks. The errors of this repeating process are depicted in Fig. 7.7. After the disturbance, the PD-ILC framework has a mean error of 0.413 m. Compared to an error of 0.127 m at iteration ten, this is a 323% increase after the mass disturbance is applied. The $\mathcal{L}_1$-ILC approach has a mean error of 0.091 m with an average error at iteration ten of 0.089 m. The $\mathcal{L}_1$-ILC approach has only a negligible increase in the error because the $\mathcal{L}_1$ adaptive controller is able to make the system behave in a repeatable way, despite the disturbances applied to the system.

7.4.2 Repeatability

As shown above, the $\mathcal{L}_1$-ILC approach is able to generalize the learned trajectory to the new situation with the added mass. To assess repeatability after a mass disturbance has been applied to the system, we compiled a total of five 10-iteration repetitions of the learned trajectory for each control framework. Fig. 7.8 shows the average error of the five sets at each iteration along with their standard deviations. The system is more repeatable

with the $\mathcal{L}_1$-ILC framework as the error and standard deviation are much smaller than with the PD-ILC framework. To be more specific, the minimum standard deviation of the PD-ILC approach (0.043 m) is almost twice the maximum standard deviation of the $\mathcal{L}_1$-ILC approach (0.023 m).

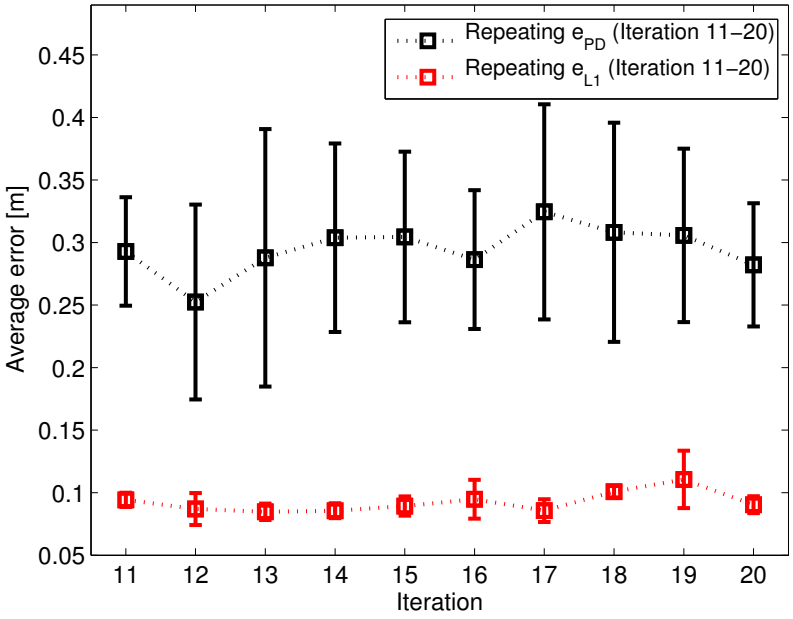
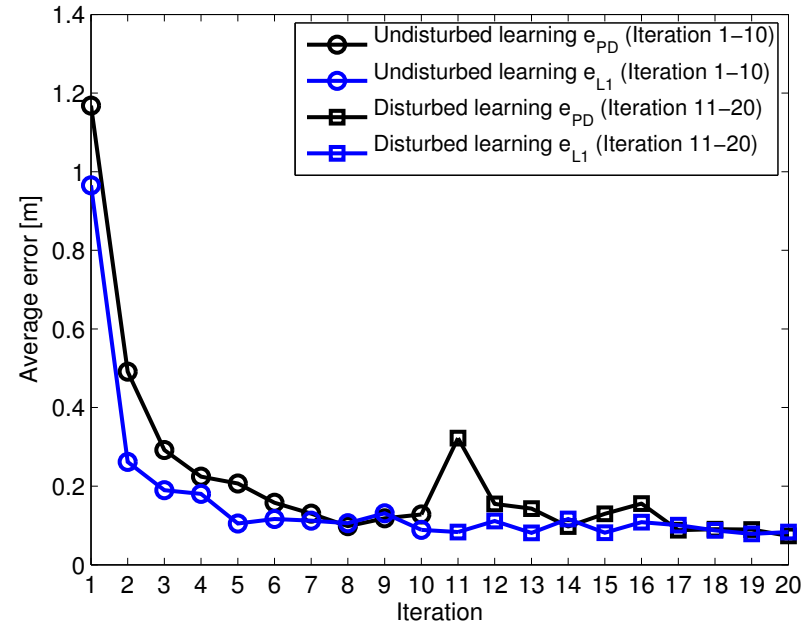


Figure 7.8: Mean of the error across five 10-iteration sets testing repeatability of the learned trajectory after a mass disturbance has been applied to the system. The PD-ILC approach displays a significantly larger standard deviation than the $\mathcal{L}_1$-ILC approach.

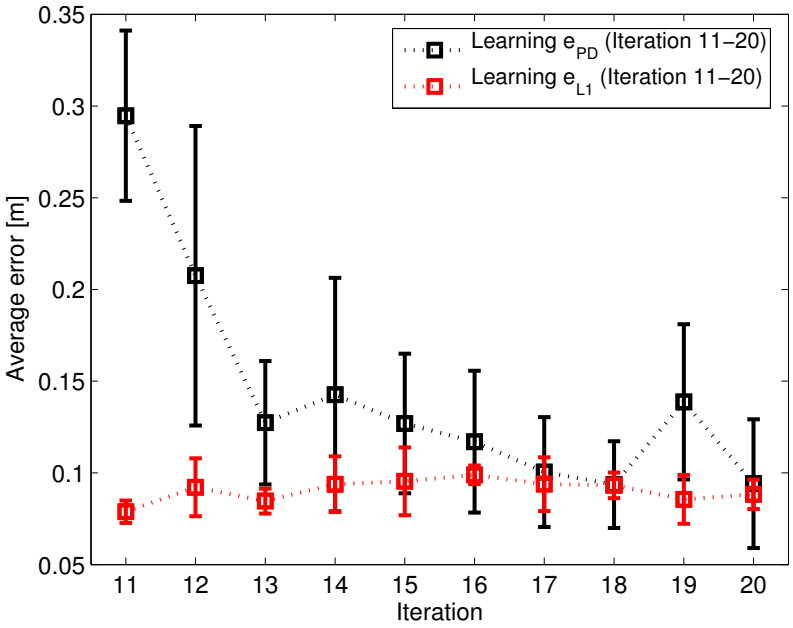
7.4.3 Performance under Changing Conditions

The ability of the system to continue to learn after a disturbance has been applied is also explored. The errors while the system is learning without disturbance (first ten iterations) and with a mass disturbance (last ten iterations) are shown in Fig. 7.9a. The error increases significantly in the PD-ILC framework after the disturbance is applied, see Table 7.2. This error rapidly decreases as the system learns; however, for some applications, this behavior may not be acceptable. The error in the $\mathcal{L}_1$-ILC framework does not change even after the mass disturbance has been applied (see Table 7.2).

The learning behavior is further explored by obtaining a total of five 10-iteration sets of the system learning after a mass disturbance is applied to the system. The average of the error and the standard deviation across the five sets are shown in Fig. 7.9b. Table 7.2 shows that the average error at iteration eleven for the PD-ILC approach is significantly



(a)



(b)

Figure 7.9: (a) Learning behavior of the PD-ILC and $\mathcal{L}_1$-ILC frameworks after a mass disturbance is applied to the system at the end of iteration ten. The error of the PD-ILC framework in iteration eleven (after the disturbance) increases dramatically, compared to iteration ten (before the disturbance); while the error of the $\mathcal{L}_1$-ILC framework in iteration eleven is similar to the error in iteration ten. (b) Average error across five sets of ten iterations of learning after a mass disturbance has been applied. The PD-ILC approach displays a significantly larger standard deviation than that of the $\mathcal{L}_1$-ILC approach.

Framework	Error at iteration 10	Error at iteration 11	Average error at iteration 11	Maximum standard deviation
PD-ILC	0.127 [m]	0.322 [m]	0.294 [m]	0.081 [m]
$\mathcal{L}_1$-ILC	0.089 [m]	0.083 [m]	0.078 [m]	0.018 [m]

Table 7.2: For the two controllers when the mass disturbance is applied after iteration ten: errors at iteration 10 and iteration 11 for a single set and average error at iteration 11 and maximum standard deviation of the five ten-iteration sets.

higher than for the $\mathcal{L}_1$-ILC approach. The standard deviation is notably higher for the PD-ILC approach than for the $\mathcal{L}_1$-ILC approach (see Table 7.2). The $\mathcal{L}_1$-ILC approach has the same behavior before and after the application of the disturbance and is able to reuse the learned trajectory at iteration ten.

The improved performance of the $\mathcal{L}_1$-ILC approach compared to the PD-ILC approach in the repeatability and learning scenarios presented is from the ability of the $\mathcal{L}_1$ adaptive controller to make the system behave in a repeatable, predefined way, despite changes in the system dynamics. As mentioned above, a repeatable system allows the ILC to improve its performance.

7.5 Continuity of the $\mathcal{L}_1$ Adaptive Control Tuning Parameters

If a change in low-pass filter cutoff frequency and the output predictor eigenvalue produce a continuous change in controller performance, then a learning scheme or optimization algorithm can be used to optimize the controller performance. This has the potential of further improving the tracking performance and possibly disturbance rejection without the large effort required during the tuning process. Although the automatic tuning of the $\mathcal{L}_1$ controller is beyond the scope of this project, the controller performance was quantified for different cutoff frequency and output predictor eigenvalue combinations.

Using the translational velocity $\mathcal{L}_1$ adaptive controller from Section 6.2.1, the cutoff frequency was varied from 1.0 to 2.0 rad/s in steps of 0.1 rad/s, and the output predictor pole was varied from -4.2 to -1.5 with a step size of 0.3. For each parameter combination pair, the cost function shown in equation 7.9 was used to quantify controller performance. It was chosen to penalize both poor tracking performance and large control inputs from

the $\mathcal{L}_1$ controller.

$$J(\mathbf{y}, \mathbf{u}) = \sum_{i=1}^{N} \frac{1}{N} \Big(3\|\mathbf{r}_{\text{des},i} - \mathbf{y}_i\|_2 + \|\mathbf{u}_i\|_2 \Big) \ . \tag{7.9}$$

This cost function contains the average tracking error metric in (7.7), but also contains a less weighted term to penalize large control inputs as well.

The cost function evaluated for each ($\omega_{x,y}$, $m_{x,y}$) pair from experimental data is shown in Figure 7.10. The minimum cost for tracking performance occurs around $(\omega_{x,y},\ m_{x,y}) = (1.8, -2.1)$.

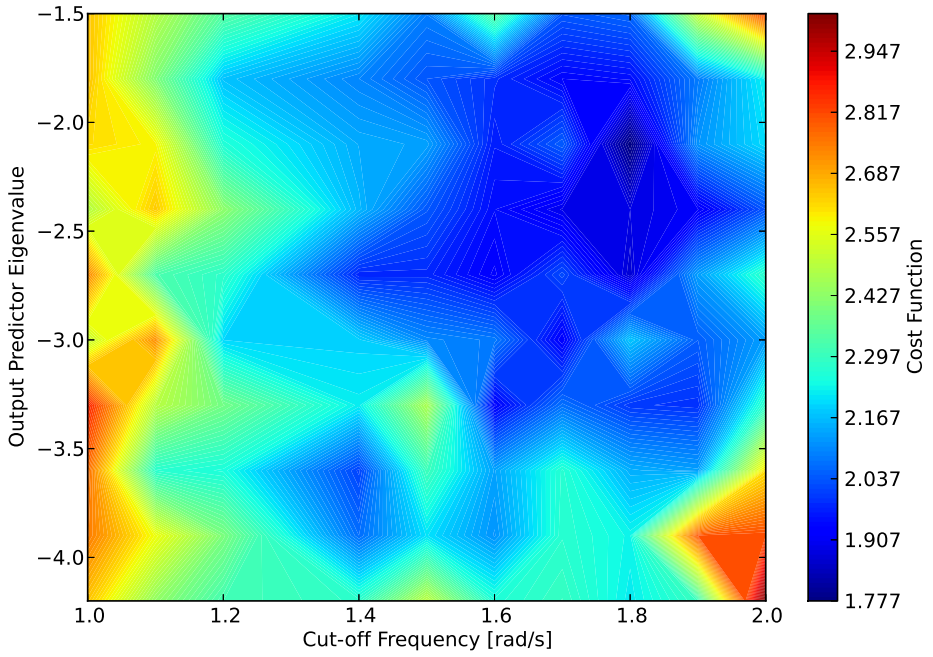


Figure 7.10: Contour plot of the cost function J($\mathbf{y}$, $\mathbf{u}$), evaluated for different values of $\omega_{x,y}$ and $m_{x,y}$.

The cost function is dominated by two high cost regions. For very low cutoff frequencies, the tracking is slow due to the low-pass filter phase shift resulting in a high penalty from a slow response. As cutoff frequency is increased, the response grows more aggressive from passing increasingly higher frequency signals into the system until large oscillations result in a high cost due to the large control inputs. The gradually changing colours in the contour plot confirms the continuity of the cost function.

Chapter 8

Concluding Remarks and Recommendations

8.1 Conclusion

The work presented here shows the complete derivation of a quadrotor dynamics model in three dimensions. In the derivation, no assumptions are made on the center of gravity location and the second moment of mass matrix preserving the generality of the mass properties. Within the scope of this project, these mass properties represent the uncertainty within the system. A complete nonlinear observability analysis was subsequently conducted showing local weak observability of all of the unknown mass properties if the quadrotor position, translational velocity, attitude, angular velocity and control inputs can be measured.

The system with unknown mass properties was chosen to be controlled using an $\mathcal{L}_1$ adaptive controller. This control architecture achieves improved performance and guaranteed robustness compared to classical adaptive control techniques. This is achieved due to its fast adaptation which is the result of the separation between adaptation and robustness. Three adaptive control architectures were designed for use on the Parrot AR.Drone, and were shown to outperform the standard nonlinear quadrotor controller in both tracking performance and hover under mass uncertainty. The $\mathcal{L}_1$ adaptive control architecture was also shown to be successfully integrated into an iterative learning control framework resulting in a quadrotor system capable of high performance trajectory tracking under uncertainty and disturbances. The adaptive controller ensured the system output remained consistent and repeatable, while the learning scheme learned to compensate for systematic tracking errors in the iteration domain. Finally, the pay-

load carrying task demonstrates the effectiveness of the $\mathcal{L}_1$ adaptive controller, and the achievement of the high-level objective.

8.2 Future Work

As mentioned in Section 7.5, $\mathcal{L}_1$ design for optimal tracking can be achieved through online learning methods due to the continuity of the cost function in the tuning parameters. Future work to achieve this can be implemented by using the safe optimization algorithm developed by [45]. Low-pass filter design can be parametrized by changing the design to a finite impulse response filter, allowing optimization schemes to more precisely adjust the frequency response of the filter.

The advantages of the $\mathcal{L}_1$ adaptive controller in maintaining system consistency and repeatability has much broader applications. The $\mathcal{L}_1$ adaptive controller can therefore also be extended to formation flight between UAVs, where any unmodelled interactions between vehicles are compensated for. All experimental results shown in Section 7 were conducted in the UTIAS indoor robotics lab. From these results, outdoor performance of a quadrotor operating with the $\mathcal{L}_1$ adaptive controller is expected to outperform typical PID or PD controllers in the presence of disturbances from wind or interactions between other surroundings.

Finally, the results of the $\mathcal{L}_1$ and ILC combined control architecture in Section 7.4 show that even under changing dynamics or in the presence of disturbances, the initial learned trajectory requires no further learning to maintain high performance after the change in dynamics. This suggests that learned trajectories can be shared among dynamically different systems, if the underlying controllers on these systems are $\mathcal{L}_1$ adaptive controllers that share the same reference system. This means that the learning process, which can be time-consuming and costly, can be avoided for additional systems required to perform the same task.

Bibliography

[1] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[2] M. D. Powers, C. and V. Kumar, "Quadrotor kinematics and dynamics," *Handbook of Unmanned Aerial Vehicles*, vol. 42, no. 5, pp. 307 – 328, 2014.

[3] T. Group, "World unmanned aerial vehicle systems, market profile and forecast," 2013.

[4] G. J. Harrison, "Unmanned aircraft systems (UAS): Manufacturing trends - report r42938," Tech. Rep., 2013.

[5] D. Jenkins and B. Vasigh, "The economic impact of unmanned aircraft systems integration in the United States," Association for Unmanned Vehicle Systems International, Tech. Rep., 2013.

[6] "Aeryon SkyRanger," http://www.aeryon.com/products/avs/aeryon-skyranger.html, accessed: 2014-11-30.

[7] I. Palunko and R. Fierro, "Adaptive control of a quadrotor with dynamic changes in the center of gravity," in *the Proc. of the 18th IFAC World Congress*, vol. 18, no. 1, 2011, pp. 2626–2631.

[8] G. Antonelli, F. Arrichiello, S. Chiaverini, and P. R. Giordano, "Adaptive trajectory tracking for quadrotor mavs in presence of parameter uncertainties and external disturbances," in *the Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2013, pp. 1337–1342.

[9] G. Antonelli, E. Cataldi, P. R. Giordano, S. Chiaverini, and A. Franchi, "Experimental validation of a new adaptive control scheme for quadrotors mavs," in *the*

Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 2439–2444.

[10] M. I. Rashid and S. Akhtar, "Adaptive control of a quadrotor with unknown model parameters," in *the Proc. of the 9th IEEE International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2012, pp. 8–14.

[11] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *the Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 2668–2673.

[12] M. Achtelik, T. Bierling, J. Wang, L. Höcht, and F. Holzapfel, "Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties," in *AIAA Infotech@ Aerospace*, 2011, pp. 1–15.

[13] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a uav with a robotic arm," in *the Proc. of the 10th International IFAC Symposium on Robot Control*, 2012, pp. 704–709.

[14] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Adaptive control for UAVs equipped with a robotic arm," in *the Proc. of the 19th IFAC World Congress*, 2014, pp. 11 049–11 054.

[15] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Lyapunov based model reference adaptive control for aerial manipulation," in *the Proc. of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 966–973.

[16] N. M. Wang, C. and M. Trentini, "Controller development and validation for a small quadrotor with compensation for model variation," in *the Proc. of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 902–909.

[17] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[18] R. Schreurs, S. Weiland, H. Tao, Q. Zhang, J. Zhu, Y. Zhu, and C. Xu, "Open loop system identification for a quadrotor helicopter system," in *the Proc. of the 10th IEEE International Conference on Control and Automation (ICCA)*, 2013, pp. 1702–1707.

[19] D. H. Shim, H. J. Kim, and S. Sastry, "Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification," in *the Proc. of the IEEE International Conference on Control Applications*, 2000, pp. 808–813.

[20] J. Niño, F. Mitrache, P. Cosyn, and R. De Keyser, "Model identification of a micro air vehicle," *Journal of Bionic Engineering*, vol. 4, no. 4, pp. 227–236, 2007.

[21] J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

[22] A. Mokhtari and M. Benallegue, "Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle," in *the Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 2359–2366.

[23] R. Lopez, S. Salazar, I. Gonzalez-Hernandez, and R. Lozano, "Real-time parameters identification for a quad-rotor mini-aircraft using adaptive control," in *the Proc. of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 499–505.

[24] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *the Proc. of the 45th IEEE International Conference on Decision and Control*, 2006, pp. 4957–4962.

[25] B.-C. Min, J.-H. Hong, and E. T. Matson, "Adaptive robust control (arc) for an altitude control of a quadrotor type uav carrying an unknown payloads," in *the Proc. of the 11th IEEE International Conference on Control, Automation and Systems (ICCAS)*, 2011, pp. 1147–1151.

[26] I. M. Gregory, E. Xargay, C. Cao, and N. Hovakimyan, "Flight test of an $\mathcal{L}_1$ adaptive controller on the nasa airstar flight test vehicle," in *the Proc. AIAA Guidance, Navigation and Control*, Toronto, Canada, 2010.

[27] K. A. Ackerman, E. Xargay, R. Choe, and N. Hovakimyan, "$\mathcal{L}_1$ stability augmentation system for calspanś variable-stability learjet," in *the Proc. AIAA Guidance, Navigation and Control*, San Diego, CA, 2016.

[28] G. Norris, "Adaptive safety system flies on variable stability f-16," *Aviation Week & Space Technology*, September 2016.

[29] Z. Zuo and P. Ru, "Augmented $\mathcal{L}_1$ adaptive tracking control of quad-rotor unmanned aircrafts," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 3090–3101, 2014.

[30] P. De Monte and B. Lohmann, "Position trajectory tracking of a quadrotor helicopter based on $\mathcal{L}_1$ adaptive control," in *the Proc. of the European Control Conference*, 2013, pp. 3346–3353.

[31] B. Michini and J. P. How, "$\mathcal{L}_1$ adaptive control for indoor autonomous vehicles: Design process and flight testing," in *the Proc. AIAA Guidance, Navigation and Control*, Chicago, IL, 2009.

[32] S. Mallikarjunan, B. Nesbit, E. Kharisov, E. Xargay, N. Hovakimyan, and C. Cao, "$\mathcal{L}_1$ adaptive controller for attitude control of multirotors," in *the Proc. AIAA Guidance, Navigation and Control*, Minneapolis, MN, 2012.

[33] M. Q. Huynh, W. Zhao, and L. Xie, "$\mathcal{L}_1$ adaptive control for quadcopter: Design and implementation," in *the Proc. of the 13th International Conference on Control, Automation, Robotics & Vision (ICARCV)*, 2014, pp. 1496–1501.

[34] C. Nicol, C. Macnab, and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics*, vol. 21, no. 6, pp. 927–938, 2011.

[35] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control," *Autonomous Robots*, vol. 33, no. 1-2, pp. 129–142, 2012.

[36] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *the Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2491–2498.

[37] A. E. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *the Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4916–4921.

[38] A. H. J. De Ruiter, C. J. Damaren, and J. R. Forbes, *Spacecraft Dynamics and Control: An Introduction*. Chichester, West Sussex: John Wiley & Sons, 2013.

[39] G. M. T. D'Eleuterio and G. R. Heppler, *Newton's Second Law and All That: A Treatment of Classical Mechanics*. Toronto: University of Toronto, 2011.

[40] J. Kelly and G. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[41] N. Hovakimyan and C. Cao, $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.

[42] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control Analysis and Design*, 2nd ed. Chichester, West Sussex: John Wiley & Sons, 2005.

[43] N. Liu, Z. Cai, and Y. Wang, "An $\mathcal{L}_1$ adaptive roll and pitch angle controller for quadrotors," in *the Proc. of the 28th Chinese Control and Decision Conference (CCDC)*, 2016, pp. 2473–2478.

[44] K. Pereida, R. R. P. R. Duivenvoorden, and A. P. Schoellig, "High-precision trajectory tracking in changing environments through $\mathcal{L}_1$ adaptive feedback and iterative learning," in *Submitted*, 2016.

[45] F. Bergenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *the Proc. of the International Conference on Robotics and Automation (ICRA)*, 2016, pp. 491–496.

Appendix A

Nonlinear Observability Analysis Script

The observability analysis of the system defined by (5.8) and (5.7) is analyzed using the following script. After defining the system, it incrementally constructs the observability matrix and calculates its rank. Since the higher order Lie derivatives are calculated recursively, the rank of the incomplete observability matrix is calculated after each set of Lie derivative orders, which reduces computational time. Once full rank has been concluded, no further computations are necessary, since subsequent rows have no effect on the full rank.

```
clear variables
close all

disp('new run')

% define symbolic real variables
syms g L gamma real
syms x y z xdot ydot zdot phi theta psi p q r m x_CG y_CG z_CG Jxx Jxy Jxz Jyy Jyz Jzz real

% create a row vector of variables to check observability of
vars_to_check = [m x_CG y_CG z_CG Jxx Jxy Jxz Jyy Jyz Jzz];

% create inverse inertia matrix
J = [Jxx, Jxy, Jxz; Jxy, Jyy, Jyz; Jxz, Jyz, Jzz];

% Define inertial position and attitude (Euler angles)
r_b_I = [x; y; z];
Theta = [phi; theta; psi];

% Create CG offset vector and its cross matrix
r_off = [x_CG; y_CG; z_CG];
r_off_cross = cross_matrix(r_off);

% Create translational and angular velocity
```

```
v_BI_B = [xdot; ydot; zdot];
om_BI_B = [p; q; r];

% Create angular velocity cross matrix
om_cross = cross_matrix(om_BI_B);

% Construct twain vector expressed in body-fixed frame with O_B
nu = [v_BI_B; om_BI_B];

% Create cross matrix
nu_cross = [om_cross cross_matrix(v_BI_B); zeros(3) om_cross];

% Create Mass operator
M = [m*eye(3) -m*r_off_cross; m*r_off_cross J];

% Define control-affine matrices
g1 = [0 0 1 0 -L gamma]';
g2 = [0 0 1 L 0 -gamma]';
g3 = [0 0 1 0 -L gamma]';
g4 = [0 0 1 -L 0 -gamma]';

%%

% Rotation matrix from inertial to body frame = C_BI
C_IB = [cos(theta)*cos(psi), cos(theta)*sin(psi), -sin(theta);
        sin(phi)*sin(theta)*cos(psi)-cos(phi)*sin(psi),
        sin(phi)*sin(theta)*sin(psi)+cos(phi)*cos(psi),
        sin(phi)*cos(theta);
        cos(phi)*sin(theta)*cos(psi)+sin(phi)*sin(psi),
        cos(phi)*sin(theta)*sin(psi)-sin(phi)*cos(psi),
        cos(phi)*cos(theta)];

% Inverse is its transpose since the rotation matrix is orthogonal
C_BI = C_IB';

% Define Inverse Euler Angle Rates matrix
S_inv = [1 sin(phi)*tan(theta) cos(phi)*tan(theta);
    0 cos(phi) -sin(phi);
    0 sin(phi)*sec(theta) cos(phi)*sec(theta)];

% Construct external force-torque from gravity
F_g_B = C_BI*[0; 0; -m*g];
f_g = [F_g_B; r_off_cross*F_g_B];

% Define the augmented state equation, f(x,u)
f0 = M\nu_cross*M*nu + M\f_g;
f0 = [C_IB*v_BI_B; S_inv*om_BI_B; f0; zeros(length(vars_to_check),1)];

f1 = [zeros(6,1); M\g1; zeros(length(vars_to_check),1)];
f2 = [zeros(6,1); M\g2; zeros(length(vars_to_check),1)];
f3 = [zeros(6,1); M\g3; zeros(length(vars_to_check),1)];
f4 = [zeros(6,1); M\g4; zeros(length(vars_to_check),1)];

% Define output equation, y = h(x)
h = [r_b_I; Theta; nu];
```

```
% Create row vector of all state variable (including augmented variables)
% to take gradient with respect to
v = [h', vars_to_check];

% Construct a matrix that allows Lie dervivatives to be iterated on f_i's
F = [f0 f1 f2 f3 f4];

% Initialize the observability matrix
O = sym(zeros(12,22));

% Initialize row counter of the observability matrix that allows indexing
% of additional rows
row_counter = 0;

% Construct observability matrix row by row, excluding any zero rows
for order = 0:length(v)-1
    % Loop through each output
    for output_num = 1:length(h)
        % Loop through f_i's
        for f = 1:size(F,2)
            fprintf('Calculating L_f%d^%d h(%d): ',f-1,order,output_num)
            observability_row = gradient(LieDerivative(h(output_num),F(:,f),v,order),v)';

            % Check if row is zero
            zero_flag = 1;
            for element = 1:length(observability_row)
                if observability_row(element) ~= 0
                    zero_flag = 0;
                    break
                end
            end

            % If the zero flag did not remain zero, add the row to the
            % observability matrix
            if not(zero_flag)
                fprintf('non-zero, adding to O\n')
                row_counter = row_counter + 1;
                O(row_counter,:) = observability_row;
            else
                fprintf('zero\n')
            end
        end
    end
    % Break the loop if full rank has been reached
    rank_obs = double(rank(O));
    if rank_obs == length(v)
        break
    end
end

fprintf('Number of non-zero rows in the observability matrix: %d\n\n', size(O,1))

fprintf('Length of augmented state %d\n', length(v))
fprintf('Rank of observability matrix %d \n\n', rank_obs)
```

The *LieDerivative* function is defined as follows:

```
function result = LieDerivative(h,f,v,order)
%LIEDERIVATIVE calculates the Lie derivative of h wrt f of order order with
%the gradient computed using elements in v

if order == 0
    result = h;
    return
end

result = gradient(LieDerivative(h,f,v,order-1),v)'*f;

end
```

The *cross_matrix* function is defined as follows:

```
function vect_cross = cross_matrix(vect)
%CROSS calculates cross matrix from 3 by 1 or 1 by 3 vector

if length(vect)~=3
    vect_cross = zeros(3);
    return
end

vect_cross = [0, -vect(3), vect(2); vect(3), 0, -vect(1); -vect(2), vect(1), 0];
end
```