

Kristian Klausen

# Coordinated Control of Multirotors for Suspended Load Transportation and Fixed-Wing Net Recovery

Thesis for the degree of Philosophiae Doctor  
Trondheim, June 2017

**Norwegian University of Science and Technology**  
Faculty of Information Technology and Electrical Engineering  
Department of Engineering Cybernetics



Norwegian University of  
Science and Technology

**NTNU**

Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering  
Department of Engineering Cybernetics

© Kristian Klausen

ISBN 978-82-326-2416-4 (printed version)  
ISBN 978-82-326-2417-1 (electronic version)  
ISSN 1503-8181  
ITK Report 2017-12-W

Doctoral theses at NTNU, 2017:172

Printed by Skipnes Kommunikasjon as

*in memory of Mr. Orange*



# Summary

As the core technology behind Unmanned Aerial Vehicles (UAVs) have matured, there have been a surge of increased interest for these vehicles over the past years. UAVs have the potential to become a ubiquitous tool for surveillance, remote sensing, inspection, asset and sea monitoring, search and rescue and many other areas. Different types of UAVs are used; fixed-wing UAVs look like an aircraft, and have long range and high endurance, usually measured in hours. They do, however, usually require special equipment and areas to perform takeoff and landing. Unlike fixed-wing UAVs, multirotor UAVs can produce lift straight upwards, and are thus able to hover in-air, and takeoff and land practically anywhere. Nevertheless, as multirotors must produce all lift from its motors and propellers, they have severely less range, with a typical flight-time of 30 minutes.

Multirotor UAVs are mostly used for short-range operations which benefit from their ability to hover. Typical use-cases are inspection, aerial photography and situational overview and surveillance of smaller areas. These are all examples of using the UAV as a remote sensing platform, where the on-board control system ensures the integrity of the UAV alone.

This monograph is motivated by the desire to enable multirotor UAVs to interact with the environment in the form of a suspended payload. This is useful for package delivery applications, sensor placement in remote areas such as on icebergs, and autonomous mine-localization.

Both single and multiple UAVs cooperatively transporting the same load for increased endurance are considered.

In addition, a novel recovery concept for fixed-wing UAVs are presented. By suspending a net below two cooperating multirotors, fixed-wing UAV operations can be performed in situations where landing was previously a challenge, such as ship-based operations and confined rural or forested areas.

Chapter 2 gives an overview of the platform developed for the experimental trials conducted in this work. The chapter gives an introduction to the various hardware components and software tools used. The multirotor platform is designed for outdoor usage, and is a hexacopter type platform weighing 2.2 kg, equipped with an autopilot, an on-board computer, precise satellite navigation systems and sensors specifically designed for payload transportation.

Chapter 3 considers a single multirotor carrying a suspended load. After giving an overview of related work, dynamic modeling of the interconnected multirotor-suspended load system is performed using Kane's equations. Based on this model, a nonlinear controller is designed, to guarantee trajectory tracking of the multirotor

UAV while being subjected to disturbances from the motion of the suspended load. To minimize the swing of the load, an open loop and a closed loop approach is considered, and combined to achieve robust swing reduction. The proposed controller is verified by numerical simulations and experimental trials.

Chapter 4 introduces multi-lift operations, which is the topic for the rest of the monograph. After an overview of the related work, an introduction to modeling and the complex dynamic coupling that occurs when multiple UAVs are transporting the same payload is given. It is illustrated how any motion of any of the UAVs has a direct influence on the motion of the other UAVs in the operation. Equations of motion are derived, which are used for simulation purposes for the rest of the thesis.

Chapter 5 presents a kinematic motion controller for multiple multirotors transporting a suspended load. By measuring the relative angles to the suspension point on each UAV, a distributed controller is designed which does not rely on relative position measurements. The controller relies on measuring the weight of the suspended load.

Chapter 6 solves the multi-agent control problem by synchronized path-following. A controller is designed on each UAV, which ensures local path-following along a pre-defined parameterized path. Further, the positions along that path is synchronized, ensuring stability of the desired relative formation. Numerical simulation based on the dynamic models presented in Chapter 4 validates the design.

Chapter 7 develops a position based cooperative controller to control the relative position of multiple multirotor UAVs transporting a suspended load. The disturbance from the suspended payload is considered unknown, and an adaptive term is designed to counteract the disturbance on the multirotors, as well as from wind. Numerical simulations and experimental trials verify the results.

Chapter 8 presents a novel method for recovery of a fixed-wing UAV in the absence of stationary nets or runways, by suspending a net between two synchronized autonomous multirotor UAVs. This method is especially suited for ship-based operations, where the landing pattern can be optimized with respect to wind direction and speed, and allows for fixed-wing UAV operations to be conducted on smaller vessels without the deck-capacity for large, stationary net. The proposed method is experimentally validated on a small-scale UAV platform.

Chapter 9 concludes the thesis and gives some remarks on future work.

# Preface

This monograph is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU).

The work presented herein has been completed at the Centre for Autonomous Marine Operations and Systems (NTNU AMOS), Department of Engineering Cybernetics, in the period of August 2013 to June 2017. My supervisors have been Professors Thor I. Fossen and Tor Arne Johansen.

## Acknowledgments

Founding was mainly provided by the Research Council of Norway through the Centres of Excellence funding scheme, Project number 223254.

The completion of this thesis and the work it represents have been one of the most challenging things I have ever done. Yet, it has been an incredible rewarding journey. I have had the fortunate opportunity to work with a group of amazingly talented people, on practical problems which aligns close with my regard for cybernetics and engineering.

To my supervisors, Thor I. Fossen and Tor Arne Johansen, your dedication and resourcefulness are the sole reason for why I am where I am today. Thanks for all the discussions, ideas and late night email responses, and I am looking forward to continue to work with you both on our future endeavors.

To Prof. António Pedro Aguiar, thank you for letting me visit your lab in Porto, and for the discussions we had leading up to our joint publication. To Prof. Murat Arcak and Chris Meissen, thank you for a fruitful cooperation on passivity-based control and for all you have taught me.

To my friends and office-mates, Frederik Stendahl Leira, Kim Lynge Sørensen and especially João Filipe Fortuna Araújo, thank you for making me look forward to going to the office every day.

To my colleagues at the department and our research centre, thank you for being who you are. We have an incredible social community, which I hope will continue to prosper. In addition, thanks for all the countless hours of discussions, feedback and support you have provided over the years. I could name so many, but I would like to thank Mikkel Cornelius Nielsen for being both a colleague and an awesome roommate for the past year, Kristoffer Rist Skøien for providing perspective when needed, and to Mikkel Eske Nørgaard Sørensen for being in my office almost as much as I have.

To our UAV pilots Lars Semb and Pål Kvaløy, the work you do is invaluable to the UAV-lab. The skills and ingenuity you possess and the dedication you show is truly inspiring.

To the incredible department administrative and technical staff, thank you for running this department so properly, efficiently and *friendly*.

I would also like to use this opportunity to thank my primary school teachers Ruth and Svein Grøndahl for awaking my interest in mathematics and natural sciences, and my lower secondary school teacher Arild Osestad for showing me the value of hard work.

To all my friends, especially Øyvind Lilletveit, Konstantin Nalum and Eivind Hope Sørbø, thank you for never giving up on me.

To my family; My parents Håkon and Marianne, and my siblings Andreas and Solveig, thank you for the everlasting encouragement, unconditional love and support.

*Kristian Klausen*  
Trondheim, 2017

# Contents

<b>Summary</b>	v
<b>Preface</b>	vii
<b>Contents</b>	ix
<b>1 Introduction</b>	1
1.1 Background . . . . .	1
1.2 Objectives and Main Contributions . . . . .	5
1.3 Publications . . . . .	6
1.4 Organization of the Thesis . . . . .	7
<b>2 System Description – Hardware and Software</b>	9
2.1 Introduction . . . . .	9
2.2 Multirotor Frame and Components . . . . .	10
2.3 Low-Level Autopilot . . . . .	11
2.4 Hardware Platform . . . . .	12
2.5 The LSTS Toolchain . . . . .	12
2.6 High-Precision Navigation using Real-Time Kinematic GNSS . . . . .	14
2.7 Communication and Other Sensors . . . . .	16
<b>3 Single Multirotor UAV with Suspended Load</b>	17
3.1 Introduction . . . . .	17
3.2 Modeling of Multirotor UAV Dynamics . . . . .	19
3.3 Control-Oriented Model for Suspended Load Transportation . . . . .	22
3.4 Problem Description and Control Strategies . . . . .	26
3.5 Nonlinear Trajectory Tracking . . . . .	26
3.6 Experimental Validation of the Controller . . . . .	34
3.7 Suppression of Suspended Load Swing . . . . .	35
3.8 Experimental Validation of Controller with Swing Damping . . . . .	41
3.9 Chapter Summary and Conclusions . . . . .	44
<b>4 Multibody Operations and Modeling</b>	45
4.1 Introduction . . . . .	45
4.2 Related Work . . . . .	46
4.3 Multi-Body Modeling . . . . .	48

## Contents

---

4.4	Equations of Motion for Multi-Lift Operations . . . . .	52
4.5	Chapter Summary . . . . .	58
<b>5</b>	<b>Distributed Kinematic Control</b>	<b>59</b>
5.1	Distributed Kinematic Control . . . . .	59
5.2	Simulation Study . . . . .	62
5.3	Chapter Summary and Conclusions . . . . .	65
<b>6</b>	<b>Cooperative Path Following</b>	<b>69</b>
6.1	Introduction . . . . .	69
6.2	Control Design . . . . .	73
6.3	Path generation . . . . .	77
6.4	Simulation Study . . . . .	79
6.5	Chapter Summary and Conclusions . . . . .	80
<b>7</b>	<b>Passivity-Based Cooperative Control</b>	<b>83</b>
7.1	Introduction . . . . .	83
7.2	System Description . . . . .	85
7.3	Design 1: Suspended Load Connected by Springs . . . . .	86
7.4	Simulation Study . . . . .	88
7.5	Design 2: Applying Formation Feed Forward . . . . .	92
7.6	Simulation Results . . . . .	98
7.7	Experimental Validation . . . . .	99
7.8	Chapter Summary and Conclusions . . . . .	102
<b>8</b>	<b>Net-Recovery of a Fixed-Wing UAV using Multirotors</b>	<b>103</b>
8.1	Introduction . . . . .	103
8.2	Autonomous Net Recovery . . . . .	105
8.3	Control Design . . . . .	108
8.4	Experimental Validation . . . . .	115
8.5	Results . . . . .	121
8.6	Chapter Summary and Conclusions . . . . .	124
<b>9</b>	<b>Conclusions</b>	<b>129</b>
<b>Appendices</b>		<b>133</b>
<b>A</b>	<b>Experimental Setup</b>	<b>135</b>
<b>B</b>	<b>Stability Analyses</b>	<b>137</b>
<b>References</b>		<b>141</b>

# Chapter 1

## Introduction

### 1.1 Background

Unmanned Aerial Vehicles is a field within both research and engineering with a rich history. Although previously most applications were aimed at military use, the advancements within satellite positioning systems, inertial sensor- and battery-technology has enabled a new breed of UAVs for civilian applications. The availability cheap *micro electromechanical systems* (MEMS) sensors, such as accelerometers and gyroscopes, with suitable specifications for UAV use has, among other things, been a key part of this development.

A note on nomenclature and naming convention; As is common on the scientific literature concerning autonomous flying vehicles, the term Unmanned Aerial Vehicle is used throughout this thesis. We generally distinguish between *fixed-wing* UAVs, *multirotor* UAVs and *helicopters*. Fixed-wing UAVs have airplane-like structures, with wings and other surfaces producing most of the lift. Thrust is generated with one or more propellers, driven by either electric motors or combustion engines. Multirotors, also sometimes called *multicopters*, have three or more constant-pitch propellers, which must be electronically stabilized to be flyable. A helicopter has a large rotor with varying pitch blades for stabilization. There also exists hybrid vehicles, such as a fixed-wing UAV with extra horizontal thrust making it capable of vertical takeoff, often referred to simply as a *Vertical Takeoff and Landing* (VTOL) UAV.

Due to their different construction, multirotors and fixed-wings inherently have different properties. Since fixed-wing UAVs use wings to generate most of the lift, they can be made highly efficient. This means that even with electrical power systems, such UAVs can have long range and high endurance, often several hours. For larger UAVs, combustion engines can provide power considerably longer, with many commercially available systems delivering 24 hours of operating time. They



**Figure 1.1:** A hexacopter type multirotor with a suspended load.

## 1. Introduction

---

do, however, require special facilities for takeoff and landing, such as runways, catapults, large stationary net or other recovery devices.

Multicopters are constructed with all their propellers pointing straight up, which enables them to hover in-air. It also enables them to takeoff and land virtually anywhere, and that makes them highly versatile. However, as all produced lift comes from its motors and propellers, they use substantially more energy to stay in the air. Hence, they have a rather shorter range, with a typical flight-time of up to 30 minutes. An example of a hexacopter, which is a multicopter with six propellers, is shown in Figure 1.1. In Figure 1.2, a Skywalker X8 fixed-wing UAV is shown. This UAV has a wingspan of approx. 2 m.

The word “drone” is an ambiguous term often used by the public, these days usually synonymous to small personal multicopter UAVs. Due to its vague definition, the word is not used in this thesis. It should also be noted that although the scientific and engineering community use the term UAV, many *Civil Aviation Authorities* (CAAs) prefer to use *Remotely Controlled Aircraft System* (RPAS) and/or *Unmanned Aerial System* (UAS) to distinguish between manned and unmanned aircraft systems.



**Figure 1.2:** A fixed-wing UAV in flight.

There is a long history of hobbyist usage of radio-controlled (RC) flying vehicles. Fixed-wing UAVs can be flown fully manually, by having the pilot control input directly affecting the angles of the control surfaces on the airplane, such as rudders, ailerons and elevators. Even though multicopters have a simple mechanical design, they *require* computerized control to be operated, even by skilled pilots. At the simplest level, a gyro is used to stabilize the angular velocity of the multicopter. By fusing data with an accelerometer, the craft can automatically

*level out*, and the first flight-controllers for multicopters are made.

In addition to commercially available flight-controllers, several *open-source* projects emerged, such as Ardupilot [9], Paparazzi [98] and more recently PX4 [108]. In the beginning they did little more than stabilize the multicopter, and was reserved for enthusiasts. Nonetheless, the capabilities the multicopters showed when used as an airborne *camera platform* drove the development forward, and opened up the market for the ready-to-fly multicopter UAVs with integrated camera systems we see available today, tailored for both recreational and professional use.

The features of most, if not all, multicopter systems available today are still built around using the multicopter as a camera platform. As the capabilities of the flight-controllers becomes more and more mature, they are now robustly able to control the attitude and position of the multicopter itself, along with any firmly attached (though often gimbal-stabilized) camera or sensor systems.

There are, however, several applications that would require the multicopter UAV to control not only itself, but also some form of payload suspended below it using a wire.

One substantial use-case for UAVs in the public is the transportation and delivery of packages. Multicopters serve as an excellent platform in many scenarios, due to their ability to hover over its final destination. This is especially true for urban

environments, where the somewhat limited flight-time does not pose a problem. Hybrid designs, such as a fixed-wing UAV equipped with vertical thruster to enable hover capabilities, could also help alleviate those limitations. However, during the hover-phase, the VTOL would behave just like a normal multirotor.

There are several commercial companies that already have announced package delivery with multirotors. Most notably Amazon [4], which are designing custom vehicles specifically tailored towards aerial package delivery, using a system to lower the package to the recipient on a wire. The same concept was used by Domino's, which used a tethered wire to lower a pizza to a hungry costumer [24].

Autonomous collection of *water samples* is another task for which multirotors are especially suited. As reported in [91], a hose can be freely hung under the multirotor, connected to a pump-and-sampling mechanism attached to the multirotor body. This allows for remote collection of samples without the need for specialized multirotors which can land on water, which typically increase weight and thus lower the flight-time and range of the system.

Even though these systems involve the use of a slung load, the accuracy they present is minimal at best. As long as the multirotor is able to hover over a desired location, the package is simply lowered without any considerations for the final position of the load once it hits the ground.

This might be more than adequate for delivering packages in consumer back-yards, but for industrial applications the position of the suspended load must be taken into account when designing the control system. When delivering packages to offshore vessels, it might not always be enough to simply drop the package on the ship deck [128].

In any practical application, the effects the suspended load have on the multirotor, along with control over the inescapable swing of load, must be addressed.

This is especially true for applications where the suspended load must be lowered at all times. In addition to delivery options where weight is saved by not needing a winch, the *purpose* of the flight can be collection data from a sensor at the end of the line.

When collecting data for geomatic surveys, a *magnetometer* is a crucial tool. Due to the variable magnetic fields induced by the multirotor motors, the magnetometer should be placed as far away as possible; at the end of a suspended line is very ideal. Figure 1.3 shows such an arrangement in operation from the Canadian company Gem Systems, which specializes in geomatic survey equipment. The multirotor needs to track survey patterns while still keeping the suspended load under control by not allowing excessive swings during maneuvering.

Similar sensors are used for autonomous detection of *land mines*. Either by radar or magnetic interference, the location of a land-mine can be deferred by hovering



**Figure 1.3:** A multirotor carrying an accurate survey-type magnetometer. *Image courtesy of gemsys.ca*

## 1. Introduction

---

the sensor close to the ground. These types of capabilities can save countless lives, and it has been estimated that usage of aerial vehicles for this task is up to 20 times faster and up to 200 times cheaper than conventional methods [133].



**Figure 1.4:** Three multirotors carrying a single payload.

In addition to control and suppress the unwanted swing of the suspended payload, it can often be advantageous to have even better control of the suspended payload than what can be achieved with a single suspension point. Instead of using multiple attachment points or complicated suspension wire setups, one can use *multiple* multirotors to lift the same payload. By using three or more, the suspended load is no longer able to swing, and can be safely transported. Multiple UAVs would also be able to share the load. One also gains operational flexibility, where the number of multirotors can be varied depending on the specific load to be transported. Finally, the operational redundancy is increased by having multiple smaller units which easily can be replaced in case of damage or malfunction.

An example of three multirotors lifting the same payload is illustrated in Figure 1.4.

By having the ability to have several multirotors carrying the same payload, new applications are also possible. Consider the ability to catch and recover another UAV by suspending a net between two multirotors, like illustrated in Figure 1.5.

Because of their long endurance and large area of coverage, using fixed-wing UAVs as a sensor platform is preferable in maritime operations from ships. While takeoff from a ship deck can be accomplished by conventional means, such as a catapult, landing and recovery is more challenging. Often, a large net is fixed to the ship deck, taking considerable space and adds operational complexity. Autonomous operations are also made difficult by the wave-induced motion of the ship, along with turbulence near the hull of the ship.

The capability of recovering UAVs in-air is also applicable for airspace control. As more and more multirotors and other UAVs fill the sky, there is an increased need to remove unwanted UAVs from restricted areas. Breach of restricted airspaces could either happen by accident of an out-of-control UAV, or intentionally for malicious reasons. The concept of catching rogue UAVs with nets have been straightforwardly tested in [132], where a net was attached under a multirotor UAV, which was manually controlled to catch smaller multirotor UAVs.

Table 1.1 summarizes the capabilities regarding slung load of commercially available systems on the market today. As can be seen, no commercial system for automatic transportation of suspended loads are available on the market today. Still, UAV slung systems have been thoroughly explored in the literature. Control aspects and modeling of a multirotor UAV with a suspended load is considered in [124], where controllers are developed on a 2D-plane to control the position of the suspended load, and verified on a small indoor platform. [19] discusses slung-load transportation

for autonomous helicopters, and verifies methods for swing damping experimentally on in a lab setting using a small helicopter UAV. [13] illustrates single- and multilift operations with helicopter UAVs in a outdoor setting, using a centralized control structure for the coordinated control part. Recovery mechanisms for fixed-wing UAVs are available on the market today, where the most prominent commercial system is the SkyHook from Insitu, with the ScanEagle UAV [56]. It is a proprietary system, where the UAV is specifically designed with a hook on the wing tip, which connects to a vertical line suspended from a crane.

A more detailed literature overview for single- and multilift operations are given in Chapters 3 and 4, respectively.



**Figure 1.5:** A fixed-wing UAV being recovered in a net.

## 1.2 Objectives and Main Contributions

The objective of this work is the design and implementation of control structures to enable single and multiple multirotors to lift a suspended load and autonomous net-recovery of a fixed-wing UAV. As mentioned briefly above, most of the literature concerns the use of small, indoor multirotors for their experiments, often aided by external camera systems for accurate positioning. This work aims to design controllers that are applicable and verifiable on larger multirotors, tested in an out-door setting with satellite navigation and local measurements.

The following constitutes the main contributions of this thesis:

- A dynamic model of a multirotor UAV carrying a suspended load.
- A robust control method with integral action, for a multirotor UAV transporting a suspended load in a swing-free manner.
- Experimental verification of the above controller.
- Three control designs for cooperative transportation of a suspended load with multiple multirotors, with different requirements for necessary available measured signals.
- Experimental verification of a passivity-based control design, where three multirotors cooperatively carry an unknown suspended load.

**Table 1.1:** Overview of multirotor capabilities available on the commercial market

Application\Level of Control	Manual	Automatic
Multirotor with light-weight slung load	Possible	Partly
Multirotor with heavy slung load	Hard	No
Coordinated Suspended Load	No	No
Recovery of other UAVs	No	No

## 1. Introduction

---

- Development, testing and experimental verification of a novel net-recovery method for fixed-wing UAVs, especially suited for UAV operations from ships.
- A patent for the net-recovery landing concept was filed on April 12, 2017.

### 1.3 Publications

This thesis work has resulted in the following international peer-reviewed publications:

- J1** [67] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control with Swing Damping of a Multirotor UAV with Suspended Load. *Journal of Intelligent & Robotic Systems*, 2017. ISSN 1573-0409. doi: 10.1007/s10846-017-0509-6
- J2** [68] K. Klausen, T. I. Fossen, and T. A. Johansen. Autonomous Recovery of a Fixed-Wing UAV Using a Net Suspended by Two Multirotor UAVs. *Journal of Field Robotics (submitted)*, 2017
- J3** [69] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen. Cooperative Control for Multirotors Transporting an Unknown Suspended Load with Environmental Disturbances. *IEEE Transactions on Control Systems Technology (submitted)*, 2017
- C1** [63] K. Klausen, T. I. Fossen, and T. A. Johansen. Suspended Load Motion Control using Multicopters. In *Proceedings of the Mediterranean Conference of Control and Automation (MED)*, pages 1371–1376, Palermo, 2014. IEEE. ISBN 9781479958993. doi: 10.1109/MED.2014.6961567
- C2** [64] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control of a Multirotor UAV with Suspended Load. In *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 176 – 184, Denver, CO, 2015. IEEE. doi: 10.1109/ICUAS.2015.7152289
- C3** [65] K. Klausen, T. I. Fossen, T. A. Johansen, and A. P. Aguiar. Cooperative path-following for multirotor UAVs with a suspended payload. *Proceedings of the 2015 IEEE Conference on Control Applications (CCA)*, pages 1354–1360, 2015. doi: 10.1109/CCA.2015.7320800
- C4** [66] K. Klausen, J. B. Moe, J. C. Van Den Hoorn, A. Gomola, T. I. Fossen, and T. A. Johansen. Coordinated Control Concept for Recovery of a Fixed-Wing UAV on a Ship using a Net Carried by Multirotor UAVs. In *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 964–973, 2016. ISBN 9781467393331. doi: 10.1109/ICUAS.2016.7502640
- C5** [84] C. Meissen, K. Klausen, M. Arcak, T. I. Fossen, and A. Packard. Passivity-based Formation Control for UAVs with a Suspended Load. In *Proceedings of the IFAC World Congress*, 2017
- P1** Recovery System for UAV (*Patent filed on April 12, 2017*)

Other published work not a part of this thesis:

- [126] M. Stokkeland, K. Klausen, and T. A. Johansen. Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 998–1007, 2015. doi: 10.1109/ICUAS.2015.7152389
- [141] A. Zolich, T. A. Johansen, K. Cisek, and K. Klausen. Unmanned Aerial System Architecture for Maritime Missions. Design & Hardware Description. In *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015

## 1.4 Organization of the Thesis

With the exception of Chapter 2 and 4, this thesis is a collection of papers, which makes the chapters mostly self-contained. The thesis is organized as follows.

**Chapter 2** Introduces the software and hardware platform used in this thesis. Based on commercially available components, a versatile hexacopter type multirotor is constructed. The UAV utilizes the open-source autopilot *Pixhawk* as a low-level controller, accompanied by the APM:Copter flight stack. In addition, an accompanying linux computer implements all the high-level control algorithms described in this thesis. The chapter also describes the various sensors and communication link necessary for the experimental validations.

**Chapter 3** (Based on **C2, J1**) In this chapter, the problem of trajectory tracking of a multirotor UAV carrying a suspended payload is considered. The movement of the suspended payload influences the dynamics of the multirotor, which must be appropriately handled by the controller to achieve satisfactory tracking results. A mathematical model of the interconnected multi-body system is derived using Kane’s equations, and a nonlinear tracking controller based on the backstepping technique is developed. In addition to suppressing the effects of the swinging payload, the controller also compensates for an unknown constant wind disturbance. The origin of the tracking error is proven UGAS (Uniformly Globally Asymptotically Stable) and ULES (Uniformly Locally Exponentially Stable) through Lyapunov analysis. To reduce the swing motion of the suspended load, a nominal swing-free path is generated through open loop shaping filters, then further perturbed through a delayed feedback approach from measured load deflection angles to achieve robustness. The proposed controller structure is verified by simulations and experiments.

**Chapter 4** gives an introduction to multibody operations, which is the topic for Chapters 5–7. After an overview of the related work, the chapter also develops equations of motion of the interconnected multirotors-slung load system using the Udwadia-Kalaba-equations.

**Chapter 5** (Based on **C1**) This chapter presents a distributed kinematic control law for group coordination for several multirotors and a payload suspended with wires from each multirotor. Using relative measurements of the suspended load wire angle only, velocity controllers for the multirotors are developed to realize

## *1. Introduction*

---

the desired motion set by the kinematic controller. This results in a system where the group of multirotors are able to maneuver the payload to a desired position while keeping a desired formation. The results are verified by simulations.

**Chapter 6** (Based on **C3**) This chapter considers the coordinated transportation of a suspended load with multirotor UAVs as a coordinated path-following problem. A robust path-following controller is developed for each vehicle in the system, and coordination is achieved through synchronization of the position along a parameterized path. Stability of the interconnected system consisting of a network of local path-following controllers are analyzed, and the proposed control design is verified through numerical simulations.

**Chapter 7** (Based on **C5, J3**) In this chapter, a passivity based approach is utilized to achieve convergence to a desired relative position in the presence of an unknown disturbance. This lets a team of multirotors cooperatively transport a suspended payload of unknown weight. Two designs are presented, that differ in how the effect of the unknown load on the multirotor controllers are analyzed. Both designs are verified by numerical simulations. Further, an experimental trial of the second design is presented, where three autonomous multirotor UAVs successfully transport an unknown load.

**Chapter 8** (Based on **C4, J2**) This chapter presents a novel recovery method for fixed-wing Unmanned Aerial Vehicles (UAVs), aimed at enabling operations from marine vessels. Instead of using the conventional method of with a fixed net on the ship deck, we propose to suspend a net under two cooperative multirotor UAVs. While keeping their relative formation, the multirotor UAVs are able to intercept the incoming fixed-wing UAV along a virtual runway over the sea, and transport it back to the ship. In addition to discussing the concept and design a control system, this paper also presents experimental validation of the proposed concept for a small-scale UAV platform.

**Chapter 9** concludes the thesis and adds some final remarks on future work.

## Chapter 2

# System Description – Hardware and Software

### 2.1 Introduction

This chapter will give an overview of the hardware and software tools used to conduct the experiments presented in Chapters 3, 7 and 8. The design is similar to what we presented in [141].

The experimental platform consists of a combination of off-the-shelf and custom components and software.

#### 2.1.1 Objectives

When designing a testbed for complex multirotor operations, there are many considerations to take. A main motivation for the design goes back to the philosophy of the UAV-lab at the university<sup>1</sup>, which is the desire to develop *applications* to the core technology that already exists. Thus, when possible, we try to leverage low-cost off-the-shelf components when applicable to avoid spending unnecessary time.

Second, the design is built around the desire to do experimental validations on a platform flying *outdoors*. This is in contrast to a lot of multirotor research in the literature, for which only indoor experiments are conducted. When operating outdoors, there are some additional design requirements that needs to be addressed:

- a) *Local aviation regulation*: In Norway, all research conducted with UAVs are considered non-recreational flights, and must be performed according to the same rules that applies for commercial UAV activities [78].
- b) *Operational Aspects*: The multirotor control software must facilitate all aspects of the operation, including takeoff and landing, emergency procedures and the ability to recover manual control.
- c) *Environmental Disturbances*: All control designs should be able to counteract effects from wind.

---

<sup>1</sup><http://www.itk.ntnu.no/english/lab/unmanned>

The concept of *airworthiness* is important in relation to a). Even though the current legislations it is not fully defined on this topic, it requires the operator to be able to justify the choice of avionic systems. That is why we rely on off-the-shelf autopilots to do low-level control. This choice also ties into b). When developing new control systems, it is important that the system have some tolerance for errors. This could be an implementation error, wrong parameter tuning, etc. It is then beneficial that a pilot always can recover the UAV in a manual control mode run by the low-level autopilot for operations within line-of-sight. Further, using an off-the-shelf autopilot systems allows us to leverage the continuous development of said autopilots, and provides a familiar interface to the pilots. Finally, as some wind is inevitable when doing outdoor experiments, c) needs to be taken into account when designing control algorithms.

## 2.2 Multirotor Frame and Components

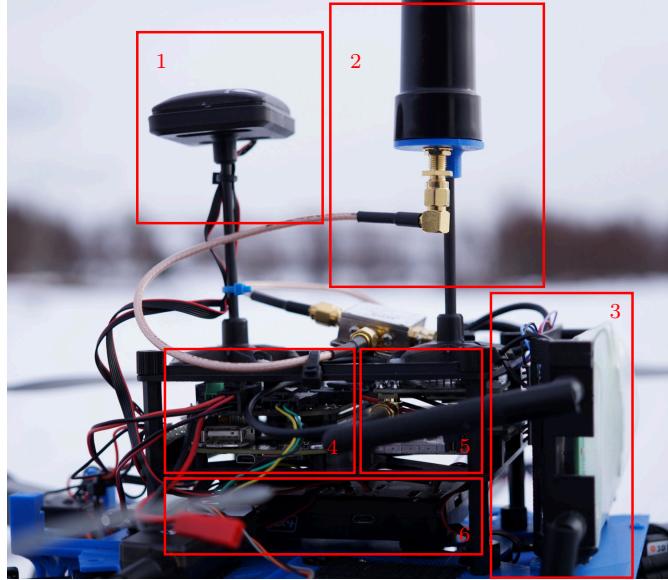
A *hexacopter*-type multirotor (with six motors) was chosen as a main platform, as it provides a bit more lifting capacity than a regular quadcopter, but is not as mechanically big and complex as a larger octocopter. The platform consists of a core of carbon-fibre plates, and the motors are attached on aluminum arms. In addition, six Electronic Speed Controllers (ESCs) control the speed of the motors, given reference values from an autopilot. The power-system consists of a three-cell Lithium Polymer (LiPo) battery with a nominal voltage of 11.1 V, with 5200 mAh capacity, which gives a flight-time of approx. 15 minutes. The standard, unaltered frame can be seen in Figure 2.1a.



(a) Basic hexacopter. *Image courtesy of 3dr. com* (b) The Pixhawk autopilot. *Image courtesy of pixhawk. org*

**Figure 2.1:** Illustration of a hexacopter (a) and the Pixhawk autopilot (b).

To facilitate the components necessary for the experimental trials conducted in this thesis, custom designed avionic stacks was 3D-printed in [113]. Continued development led to the final design, which can be seen in Figure 2.2, where the numbered boxes are described below.



**Figure 2.2:** Overview of the avionic-stack used on the experimental platform. (1) Autopilot GPS+Compass, (2) RTK GNSS Antenna, (3) Ubiquity Rocket M5 data link, (4) Beaglebone Black on-board computer, (5) RTK GNSS Receiver, (6) Pixhawk autopilot.

1. Code-based GPS (u-blox NEO6 w/ 3-axis magnetometer)
2. Active Helical L1 antenna for usage with RTK GPS
3. Ubiquity Rocket M5 5.8 GHz AirMax radio system
4. Beaglebone Black w/ custom adapter board for power conversion and communication
5. GNSS Receiver w/ timing (raw) output for RTK GNSS calculations by RTKLIB (u-blox LEA-M8T)
6. Pixhawk autopilot w/ APM:Copter flight stack

### 2.3 Low-Level Autopilot

The Pixhawk [107] hardware with Ardupilot<sup>2</sup>[9] software stack was chosen as the low-level autopilot. The hardware platform is well known, and is widely used for both recreational and professional use. The open-source nature of Ardupilot meant that alterations to the code could be made to facilitate new functionality, and the vast community surrounding the project can be leveraged. Relative to the alternatives, such as the PX4 [108] flight stack, the ardupilot software is in some ways more mature, and was already familiar when the project started in 2013.

---

<sup>2</sup>Ardupilot includes software specific for both multirotors and fixedwings, called APM:Copter (also called ArduCopter) and APM:Plane (also called ArduPlane), respectively.

However, since then, the first non-beta version of the PX4 stack has been released, which features a more modular code suitable further development. Still, Ardupilot is in active development, and for new projects the choice between the two often comes down to licensing terms (GPLv3 vs BSD). To be able to keep up with the regular updates of the software platform, only minor code-changes was introduced to the autopilot software, mainly to enable new ways to control the desired behavior of the multirotor using outputs from an on-board computer, which will run most of the custom-made control software.

An image of the Pixhawk autopilot can be seen in Figure 2.1b.

### 2.4 Hardware Platform

Regarding the on-board computer, a Beaglebone Black was used. This is a low-cost, System-On-a-Chip credit-card sized computer, leveraging a 1 GHz ARM Cortex A8 processor, designed for usage with embedded systems due to its many connection options [11]. A custom *cape* was designed, which routes signals from easy to use connectors to the Beaglebone. The cape provides power to the computer through a 1.5 A switching regulator, and features an on-board *real-time clock* w/ a coin-cell battery to keep the time between power cycles.

The platform is designed with two separate power sources. The larger battery battery described in Section 2.2, powers the motors and autopilot, while a smaller battery powers the on-board computer, sensors and radio. This way, the large battery can be changed between each flight, without interrupting the power to the on-board computer. As stated above, the typical flight-time is about 15 minutes, while the smaller battery provides power to the other electronics for several hours.

### 2.5 The LSTS Toolchain

The LSTS toolchain [106] is a software framework specifically designed for multi-vehicle, heterogeneous operations. The toolchain is developed by the Underwater Systems and Technology Laboratory (LSTS) at the University of Porto, Portugal. Originally designed for underwater vehicles, the software is now used by researchers and commercial companies to facilitate advanced operations with surface vessels, UAVs and underwater vehicles.

The main alternative to the LSTS toolchain is the Robot Operating System (ROS) [114]. Although ROS has a larger user-base and available plugins, the LSTS toolchain is, from a technical perspective, better suited for multi-vehicle operations. In addition, the LSTS toolchain has been extensively used for UAV operations by partners in Portugal.

The toolchain has the following key components, which is described in the following sections:

- *DUNE*: Software developing framework
- *IMC*: Communication protocol
- *Neptus*: Ground Control Station
- *GLUED*: A minimal Linux distribution

### 2.5.1 DUNE

DUNE: Unified Navigation Environment<sup>3</sup> [105] is a software framework written in C++. It allows the user the build *tasks*, which are separate entities executing a specific purpose. For example, one task could be reading a sensor value, while another task uses that sensor value as input and generates a motor control output value. The tasks communicate over the message protocol IMC.

All tasks in DUNE is compiled together to a single binary. This ensures fast execution, and enables the tasks to communicate via *shared memory* with minimal delay, rather than operating-system level communication sockets. A configuration file is used to describe, for each vehicle, which tasks should be initiated at startup.

DUNE is open-source, and the code can be found online<sup>4</sup>.

The core of DUNE has tasks that keeps track of the state of the vehicle, and functionality for logging, transmission of data to a ground station (Neptus), and drivers for key components such as the Pixhawk autopilot. Communication with the Pixhawk is performed over a serial link using the MAVlink [82] protocol.

DUNE uses the notion of *plans* to perform operations. A plan consists of several *maneuvers*, which can be waypoints, circular loiters or more complex lawn-mower scan patterns. For each type of maneuver, certain controllers are activated based on settings from operator. This leads to a *layered* control structure, where one controller can relay commands to a lower-level controller.

For example, consider the multirotor UAV performing a waypoint route. The current waypoint is sent to a *guidance controller*, which computes a desired control output. This is sent to the ardupilot driver, which translates the command to a MAVlink package, and transmits the message to the pixhawk. Finally, the pixhawk executes the output by appropriately controlling the motors.

The core functionality is expanded with new tasks to implement the controllers described in this thesis. Further, a number of drivers, utility functions, monitors and failsafes have been added.

### 2.5.2 IMC

The Inter Module Communication protocol (IMC) is a message-oriented protocol, which serves as the communication protocol both between each task in a DUNE instance, but also between vehicles and ground stations [81].

IMC defines a set of core messages, which is used by the system to relay important information to the ground control station such as current health status, battery capacity, as well as system data such as position and attitude. In addition, new messages can be added by a developer for specific needs, such as configuration of formation controllers between vehicles or new sensor data.

### 2.5.3 Neptus

Neptus is the Command and Control software in the LSTS toolchain [104]. It is written in Java, and runs on both Windows and Unix-based ground stations

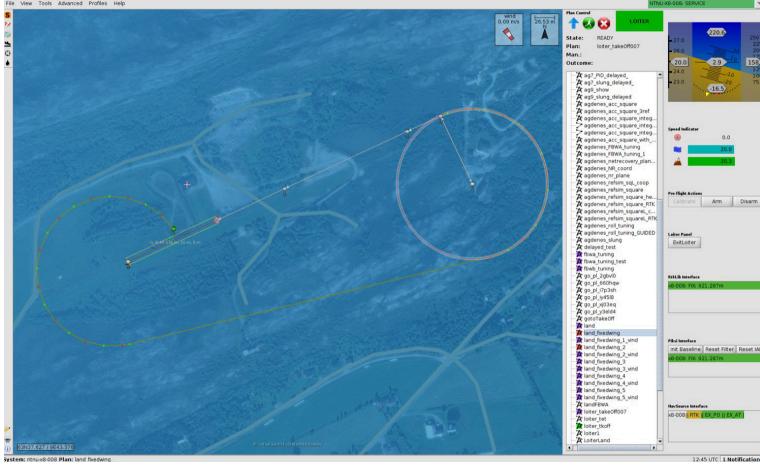
---

<sup>3</sup>DUNE is a *recursive acronym*

<sup>4</sup>[github.com/LSTS/dune](https://github.com/LSTS/dune)

computers. It allows the operator both plan, execute and review a mission through its modularized interface. Parameters and flight data can be viewed in real-time, and the operator get a clear overview of all vehicles involved in an operation on the same screen.

It communicates with vehicles running DUNE over IMC, and has a generalized interface for changing settings and parameters for every task that is currently executing. This allows for easy tuning of control parameters or settings, without the need to develop custom interfaces. An example of a Neptus console is given in Figure 2.3.



**Figure 2.3:** The Neptus console.

### 2.5.4 GLUED

GNU/Linux Uniform Environment Distribution (GLUED) is a light-weight linux-distribution aimed at embedded computers [79]. It is minimal in the sense that only packages necessary for operations are included on the target, and the package manager allows easy configuration depending on need. It allows generation of specific distribution configurations to a particular vehicle, whose configuration include network settings such as hostname and network address, as well as specific services to run at startup.

This means that the configurations for each vehicle can be completely scripted, and thus maintaining a fleet of vehicles is easy. GLUED also supports upgrading the operating system itself over a network connection.

## 2.6 High-Precision Navigation using Real-Time Kinematic GNSS

The autopilot internally fuses data from a MEMS-based Inertial Measurement Unit (IMU) with a magnetometer and Global Navigation Satellite Systems (GNSS) to

provide a full state attitude and position reference solution using an Extended Kalman Filter. However, the position acquired from traditional GNSS is without the required accuracy to do precision landing and formation flight [122]. Nonetheless, by using real-time kinematic (RTK) techniques with a differential correction from a base station, centimeter-level real-time positioning can be achieved due to the short signal wavelength (19 cm for GPS L1) of GNSS signals. This requires that integer carrier phase ambiguities are successfully resolved [122]. RTKLIB [129] is an open-source library for computing these ambiguities, and providing real-time position updates using raw data from a GNSS receiver, in our case the u-blox M8T [136]. The same receiver is used at both the base station and in the multirotors.

The base-station is integrated into a portable waterproof case, to provide a robust package suitable for most weather conditions, as illustrated in Figure 2.4. The basestation includes much of the same hardware as the multirotors such as a Beaglebone Black running GLUED and the same u-blox M8T RTK GNSS receiver. However, it uses a larger GPS antenna for better signal reception. In addition, it houses an internal network router with Wifi for optional wireless connection to ground stations (for normal operations, a wired ethernet connection was usually used). Furthermore, the base radio antenna for the telemetry system, described in the next section, is connected to the base.

As mentioned in Section 2.2, a helical L1 active antenna from Maxtena is used for the RTK GPS system onboard the multirotors. The use of a helical antenna ensures good signal reception for a wide range of deflection angles, while still keeping a very low weight. Additionally, it does not need a ground plane, which would be required for a more traditional patch-antenna.

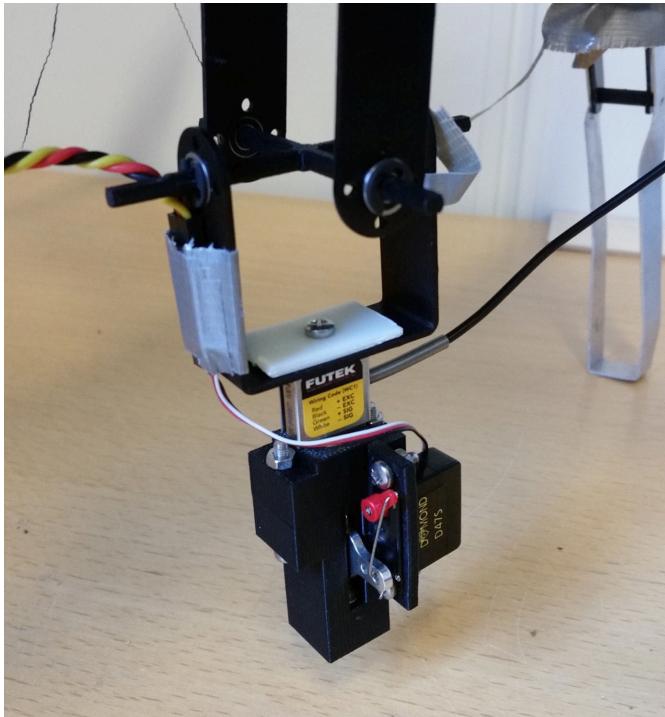


**Figure 2.4:** The base station (black case) in the snow, here shown with the external RTK GPS antenna and telemetry radio system.

## 2.7 Communication and Other Sensors

The vehicles communicate over a wireless 5.8 GHz network, using a radio from Ubiquity Networks. This is an ethernet/IP based radio, with *Time Division Multiple Access* medium control. This ensures constant transfer-delay in the network.

To measure the weight of the suspended load on the multirotors, we have constructed a device to measure the tension and angle of the attached wire. The angles are measured using two digital magnetic encoders (MTS 360 from *PIHER* [103]) in a gimbal-like structure, while the tension is measured using a light-weight load cell (LSB200 from *Futek* [40]).



**Figure 2.5:** Release mechanism for the suspended load, including a force-measurement sensor. The release is driven by a micro-servo, and is designed to be able to release even with tensile forces up to 10 kg.

## Chapter 3

# Single Multirotor UAV with Suspended Load

This chapter is based on **C2** ([64] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control of a Multirotor UAV with Suspended Load. In *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 176 – 184, Denver, CO, 2015. IEEE. doi: 10.1109/ICUAS.2015.7152289)

and **J1** ([67] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control with Swing Damping of a Multirotor UAV with Suspended Load. *Journal of Intelligent & Robotic Systems*, 2017. ISSN 1573-0409. doi: 10.1007/s10846-017-0509-6).

### 3.1 Introduction

This chapter introduces the problem of suspended load transportation using a single multirotor UAV. As discussed in Chapter 1, the ability to safely handle a suspended load enables several interesting applications such as package pickup and delivery, and sensor placement.

When a load is suspended below a multirotor UAV, the UAV is subjected to a disturbance from that load, both due to the added weight and a *centripetal*-like force stemming from the pendulum-motion when the load is swinging. Thus, to synthesize feasible control laws for the UAV, there are two main aspects that needs to be addressed:

- (i) Tracking control of the UAV, while suppressing effects from the swinging suspended mass and environmental forces.
- (ii) Reducing the swing motion of the suspended load.

The next sections gives an overview of the related work, followed by the organization and contributions of this chapter.

#### 3.1.1 Related work

Slung-load systems have been a topic of research for many years. Although this thesis considers the problem of a suspended load below UAVs, it bears close resemblance

### 3. Single Multirotor UAV with Suspended Load

---

to control of overhead cranes. In fact, many of the proposed control solutions for the UAV-slung load transportation problem have its roots in the literature of crane control.

This is especially true for aspects of control related to performing swing-free maneuvers of the load. A suspended load behaves like a pendulum, often with very little natural damping, and excessive oscillations should be avoided. Suppression of unwanted oscillations using feed-forward open loop methods stem from the late 50's [123], and was formalized and applied to mechanical systems in [117]. Here, the authors propose to *pre-shape* the input commands to a system through a filter, which is designed based on the system plant. The filter can be designed to be robust to parametric uncertainties and unmodeled dynamics, at the cost of added response delay. Today, these methods are called *Input Shaping* [118], and have been successfully applied to suppress oscillations of suspended loads attached to overhead gantry cranes [60, 116], and helicopter UAVs [16]. The fundamental of these open-loop approaches can also be adapted to form feedback-loops [53]. New developments remove the induced phase-shift in non rest-to-rest applications such as for automatic assembly on moving conveyer belts [99]. See [119] for an extensive review of applications and developments on input shaping.

For UAVs with a suspended load, reference trajectories for the UAV can be generated to ensure minimal oscillations of the load. In [97], the authors utilize *dynamic programming* to facilitate such open-loop swing reduction using a minimum energy principle. The optimization algorithm is capable of near real-time implementation, and was verified experimentally in an indoor facility. A *machine learning* approach was considered in [33], which relaxed the requirement of known initial conditions from [97]. *Model-Predictive Control* is applied to facilitate swing-free transportation maneuvers in [6, 131, 135]. In [130], the authors consider the multirotor-slung system a *hybrid* system, consisting of two modes that depend on the wire being taut or not. Mixed-integer optimization is applied to generate trajectories that allow the multirotor to switch between the two, to transport the payload in obstacle-filled environments.

By measuring the relative angle of the suspended load, feedback control designs to remove oscillations of the suspended load can be applied. In [72], a *proportional-derivative* term is added to the translational controller of a multirotor UAV, using the suspended load angles as an error input. Although a straight-forward approach, their experimental validation on a small (0.4 kg) indoor platform shows value. Further exploitations of the oscillating nature of the suspended load is presented in [95], where a *delayed feedback* approach is utilized to generate relative position offsets based on delayed measurements of the swing load angle. This allowed rapid attenuations of load swing, and was applied to helicopter slung-systems in [93, 94]. Experimental verification of the method was successfully conducted on a small-scale helicopter UAV in [18].

Reducing the energy of the suspended payload can also be achieved by directly incorporating it into the control dynamics. In [49, 50], the under-actuated nature of the problem is modeled as a *port-controlled Hamiltonian*(PCH) system, and an energy-shaping control technique, *interconnection and damping assignment* (IDA) [96] is applied. The closed loop total-energy storage function is obtained through the solution of a partial differential equation as a result of the choice of desired energy

function form and damping. To simplify the resulting mathematical complexity, the authors divide system into a longitudinal and lateral subsystem, and similarly apply the technique to each, while also only considering horizontal movements. While this might hide some underlying coupling effects, the authors present promising experimental results on a small-scale platform with a very short suspension wire length.

Geometric control control is utilized in [46], which also models the suspended cable as a series of links, rather than a single wire (oft assumed taut) which is common elsewhere. [47] also presents preliminary experimental results, conducted on an indoor platform. See also [124, 125], which establishes the multirotor-load system to be a *differentially flat hybrid system*, and discusses trajectory tracking and geometric control with experimental validation. Robustness to disturbances is also considered in [100], which designs a backstepping type trajectory controller. The authors derive conditions on a desired trajectory to guarantee that the suspension cable remains taut, and preliminary experimental validation on an indoor platform is shown.

For small-size helicopters, an adaptive control and slung load pose estimation is considered in [19], which is verified in a laboratory setting. By using a combination of feed-forward and feed-back control, the swing angles of the slung load is effectively reduced, and serves as an inspiration to the work presented in this chapter on swing damping. Further, in [13, 14, 83], several experimental trials are conducted where a single and several small-scale helicopters transports payloads and drop sensor packages.

A main conclusion to draw is that only a limited number of publications deal with experimental results from outdoor trials, and very few consider environmental disturbances such as wind into the control design.

### 3.1.2 Overview of the Chapter

The rest of the chapter is organized as follows. First, some preliminaries for modeling a multirotor UAV is presented in Section 3.2. Based on that, a control-oriented model is presented in Section 3.3, where the dynamics of the connected suspended load is derived using Kane's equations [28]. Next, a nonlinear backstepping [70] controller *with integral action* is proposed, to create stabilizing shaping functions to achieve a robust tracking controller to solve objective (i). The controller is then verified by numerical simulations and experimental tests. *This is the main contribution of this chapter.* Methods from the literature for suspended load swing angle control is discussed in Section 3.7, in which the results from [18], combined with well known open-loop trajectory generation tools, will be utilized to robustly solve objective (ii). This is followed by a second case-study of numerical simulations and experiments. The chapter is concluded in Section 3.9.

## 3.2 Modeling of Multirotor UAV Dynamics

This section covers some preliminary modeling of the dominant dynamics of a multirotor UAV, which has been well studied in the literature. For a detailed



**Figure 3.1:** Hexacopter type multirotor carrying a suspended load. The mechanism under the UAV body provides measurements of the deflection angle of the attachment wire.

survey, see [80] and references therein. Different choices of kinematic representation of attitude are made throughout the literature. Euler angles are popular and common, and forms the basis for linearized PID control around the hover-state, and will be presented next. Obviously, other representations, such as quaternions or DCM<sup>1</sup>-styles can be used, which would benefit from being singularity free. As the main purpose of this section is the development of a non-high-fidelity simulation environment, the singularity caused by the Euler-angle representation is easily avoided.

### 3.2.1 Kinematics

This section is a brief introduction to the nomenclature used extensively in the field of surface vessels and UAVs, as introduced e.g. [37] and [38]. The NED frame, denoted  $\{n\}$ , is assumed inertial. Coordinates in this frame, denoted  $\boldsymbol{\eta} \in \mathbb{R}^6$  are  $[x^n \ y^n \ z^n \ \boldsymbol{\Theta}^\top]^\top$ .  $\boldsymbol{\Theta} = [\phi \ \theta \ \psi]^\top$  denotes the Euler-angles, as defined by the rotation sequence zyx [37]. The body-fixed coordinate system, denoted  $\{b\}$ , is attached to each rigid body. The body-velocities is given by

$$\boldsymbol{\nu} = [u \ v \ w \ p \ q \ r]^\top \quad (3.1)$$

where  $\boldsymbol{\nu}_1 = [u, v, w]^\top$  and  $\boldsymbol{\nu}_2 = [p, q, r]^\top$  refer to the translational and rotational motions, respectively, and likewise for  $\boldsymbol{\eta}_1$  and  $\boldsymbol{\eta}_2$ . When working with kinematics, it is useful to define the following matrices [37]:

$$\mathbf{R}_b^n := \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (3.2)$$

where  $c \cdot = \cos(\cdot)$  and  $s \cdot = \sin(\cdot)$ .  $\mathbf{R}_b^n$  is the rotation matrix describing the rotation of the frame  $\{b\}$  relative to a frame  $\{n\}$  subject the rotation  $\boldsymbol{\Theta}$ . Further,

---

<sup>1</sup>Direct Cosine Matrix

$$\mathbf{S}(\lambda) := \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad (3.3)$$

generates a skew symmetric matrix of  $\lambda \in \mathbb{R}^3$  satisfying  $\mathbf{S}(\lambda)^\top = -\mathbf{S}(\lambda)$  and  $\lambda \times \lambda = \mathbf{S}(\lambda)\lambda$ . Finally, let

$$\mathbf{T}_\Theta := \begin{bmatrix} 1 & s\phi s\theta/c\theta & c\phi s\theta/c\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \forall \theta \neq \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad (3.4)$$

be the angular transformation matrix, relating angular speeds in  $\{n\}$  and  $\{b\}$  by  $\dot{\Theta} = \mathbf{T}_\Theta \boldsymbol{\nu}_2$

To summarize, velocities in  $\{n\}$  and  $\{b\}$  are related by

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta \boldsymbol{\nu} \quad (3.5)$$

where

$$\mathbf{J}_\Theta = \begin{bmatrix} \mathbf{R}_b^n(\Theta) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_\Theta \end{bmatrix} \quad (3.6)$$

as defined in [37].

### 3.2.2 Kinetics

The rigid-body kinetics of a generic body can be written [37]:

$$m(\dot{\boldsymbol{\nu}}_1 + \boldsymbol{\nu}_2 \times \boldsymbol{\nu}_1) = \boldsymbol{\tau}_1 \quad (3.7)$$

$$\mathbf{I}_{CG}\dot{\boldsymbol{\nu}}_2 + \boldsymbol{\nu}_2 \times (\mathbf{I}_{CG}\boldsymbol{\nu}_2) = \boldsymbol{\tau}_2 \quad (3.8)$$

where  $m$  is the mass of the body,  $\mathbf{I}_{CG} \in \mathbb{R}^{3 \times 3}$  is the moment of inertia about the centre of gravity,  $\boldsymbol{\tau}_1 \in \mathbb{R}^3$  is external forces and  $\boldsymbol{\tau}_2 \in \mathbb{R}^3$  is external moments.

If one assume that  $\{b\}$  is located in the centre of gravity, (3.7)–(3.8) can be rewritten to

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}\boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \quad (3.9)$$

where

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{CG} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} m\mathbf{S}(\boldsymbol{\nu}_2) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{I}_{CG}\boldsymbol{\nu}_2) \end{bmatrix} \end{aligned} \quad (3.10)$$

Further, the gravitational forces acting on the body,  $\mathbf{g}^b$ , can be written

$$\mathbf{g}^b = - \begin{bmatrix} (\mathbf{R}_b^n)^\top \mathbf{f}_G \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (3.11)$$

where  $\mathbf{f}_G = [0 \ 0 \ mg]^\top$  is the gravitational force in  $\{n\}$ . The resulting model now becomes:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta \boldsymbol{\nu} \quad (3.12)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}\boldsymbol{\nu} + \mathbf{g}^b = \boldsymbol{\tau}_A \quad (3.13)$$

where  $\boldsymbol{\tau}_A$  consists of all external forces (except gravity) that affects the system. This includes control forces and air drag.

### 3.2.3 Dominant Multirotor Dynamics

Based on the rigid-body equations derived above, the dominant dynamics of the multirotor UAV is presented next. First, due to the symmetric nature of a multirotor UAV and it's ability to generate thrust in any direction by manipulating it's roll-and pitch angle, it is common in the literature to leave the translational dynamics represented in  $\{n\}$ , rather than  $\{b\}$ . Thus, we re-write the rigid body equations (3.5) and (3.12)–(3.13) as

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3.14)$$

$$\dot{\boldsymbol{\Theta}} = \mathbf{T}(\boldsymbol{\Theta})\boldsymbol{\omega} \quad (3.15)$$

where  $\mathbf{p} = [x^n \ y^n \ z^n] \in \mathbb{R}^3$ .

In total, we get

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3.16)$$

$$m_c \dot{\mathbf{v}} = m_c \mathbf{g} + \mathbf{R}(\boldsymbol{\Theta})\mathbf{f} + \mathbf{w} \quad (3.17)$$

$$\dot{\boldsymbol{\Theta}} = \mathbf{T}(\boldsymbol{\Theta})\boldsymbol{\omega} \quad (3.18)$$

$$I\dot{\boldsymbol{\omega}} = \mathbf{S}(I\boldsymbol{\omega})\boldsymbol{\omega} + \mathbf{M} \quad (3.19)$$

where  $\mathbf{p} \in \mathbb{R}^3$  is the UAV position in the inertial frame  $\{n\}$ ,  $\mathbf{v} \in \mathbb{R}^3$  the linear velocity in  $\{n\}$ ,  $\mathbf{R}(\boldsymbol{\Theta}) \in \mathcal{SO}^3$  a rotation matrix from the inertial frame  $\{n\}$  to the body-fixed frame  $\{b\}$ ,  $\boldsymbol{\omega} \in \mathbb{R}^3$  the angular velocity of the UAV, represented in  $\{b\}$ .  $m_c$  is the mass of the UAV, and  $I$  the body-fixed inertia matrix.  $\mathbf{f}$  is vertical thrust directed along the negative body-aligned  $z$ -axis, and  $\mathbf{M}$  are applied torque about the UAV from the motors.  $\mathbf{w}$  is additional environmental forces, such as wind disturbance and finally  $\mathbf{g} = [0 \ 0 \ g]^\top$ , where  $g$  is the gravitational constant.

## 3.3 Control-Oriented Model for Suspended Load Transportation

Consider now a load being suspended in the centre of gravity of the UAV. This will affect the translational motion (3.17) by a term  $\boldsymbol{\tau}_L$ , parameterized by the load dynamics, but the rotational motion (3.19) is unaffected. As control of the attitude of the multirotor is not the task of the trajectory tracking controller, the model describing the translational motion is now

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3.20)$$

$$m_c \dot{\mathbf{v}} = m_c \mathbf{g} + \mathbf{R}(\Theta) f + \boldsymbol{\tau}_L + \mathbf{w} \quad (3.21)$$

Further, assume now that a sufficiently fast attitude controller is present. The direction of the applied force for translational motion is given by  $\mathbf{R}$ , and by manipulating the roll and pitch angles of the UAV we can apply force in a desired direction. An example of such a controller is given in [80], and directly for Euler angles in the next section. Thus, the term  $\mathbf{R}(\Theta)f$  can be replaced by a virtual inertial control force  $\mathbf{F} = \mathbf{R}(\Theta)f \in \mathbb{R}^3$ , resulting in

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3.22)$$

$$m_c \dot{\mathbf{v}} = m_c \mathbf{g} + \mathbf{F} + \boldsymbol{\tau}_L + \mathbf{w} \quad (3.23)$$

### Generating desired attitude

This section outlines how one translates a desired force  $\mathbf{F}$  to desired Euler angles  $\phi_d$ ,  $\theta_d$  and  $\psi_d$ . It turns out, the desired yaw angle  $\psi_d$  can be set independently of the desired force. The body-oriented forces  $\bar{\mathbf{F}}$  obtained by a principal rotation of  $\mathbf{F}$  around the  $z$ -axis with the current yaw  $\psi$  relates to the roll- and pitch angles as follows

$$F_x^b = -k_f f \cos \phi \sin \theta \quad (3.24)$$

$$F_y^b = k_f f \sin \phi \quad (3.25)$$

$$F_z^b = -k_f f \cos \phi \cos \theta \quad (3.26)$$

where  $f \in [0 \dots 1]$ , and  $k_f \in \mathbb{R}$  is a coefficient of the thrust configuration typically s.t. at  $f = 0.5$  the vehicle is at hover. This equation set can be solved by first solving (3.26) for  $f$  using the current values for  $\psi$  and  $\theta$ . Then, solve (3.24)–(3.25) for  $\phi$  and  $\theta$ , giving the desired values.

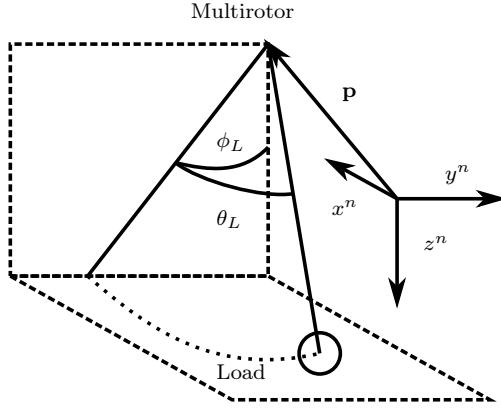
To further simplify; if one assumes low vertical accelerations, the desired roll- and pitch angles can be found by

$$\theta_d = \arctan \frac{-F_x^b}{m_c g} \quad (3.27)$$

$$\phi_d = \arctan \frac{F_y^b \cos \theta_d}{m_c g} \quad (3.28)$$

### Suspended Load Dynamics

Next, the suspended load dynamics is modeled as a pendulum. This simple model has been used with success in earlier work [14]. We consider the suspended load as a point-mass, connected by a rigid link to the centre of gravity of the UAV. This is



**Figure 3.2:** Illustration of pendulum angle parameterization by two consecutive rotations  $\phi_L$  and  $\theta_L$  about the  $x$ - and  $y$ -axis, respectively.

valid for non-aggressive maneuvers where the wire remains taut. We parameterize the pendulum displacements by  $\phi_L$  and  $\theta_L$ , which are the load rotation angles about the inertial  $x$ - and  $y$ -axis, respectively, see Figure 3.2. The generalized coordinates for the system are thus  $\boldsymbol{\eta} = [\mathbf{p}^\top, \phi_L, \theta_L]^\top$ . Physical damping of the pendulum swing is modeled by a linear damping term, which is valid for low speeds. By utilizing Kane's equation [28], we get the dynamical model (3.29):

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{\nu} \\ \mathbf{M}^*(\boldsymbol{\eta})\dot{\boldsymbol{\nu}} + \mathbf{C}^*(\boldsymbol{\eta}, \boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{G}(\boldsymbol{\eta}) + \mathbf{D}\boldsymbol{\nu} &= \boldsymbol{\tau} + \boldsymbol{\tau}_a \end{aligned} \quad (3.29)$$

where  $\mathbf{M}^*(\boldsymbol{\eta})$  and  $\mathbf{C}^*(\boldsymbol{\eta}, \boldsymbol{\nu})$  can be seen in (3.31)–(3.32),  $\boldsymbol{\tau} := [\mathbf{F}^\top \quad \mathbf{0}_{1 \times 2}]^\top$ ,  $\boldsymbol{\nu} := [\mathbf{v}^\top, \dot{\phi}_L, \dot{\theta}_L]^\top$ ,  $\boldsymbol{\tau}_a := [\mathbf{w}^\top \quad \mathbf{0}_{1 \times 2}]^\top$  and

$$\mathbf{G}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ -g(m_L + m_c) \\ L g m_L \cos \theta_L \sin \phi_L \\ L g m_L \cos \phi_L \sin \theta_L \end{bmatrix}. \quad (3.30)$$

The term  $\mathbf{D}\boldsymbol{\nu}$  is the linear damping force, incorporating the effects of air drag at low speed.  $\mathbf{D} = \text{diag}\{[0, 0, 0, d, d]\}$ , where  $d > 0$  is a drag coefficient.  $\mathbf{w}$  is an unknown disturbance acting on the body of the UAV, typically wind. This is assumed constant, or slowly varying.

As can be seen in (3.29), the equations are organized in matrix form, which is typical in the literature of robotic manipulation. This form greatly simplifies the control analysis. In fact, as is common in robotic manipulators, the matrix  $\dot{\mathbf{M}}^*(\boldsymbol{\eta}) - 2\mathbf{C}^*(\boldsymbol{\eta}, \boldsymbol{\nu})$  is skew symmetric, which is a very useful property in Lyapunov analysis. As can be seen from (3.31),  $\mathbf{M}^*(\boldsymbol{\eta})$  is singular at  $\theta_L = \pi/2$ . This is due to the representation of the pendulum configuration, and it corresponds to the case where the suspended load is directed out of the nose of the UAV. Although while

$$\begin{aligned}
 \mathbf{M}^*(\boldsymbol{\eta}) &= \begin{bmatrix} m_L + m_c & 0 & 0 & 0 \\ 0 & m_L + m_c & 0 & -Lm_L c\phi_L c\theta_L \\ 0 & 0 & m_L + m_c & -Lm_L s\phi_L s\theta_L \\ 0 & -Lm_L c\phi_L c\theta_L & -Lm_L c\theta_L s\phi_L & -Lm_L c\phi_L s\theta_L \\ Lm_L c\theta_L & Lm_L s\phi_L s\theta_L & -Lm_L c\phi_L s\theta_L & 0 \end{bmatrix} \quad (3.31) \\
 \mathbf{C}^*(\boldsymbol{\eta}, \boldsymbol{\nu}) &= \begin{bmatrix} 0 & 0 & 0 & -L\dot{\theta}_L m_L s\theta_L \\ 0 & 0 & Lm_L (\dot{\phi}_L c\theta_L s\phi_L + \dot{\theta}_L c\phi_L s\theta_L) & Lm_L (\dot{\phi}_L c\phi_L s\theta_L + \dot{\theta}_L c\theta_L s\phi_L) \\ 0 & 0 & -Lm_L (\dot{\phi}_L c\phi_L c\theta_L - \dot{\theta}_L s\phi_L s\theta_L) & -Lm_L (\dot{\theta}_L c\phi_L c\theta_L - \dot{\phi}_L s\phi_L s\theta_L) \\ 0 & 0 & -\frac{1}{2}L^2 \dot{\theta}_L m_L s(2\theta_L) & -\frac{1}{2}L^2 \dot{\phi}_L m_L s(2\theta_L) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.32)
 \end{aligned}$$

in this configuration the rigid-link assumption is unlikely to hold in practice, and would most likely result in a crash due to impact between the suspended load and multirotor propellers, this situation is prevented by the design of the controller. We illustrate in the next section how a infinitesimal *singularity avoidance* term can be added to use the model for theoretically valid control design and analysis.

### 3.4 Problem Description and Control Strategies

We consider the objective of the UAV to safely transport a suspended payload, by tracking a user defined trajectory or waypoint route, while suppressing the disturbance from the load swing and external environmental effects, such as wind. The UAV in question is equipped with an internal Inertial Measurement Unit (IMU) and GNSS system, with the accompanied navigation filter providing full state (attitude and position) output for the controllers. Further, the attachment-point of the suspended load is a gimbal-like structure, providing measurements of the displacement angles (relative to the UAV body) of the load suspended by a wire, as seen in Figure 3.1.

### 3.5 Nonlinear Trajectory Tracking

In this section, we design a nonlinear trajectory-tracking controller for the UAV, to resolve objective (i). We utilize the backstepping technique [70], to design the controllers in two steps. To achieve this, with a slight abuse of terminology, we view (3.29) as an underactuated system, in which elements 4 and 5 of the control input vector  $\tau$  are constrained to be zero. The design procedure is inspired by [36].

The model developed in the previous section has a representation singularity at  $\theta_L = \pi/2$ . This is an undesired feature when designing the control system. Thus, we create a perturbed model, in which we avoid the singularity by adding an infinitesimal *singularity avoidance* term at  $M_{4,4}^*$ . The model used for control design are then:

$$\dot{\eta} = \nu \quad (3.33)$$

$$M(\eta)\dot{\nu} + C(\eta, \nu)\nu + G(\eta) + D\nu = \tau + \tau_a \quad (3.34)$$

where  $M(\eta) = M^*(\eta)$ ,  $C(\eta, \nu) = C^*(\eta, \nu)$  except for

$$M(\eta)_{4,4} = L^2 m_L \cos \theta_L^2 + \varepsilon \sin \theta_L^2, \quad (3.35)$$

and

$$C(\eta, \nu)_{4,4} = -\frac{1}{2} L^2 m_L \dot{\theta}_L \sin 2\theta_L + \frac{1}{2} \varepsilon \dot{\theta}_L \sin 2\theta_L. \quad (3.36)$$

$\varepsilon > 0$  is a small positive constant. As can be seen,  $C(\eta, \nu)$  is also perturbed to maintain the skew-symmetric property. This added term ensures that the mass matrix is non-singular for all  $(\theta_L, \phi_L)$  and close to the original model for small  $(\theta_L, \phi_L)$ . This perturbed model is now used in the rest of this section.

Consider now a sufficiently smooth desired trajectory  $\mathbf{r}(t)$ . We define the error signals  $\mathbf{z}_1 \in \mathbb{R}^3$  and  $\mathbf{z}_2 \in \mathbb{R}^5$  as

$$\mathbf{z}_1 := \mathbf{p} - \mathbf{r} \quad (3.37)$$

$$\mathbf{z}_2 := [z_{2,1}, z_{2,2}, z_{2,3}, z_{2,4}, z_{2,5}]^\top := \boldsymbol{\nu} - \boldsymbol{\alpha} \quad (3.38)$$

where  $\boldsymbol{\alpha} := [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5]^\top \in \mathbb{R}^5$  is a vector of stabilizing functions to be specified later.

Now, we utilize a selection matrix  $\mathbf{H}$  to select the states of  $\mathbf{z}_1$  we are interested in for control. Let

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.39)$$

such that the error-dynamics becomes

$$\dot{\mathbf{z}}_1 = \mathbf{H}\boldsymbol{\nu} - \dot{\mathbf{r}} \quad (3.40)$$

and

$$\begin{aligned} \mathbf{M}(\boldsymbol{\eta})\dot{\mathbf{z}}_2 &= \mathbf{M}(\boldsymbol{\eta})\dot{\boldsymbol{\nu}} - \mathbf{M}(\boldsymbol{\eta})\dot{\boldsymbol{\alpha}} \\ &= \boldsymbol{\tau} + \boldsymbol{\tau}_a - \mathbf{C}(\boldsymbol{\eta}, \boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{G}(\boldsymbol{\eta}) \\ &\quad - \mathbf{D}\boldsymbol{\nu} - \mathbf{M}(\boldsymbol{\eta})\dot{\boldsymbol{\alpha}} \end{aligned} \quad (3.41)$$

### 3.5.1 Step 1

Consider a Lyapunov function candidate (LFC) for the first sub-system (3.40) as

$$V_1(\mathbf{z}_1, t) = \frac{1}{2}\mathbf{z}_1^\top \mathbf{z}_1 \quad (3.42)$$

It's derivative along the solution of  $\mathbf{z}_1(t)$  is

$$\begin{aligned} \dot{V}_1(\mathbf{z}_1, t) &= \mathbf{z}_1^\top (\mathbf{H}\boldsymbol{\nu} - \dot{\mathbf{r}}) \\ &= \mathbf{z}_1^\top (\boldsymbol{\alpha}_{1:3} + \mathbf{H}\mathbf{z}_2 - \dot{\mathbf{r}}) \end{aligned} \quad (3.43)$$

where  $(\cdot)_{i:j}$  represents elements  $i$  to  $j$  of the vector  $(\cdot)$ . By designing the stabilizing functions  $\boldsymbol{\alpha}_{1:3}$  as

$$\boldsymbol{\alpha}_{1:3} = \dot{\mathbf{r}} - \mathbf{K}_1 \mathbf{z}_1, \quad (3.44)$$

where  $\mathbf{K}_1 = \mathbf{K}_1^\top > 0$  is a positive definite matrix. The derivative of  $V_1(\mathbf{z}_1, t)$  becomes

$$\dot{V}_1(\mathbf{z}_1, t) = -\mathbf{z}_1^\top \mathbf{K}_1 \mathbf{z}_1 + \mathbf{z}_1^\top \mathbf{H}\mathbf{z}_2 \quad (3.45)$$

### 3.5.2 Step 2

Take the second LFC as

$$V_2(\mathbf{z}_1, \mathbf{z}_2, \boldsymbol{\xi}, t) = V_1(\mathbf{z}_1, t) + \frac{1}{2} \mathbf{z}_2^\top \mathbf{M}(\boldsymbol{\eta}) \mathbf{z}_2 \quad (3.46)$$

which is positive definite for  $(\mathbf{z}_1, \mathbf{z}_2) \neq 0$ . Taking the derivative of (3.46) yields:

$$\begin{aligned} \dot{V}_2(\mathbf{z}_1, \mathbf{z}_2, t) &= -\mathbf{z}_1^\top \mathbf{K}_1 \mathbf{z}_1 \\ &\quad + \mathbf{z}_2^\top (\boldsymbol{\tau} + \boldsymbol{\tau}_a - \mathbf{C}\boldsymbol{\alpha} - \mathbf{G} - \mathbf{D}\boldsymbol{\nu} + \mathbf{H}^\top \mathbf{z}_1 - \mathbf{M}\dot{\boldsymbol{\alpha}}) \end{aligned}$$

where we have utilized the skew-symmetric property of  $\frac{1}{2} \dot{\mathbf{M}}(\boldsymbol{\eta}) - \mathbf{C}(\boldsymbol{\eta}, \boldsymbol{\nu})$ . Suppose now that the control  $\boldsymbol{\tau}$  can be set to

$$\boldsymbol{\tau} = \mathbf{C}\boldsymbol{\alpha} + \mathbf{G} + \mathbf{D}\boldsymbol{\alpha} - \mathbf{H}^\top \mathbf{z}_1 + \mathbf{M}\dot{\boldsymbol{\alpha}} - \mathbf{K}_2 \mathbf{z}_2 \quad (3.47)$$

where  $\mathbf{K}_2 = \mathbf{K}_2^\top > 0$ . By remembering  $\boldsymbol{\tau}_a = \mathbf{H}^\top \mathbf{w}$ , this results in

$$\dot{V}_2(\mathbf{z}_1, \mathbf{z}_2, t) = -\mathbf{z}_1^\top \mathbf{K}_1 \mathbf{z}_1 - \mathbf{z}_2^\top \mathbf{K}_2 \mathbf{z}_2 - \mathbf{z}_2^\top \mathbf{D} \mathbf{z}_2 + \mathbf{z}_2^\top \mathbf{H}^\top \mathbf{w}$$

Clearly, if  $\mathbf{w} \equiv 0$ , or if it was exactly known, it could be cancelled by the controller. Since this is quite unrealistic, we use the augmented controller

$$\boldsymbol{\tau} = \mathbf{C}\boldsymbol{\alpha} + \mathbf{G} + \mathbf{D}\boldsymbol{\alpha} - \mathbf{H}^\top \mathbf{z}_1 + \mathbf{M}\dot{\boldsymbol{\alpha}} - \mathbf{K}_2 \mathbf{z}_2 - \mathbf{H}^\top \hat{\mathbf{w}} \quad (3.48)$$

where  $\hat{\mathbf{w}}$  is the estimate of  $\mathbf{w}$ , given by

$$\dot{\hat{\mathbf{w}}} = \rho \mathbf{H} \mathbf{z}_2 \quad (3.49)$$

where  $\rho \in \mathbb{R}$  is a tuning parameter.

In the following lemma, we show that with this choice of  $\boldsymbol{\tau}$  and update law for  $\hat{\mathbf{w}}$ , the origin of the system (3.40)–(3.41) and (3.49) is *Uniformly Globally Asymptotically Stable* (UGAS) and *Uniformly Locally Exponentially Stable* (ULES). However, as we cannot set a desired moment about the suspension point of the load, the fourth and fifth row of (3.48) must instead be equal to zero. In Section 3.5.3, we design the remaining stabilizing functions to achieve this.

The closed-loop system can be written as

$$\begin{bmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{K}_1 & \mathbf{H} \\ A_{21}(t) & A_{22}(t) \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\Theta(t)) \mathbf{H}^\top \end{bmatrix} \tilde{\mathbf{w}} \quad (3.50)$$

$$\dot{\tilde{\mathbf{w}}} = \rho \mathbf{H} \mathbf{z}_2 \quad (3.51)$$

where

$$A_{21}(t) := -\mathbf{M}^{-1}(\Theta(t)) \mathbf{H}^\top$$

$$A_{22}(t) := \mathbf{M}^{-1}(\Theta(t))(-\mathbf{K}_2 - \mathbf{D} - \mathbf{C}(\Theta(t), \dot{\Theta}(t)))$$

and  $\Theta(t) := [\theta_L(t), \phi_L(t)]^\top$ .

**Lemma 3.1.** *The origin of (3.50)–(3.51) is UGAS and ULES.*

*Proof.* To prove UGAS and ULES of the equilibrium point  $(\mathbf{z}_1, \mathbf{z}_2, \tilde{w}) = 0$  of (3.50)–(3.51), we apply the results from [39]. To this end, the closed-loop system (3.50)–(3.51) can be restated in the form:

$$\dot{x}_1 = h(x_1, t) + G(x, t)x_2 \quad (3.52)$$

$$\dot{x}_2 = -PG(x, t)^\top \left( \frac{\partial W(x_1, t)}{\partial x_1} \right)^\top, \quad P = P^\top > 0 \quad (3.53)$$

where

$$\begin{aligned} x_1 &:= [\mathbf{z}_1 \quad \mathbf{z}_2]^\top \\ x_2 &:= \tilde{\mathbf{w}} \\ h(x_1, t) &:= \begin{bmatrix} -\mathbf{K}_1 & \mathbf{H} \\ A_{21}(t) & A_{22}(t) \end{bmatrix} x_1 \\ G(x, t) &:= [0 \quad \mathbf{M}^{-1}(\Theta(t))\mathbf{H}^\top]^\top \\ P &:= \rho\mathbf{I} \\ W(x_1, t) &= \frac{1}{2}\mathbf{z}_2^\top \mathbf{M} \mathbf{z}_2 + \frac{1}{2}\mathbf{z}_1^\top \mathbf{z}_1 \\ &\Rightarrow \frac{\partial W(x_1, t)}{\partial x_1} = [\mathbf{z}_1 \quad \mathbf{M}(\Theta(t))\mathbf{z}_2] \end{aligned}$$

[39, Theorem 1] states that if the conditions (A1)–(A2) below hold, the origin of the system (3.52)–(3.53) is UGAS and ULES. Let  $G_0(x_2, t) := G(x, t)|_{x_1=0}$ , and  $\rho_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ , ( $j = 1, 2, 3$ ) be continuous nondecreasing functions. The conditions of A1 are

$$\max \left\{ \|h(x_1, t)\|, \left\| \frac{\partial W(x_1, t)}{\partial x_1} \right\| \right\} \leq \rho_1(\|x_1\|) \|x_1\| \quad (\text{A1.a})$$

$$\max \{\|G(x, t)\|, \|G_0(x_2, t)\|\} \leq \rho_2(\|x\|) \quad (\text{A1.b})$$

$$\max \left\{ \left\| \frac{\partial G_0(x_2, t)}{\partial x_2} \right\|, \left\| \frac{\partial G_0(x_2, t)}{\partial t} \right\| \right\} \leq \rho_3(\|x_2\|) \quad (\text{A1.c})$$

Indeed, (A1.a) holds because  $\mathbf{M}(\Theta(t))$  is non-singular, and is bounded for all  $\Theta(t)$ . Further, as we have  $G_0(x_2, t) = [0 \quad \mathbf{M}^{-1}(\Theta(t))\mathbf{H}^\top]^\top$ , (A1.b) holds since  $\mathbf{M}(\Theta(t))$  has an upper bound. In fact, there exists  $m_m, m_M \in \mathbb{R}$  s.t.  $\mathbf{I}m_m \leq \mathbf{M}(\Theta(t)) \leq \mathbf{I}m_M$ .

By noting that the system (3.52)–(3.53) is *forward complete* [7], we have existence and boundedness of  $\dot{\Theta} \forall t$ . Thus, the partial time derivative of  $G_0(x_2, t)$  is bounded, and (A1.c) holds. Additionally, we need the condition

$$G_0(x_2, t)^\top G_0(x_2, t) \geq b_m \mathbf{I} \quad (\text{A1.d})$$

which holds with in our case with  $b_m = (1/m_M^2)\mathbf{I}$

Further, let  $\kappa_1, \kappa_2$  be class- $\mathcal{K}_\infty$  functions, and  $c > 0$  a strictly positive real number. Then, we need

$$\kappa_1(\|x_1\|) \leq W(x_1, t) \leq \kappa_2(\|x_1\|) \quad (\text{A2.a})$$

$$\frac{\partial W(x_1, t)}{\partial t} + \frac{\partial W(x_1, t)}{\partial x_1} h(x_1, t) \leq -c \|x_1\|^2 \quad (\text{A2.b})$$

Clearly, by setting  $\kappa_1 = 0.5k_1 \|x_1\|^2$ ,  $\kappa_2 = 0.5k_2 \|x_1\|^2$  where  $k_1 = \min(m_m, 1)$ ,  $k_2 = \max(m_M, 1)$ , (A2.a) holds. (A2.b) expands to

$$-\mathbf{z}_1^\top \mathbf{K}_1 \mathbf{z}_1 - \mathbf{z}_2^\top \mathbf{K}_2 \mathbf{z}_2 \leq -c \|x_1\|^2 \quad (3.54)$$

with  $c = \min\{\inf(\mathbf{K}_1), \inf(\mathbf{K}_2)\}$ . In addition, we have  $\kappa_2(s) \propto s^2$ .

We have now shown that all the assumptions (A1-A2) of [39, Theorem 1] holds, and conclude that the origin of (3.52)–(3.53) is UGAS and ULES.  $\square$

### 3.5.3 Remaining stabilizing functions

We now design the remaining stabilizing functions  $\alpha_4$  and  $\alpha_5$  to ensure that the fourth and fifth row of (3.48) is indeed zero. By extracting these rows from (3.48), we get the constraint equations

$$\begin{aligned} \boldsymbol{\tau}_{4:5} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{C}_{4:5;4:5} \boldsymbol{\alpha}_{4:5} + \mathbf{G}_{4:5} + \mathbf{D}_{4:5;4:5} \boldsymbol{\alpha}_{4:5} \\ &\quad + \mathbf{M}_{4:5;4:5} \dot{\boldsymbol{\alpha}}_{4:5} + \mathbf{M}_{4:5,1:3} \dot{\boldsymbol{\alpha}}_{1:3} - \mathbf{K}_{2,4:5} \mathbf{z}_{2,4:5} \end{aligned}$$

where the arguments of  $\mathbf{C}$  and  $\mathbf{M}$  have been dropped for notational clarity, and  $(\cdot)_{i:j;k:l}$  extracts rows  $i - j$ , columns  $k - l$  of  $(\cdot)$ . Solving for  $\dot{\boldsymbol{\alpha}}_{4:5}$ , this results in a dynamic equality constraint

$$\begin{aligned} \mathbf{M}_\alpha(\boldsymbol{\eta}) \dot{\boldsymbol{\alpha}}_{4:5} &= -\mathbf{D}_\alpha \boldsymbol{\alpha}_{4:5} - \mathbf{C}_\alpha(\boldsymbol{\eta}, \boldsymbol{\nu}) \boldsymbol{\alpha}_{4:5} \\ &\quad + \boldsymbol{\gamma}(\boldsymbol{\eta}, \mathbf{z}_1, \mathbf{z}_2, \ddot{\mathbf{r}}) \end{aligned} \quad (3.55)$$

where

$$\begin{aligned} \mathbf{M}_\alpha(\boldsymbol{\eta}) &= \mathbf{M}_{4:5;4:5}(\boldsymbol{\eta}), \\ &\quad \text{the lower } 2 \times 2 \text{ block of } \mathbf{M}(\boldsymbol{\eta}) \\ \mathbf{C}_\alpha(\boldsymbol{\eta}, \boldsymbol{\nu}) &= \mathbf{C}_{4:5;4:5}(\boldsymbol{\eta}, \boldsymbol{\nu}), \\ &\quad \text{the lower } 2 \times 2 \text{ block of } \mathbf{C}(\boldsymbol{\eta}, \boldsymbol{\nu}) \\ \mathbf{D}_\alpha &= \text{diag}\{[d, d]\} > 0 \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{\gamma}(\boldsymbol{\eta}, \mathbf{z}_1, \mathbf{z}_2, \ddot{\mathbf{r}}) &= -\mathbf{G}_{4:5} + \mathbf{K}_{2,4:5} \mathbf{z}_{2,4:5} \\ &\quad - \mathbf{M}_{4:5,1:3} (\ddot{\mathbf{r}} - \mathbf{K}_1 \mathbf{H} \mathbf{z}_2 + \mathbf{K}_1 \mathbf{K}_1 \mathbf{z}_1) \end{aligned}$$

in which we have used the fact that

$$\begin{aligned} \dot{\boldsymbol{\alpha}}_{1:3} &= \ddot{\mathbf{r}} - \mathbf{K}_1 \dot{\mathbf{z}}_1 \\ &= \ddot{\mathbf{r}} - \mathbf{K}_1 \mathbf{H} \mathbf{z}_2 + \mathbf{K}_1 \mathbf{K}_1 \mathbf{z}_1 \end{aligned} \quad (3.56)$$

Also, note that the matrix  $\dot{\mathbf{M}}_\alpha - 2\mathbf{C}_\alpha$  retains the skew-symmetric property. The variables  $\boldsymbol{\alpha}_{4:5}$  becomes a dynamic state in the controller, according to (3.55). In fact, (3.55) is a stable differential equation driven by the converging error signals  $(\mathbf{z}_1, \mathbf{z}_2)$  and the bounded signal  $\ddot{\mathbf{r}}$ . As  $\mathbf{z}_{2,4:5}(t) \rightarrow 0$ , it follows that  $|\boldsymbol{\alpha}_{4:5} - [\phi_L, \theta_L]^\top| \rightarrow 0$  as  $t \rightarrow \infty$ . This is stated more formally in Theorem 3.2.

**Theorem 3.2.** *Let the trajectory tracking problem (3.37)–(3.38) of a multirotor UAV be solved by applying the first three rows of the control law (3.48) to (3.34):*

$$\begin{aligned}\mathbf{F} = & \mathbf{C}_{1:3;4:5}(\boldsymbol{\eta}, \boldsymbol{\nu})\boldsymbol{\alpha}_{4:5} + \mathbf{G}_{1:3} - \mathbf{z}_1 \\ & + \mathbf{M}_{1:3;1:3}(\ddot{\mathbf{r}} - \mathbf{K}_1 \mathbf{H} \mathbf{z}_2 + \mathbf{K}_1 \mathbf{K}_1 \mathbf{z}_1) \\ & + \mathbf{M}_{1:3;4:5} \dot{\boldsymbol{\alpha}}_{4:5} - \mathbf{K}_{2,1:3} \mathbf{z}_{2,1:3} - \hat{\mathbf{w}}\end{aligned}\quad (3.57)$$

where  $\mathbf{K}_1 > 0 \in \mathbb{R}^{3 \times 3}$ ,  $\mathbf{K}_2 > 0 \in \mathbb{R}^{5 \times 5}$ ,  $\mathbf{z}_1 := \mathbf{p} - \mathbf{r}$ ,  $\mathbf{z}_2 := \boldsymbol{\nu} - \boldsymbol{\alpha}$ , and

$$\boldsymbol{\alpha}_{1:3} = \dot{\mathbf{r}} - \mathbf{z}_1 \quad (3.58)$$

The smooth reference signals  $\mathbf{r}$ ,  $\dot{\mathbf{r}}$  and  $\ddot{\mathbf{r}}$  are provided by an external guidance system or a reference model, while  $\boldsymbol{\alpha}_{4:5}$  are given by the dynamic system

$$\begin{aligned}\mathbf{M}_\alpha(\boldsymbol{\eta}) \dot{\boldsymbol{\alpha}}_{4:5} = & -\mathbf{D}_\alpha \boldsymbol{\alpha}_{4:5} - \mathbf{C}_\alpha(\boldsymbol{\eta}, \boldsymbol{\nu}) \boldsymbol{\alpha}_{4:5} \\ & + \gamma(\boldsymbol{\eta}, \mathbf{z}_1, \mathbf{z}_2, \ddot{\mathbf{r}})\end{aligned}\quad (3.59)$$

Then the equilibrium point  $(\mathbf{z}_1, \mathbf{z}_2) = 0$  is UGAS,  $\boldsymbol{\alpha}_{4:5} \in \mathcal{L}_\infty$  and satisfies

$$\lim_{t \rightarrow \infty} |\boldsymbol{\alpha}_{4:5}(t) - [\dot{\phi}_L(t), \dot{\theta}_L(t)]^\top| = 0 \quad (3.60)$$

*Proof.* The closed-loop system can be expressed as

$$\Sigma_1 \left\{ \begin{array}{l} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \\ \dot{\tilde{\mathbf{w}}} \end{array} \right\} = \begin{bmatrix} -\mathbf{K}_1 & \mathbf{H} \\ A_{21}(t) & A_{22}(t) \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\Theta(t)) \mathbf{H}^\top \end{bmatrix} \tilde{\mathbf{w}} \quad (3.61)$$

$$\Sigma_2 \left\{ \begin{array}{l} \mathbf{M}_\alpha(\boldsymbol{\eta}) \dot{\boldsymbol{\alpha}}_{4:5} \\ \gamma(\boldsymbol{\eta}, \mathbf{z}_1, \mathbf{z}_2, \ddot{\mathbf{r}}) \end{array} \right\} = -\mathbf{D}_\alpha \boldsymbol{\alpha}_{4:5} - \mathbf{C}_\alpha(\boldsymbol{\eta}, \boldsymbol{\nu}) \boldsymbol{\alpha}_{4:5} \quad (3.62)$$

From Lemma 3.1 we know that the origin of  $\Sigma_1$  is UGAS and ULES. The unforced  $\boldsymbol{\alpha}_{4:5}$  system,  $\Sigma_2$  with  $\gamma = 0$ , is globally exponentially stable. This can be seen from the Lyapunov function

$$V_\alpha(\boldsymbol{\alpha}_{4:5}, t) = \frac{1}{2} \boldsymbol{\alpha}_{4:5}^\top \mathbf{M}_\alpha(\boldsymbol{\eta}) \boldsymbol{\alpha}_{4:5} \quad (3.63)$$

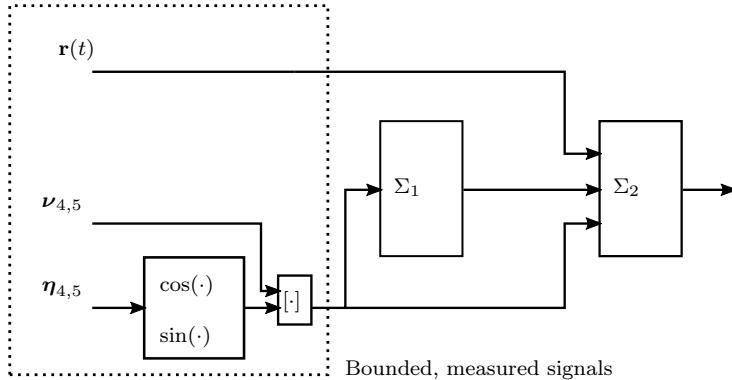
whose derivative along (3.55) is

$$\begin{aligned}\dot{V}_\alpha(\boldsymbol{\alpha}_{4:5}, t) = & \boldsymbol{\alpha}_{4:5}^\top (-\mathbf{D}_\alpha \boldsymbol{\alpha}_{4:5} + \gamma) \\ & \leq -\frac{1}{2} \boldsymbol{\alpha}_{4:5}^\top \mathbf{D}_\alpha \boldsymbol{\alpha}_{4:5} \leq 0\end{aligned}$$

for  $\|\alpha_{4:5}\|_2 > 2\|\gamma\|_2/d$ . Since  $(\mathbf{z}_1, \mathbf{z}_2) \in \mathcal{L}_\infty$ ,  $\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}} \in \mathcal{L}_\infty$  and since  $\boldsymbol{\eta}$  enters  $\gamma$  through the bounded functions  $\cos$  and  $\sin$ , we have  $\gamma(\boldsymbol{\eta}, \mathbf{z}_1, \mathbf{z}_2, \ddot{\mathbf{r}}) \in \mathcal{L}_\infty$ . Thus, the  $\alpha_{4:5}$  subsystem is input-to-state stable from  $\gamma$  to  $\alpha_{4:5}$  by [61, Theorem 4.19]. Hence,  $\alpha_{4:5}$  converges to the bounded set  $\{\alpha_{4:5} : \|\alpha_{4:5}\| \leq (2/d)\|\gamma\|\}$ . The limit (3.60) is obtained from the fact that  $\mathbf{z}_{2,4:5}(t) \rightarrow 0$  as  $t \rightarrow \infty$ .  $\square$

**Remark 3.1.** Note that the global results presented in Theorem 3.2 applies to the perturbed model (3.34). This model is valid as long as the wire is taut, and we stay away from the representation singularity at  $\theta_L = \pi/2$ . At this point, the model is perturbed by the infinitesimal constant  $\varepsilon$ , which in practice will have negligible impact on the results.

**Remark 3.2.** Also, note that the load angles  $\phi_L$  and  $\theta_L$  are not included in the coordinate change (3.37)–(3.38), but their derivatives are. The angles acts as external inputs to the system dynamics through the saturating geometric functions  $\cos$  and  $\sin$ , and are thus treated as bounded inputs in the analysis. See Figure 3.3.



**Figure 3.3:** Structure illustrating the proof. By Theorem 3.2, the feedback law (3.57) renders the origin of  $[\mathbf{z}_1, \mathbf{z}_2]$  globally asymptotically stable. The signal  $\alpha_{4:5}$ , needed by (3.57), is shown to be bounded. The load angles act on both sub-systems through the saturating trigonometric geometric functions  $\cos(\cdot)$  and  $\sin(\cdot)$ , and  $\nu_{4,5}$  is bounded by (3.60).

### 3.5.4 Implementational Aspects

When implementing backstepping controllers, it is often advantageous to re-write the final control output in the original UAV coordinates to facilitate easier tuning of the various gains. Indeed, by re-writing the controller equation (3.57), and define the error signal  $\tilde{\mathbf{p}} := \mathbf{p} - \mathbf{r}$ , the PID-like structure can be recovered:

$$\begin{aligned} \mathbf{F} = & \beta(\cdot) + \mathbf{G}(\boldsymbol{\eta}) + (m_L + m_c)\ddot{\mathbf{r}} \\ & - \mathbf{K}_d \dot{\tilde{\mathbf{p}}} - \mathbf{K}_p \tilde{\mathbf{p}} - \mathbf{K}_i \int_0^t \tilde{\mathbf{p}} d\tau \end{aligned} \quad (3.64)$$

where we recognize the gravity compensation, reference acceleration feed forward, the proportional, derivative and integral feedback terms. In addition, the stabilizing load compensator  $\beta(\cdot)$  which is a function of  $(\boldsymbol{\eta}, \boldsymbol{\nu}, \mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \boldsymbol{\alpha}_{4,5})$ , acts to cancel the effects of the suspended load. It is given by

$$\boldsymbol{\beta} = \mathbf{C}_{1:3,4:5}(\boldsymbol{\eta}, \boldsymbol{\nu})\boldsymbol{\alpha}_{4:5} + \mathbf{M}_{1:3,4:5}\dot{\boldsymbol{\alpha}}_{4:5} \quad (3.65)$$

and where  $\boldsymbol{\alpha}_{4:5}$  is driven by the dynamic constraint

$$\begin{aligned} \mathbf{M}_\alpha \dot{\boldsymbol{\alpha}}_{4:5} = & -(\mathbf{D}_\alpha + \mathbf{C}_\alpha(\boldsymbol{\eta}, \boldsymbol{\nu}) + \mathbf{K}_{2,4:5})\boldsymbol{\alpha}_{4:5} \\ & - \mathbf{G}_{4:5} + \mathbf{K}_{2,4:5}\dot{\boldsymbol{\Theta}} - \mathbf{M}_{4:5,1:3}(\boldsymbol{\eta})(\ddot{\mathbf{r}} - \mathbf{K}_1\dot{\mathbf{p}}) \end{aligned} \quad (3.66)$$

and the gains are given by

$$\mathbf{K}_p = \mathbf{I} + \mathbf{K}_{2,1:3}\mathbf{K}_1 + \rho\mathbf{I} \quad (3.67)$$

$$\mathbf{K}_d = (m_L + m_c)\mathbf{K}_1 + \mathbf{K}_{2,1:3} \quad (3.68)$$

$$\mathbf{K}_i = \rho\mathbf{K}_1 \quad (3.69)$$

As the dominating elements of (3.64) are the proportional and derivative terms, (3.67)–(3.69) allows the use of tools like pole-placement for linear systems to be used when tuning the final controller.

### 3.5.5 Generating Reference Signals

The controller presented above requires a smooth  $\mathcal{C}^3$  trajectory. Given a sequence of waypoints given by the operator, the trajectory can be generated by feeding the waypoints through a reference model of sufficient order. Consider a fourth order low-pass filter:

$$\mathbf{x}^{(4)} + 4\zeta\omega_0\mathbf{x}^{(3)} + (2 + 4\zeta^2)\omega_0^2\ddot{\mathbf{x}} + 4\zeta\omega_0^3 + \omega_0^4\mathbf{x} = \omega_0^4\mathbf{x}_d \quad (3.70)$$

where  $\mathbf{x} \in \mathbb{R}^3$  is the reference signal,  $\mathbf{x}_d \in \mathbb{R}^3$  is the current desired position (waypoint), and  $\zeta, \omega \in \mathbb{R}$  are tuning parameters. Consider only the first dimension of  $\mathbf{x}$ , a reference position of  $\mathbf{x}_{d,1} = 20$  would give the trajectory illustrated by the dashed lines in Figure 3.4. However, this is not a feasible trajectory for the UAV as the UAV encompasses certain dynamic constraints, like maximum acceleration and jerk, that is not taken into account. Further, the operator would typically like to set a prescribed velocity for the maneuver. Thus, in this work, we re-arrange (3.70) as a set of cascaded controllers, and impose saturations on each level of the cascade to facilitate the constraints of the UAV. That is;

$$\mathbf{x}^{(4)} = \mathbf{u} \quad (3.71)$$

$$\tau_1 = \text{sat}(k_1(\mathbf{x}^d - \mathbf{x}), v_{\max}) \quad (3.72)$$

$$\tau_2 = \text{sat}(k_2(\tau_1 - \mathbf{x}^{(1)}), a_{\max}) \quad (3.73)$$

$$\tau_3 = \text{sat}(k_3(\tau_2 - \mathbf{x}^{(2)}), j_{\max}) \quad (3.74)$$

$$\mathbf{u} = k_4(\tau_3 - \mathbf{x}^{(3)}) \quad (3.75)$$

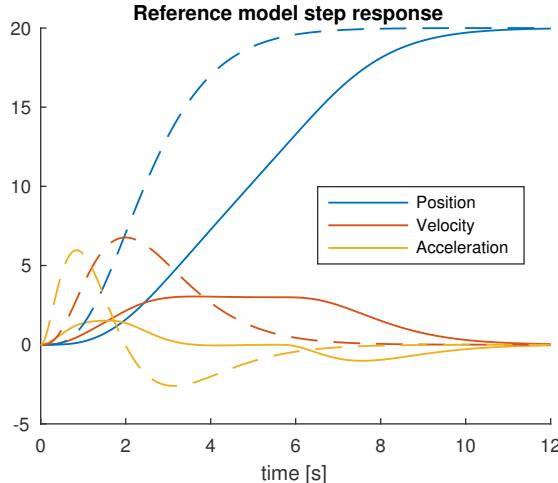
### 3. Single Multirotor UAV with Suspended Load

---

The parameters  $k_i, i \in \{1, \dots, 4\}$  are found by inspection of (3.70) as

$$\begin{aligned} k_4 &= 4\zeta\omega_0 & k_3 &= \frac{(2 + 4\zeta^2)\omega_0^2}{k_4} \\ k_2 &= \frac{4\zeta\omega_0^3}{k_4 k_3} & k_1 &= \frac{\omega_0^4}{k_4 k_3 k_2} \end{aligned}$$

We are now able to impose constraints on the motion by setting  $v_{max}$ ,  $a_{max}$  and  $j_{max}$  appropriately. Consider again the first dimension, with the aforementioned constraints set 3 m/s, 1.5 ms<sup>2</sup> and 3 m/s<sup>3</sup> respectively, the resulting trajectory can be seen by the solid lines in Figure 3.4. Clearly, we see that the response is slower, but complies the constraints. This reference-model is now used to generate feasible, smooth  $C^3$  trajectories for the controller.



**Figure 3.4:** Step response of the reference model. The dashed lines corresponds to a regular low-pass filter, while the solid lines represents the response of the augmented reference model, which complies with the dynamical constraints of the UAV.

## 3.6 Experimental Validation of the Controller

This section presents the results of the performance of the controller structure derived above. The results are verified in a numerical simulation and in a UAV experiment with the equipment as described in Chapter 2, with a step-like trajectory.

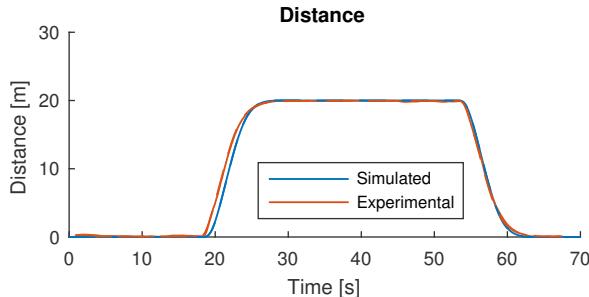
The simulation parameters are set to match that of the experimental platform, and are summarized in Table 3.1. The simulation is conducted in MATLAB, using the model presented Section 3.3. Integration is performed using *Runge-Kutta-4* at 20 Hz.

**Table 3.1:** UAV Data

UAV Mass	2.5	kg
Payload Mass	250	g
Suspension Length	4.2	m
Speed Setpoint	4	m/s
Maximum Acceleration	7	m/s <sup>2</sup>

### 3.6.1 Step-like Trajectory

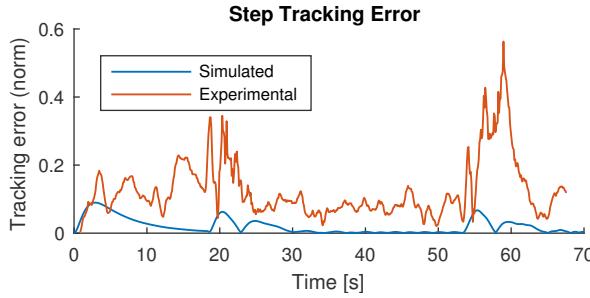
In this validation case, the desired trajectory is a 20 m displacement in the horizontal plane. In both the simulation and experiment, the controller needs to compensate for an unknown wind disturbance. Figure 3.5 illustrates the distance traveled during the test, while Figure 3.6 shows the tracking-error of the controller in both the experimental and simulated setting. As can be seen, the controller tracks the desired trajectory satisfactorily, where in the experimental case the average tracking error is 12.5 cm. There are peaks approaching 60 cm however, which corresponds to the time when the UAV stops at the final position. The wind disturbance estimation magnitude is illustrated in Figure 3.7, which we see for the simulation case that the controller approaches the correct steady-state value. In the experimental case, we see fluctuations, which is induced by wind gusts and the elevated tracking error during aggressive parts of the maneuver. As Figure 3.8 shows, there is naturally a substantial load deflection during acceleration. Especially for the simulated case, we see that the load continues to oscillate when the UAV reaches hover state. Due to un-modeled effects the response in the experiments are of lower amplitude, but the oscillations are clearly visible. In the next section, we discuss how to design the reference trajectory as to minimize the suspended load swing.



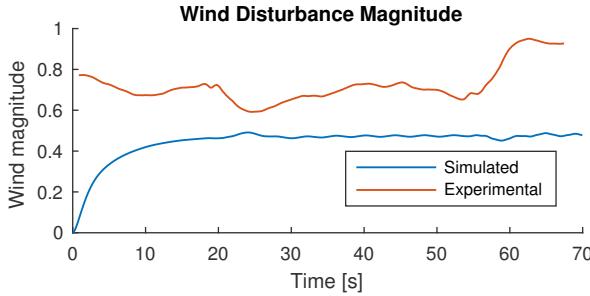
**Figure 3.5:** The distance of the trajectory during the step test case. The deviation from the desired trajectory can be seen in Figure 3.6.

## 3.7 Suppression of Suspended Load Swing

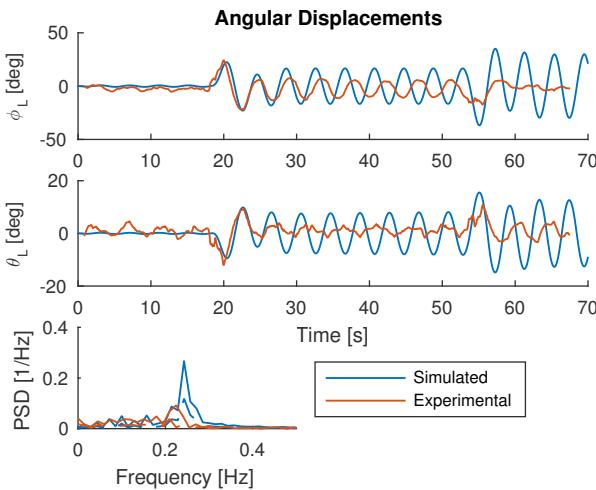
To safely transport a suspended payload with an UAV, the swing motion of the payload should be kept small. This type of problem has a vast history in the literature, especially for usage on overhead cranes [1], but also on various flying



**Figure 3.6:** Tracking error when running the controller with a step-input. The desired position is 20 m away from the initial position.



**Figure 3.7:** Magnitude of the wind disturbance estimation during the step test case. In the simulated case, the magnitude trends towards the unknown constant bias. For the experimental case, the wind estimate fluctuates significantly more due to the shifting wind conditions during the experiment.



**Figure 3.8:** Angles of the suspended load running the step type trajectory. The lower-left plot illustrates the *Power Spectral Density* over the frequency range of pendulum swing.

vehicles. The use of Input Shaping is one of the most practical open-loop control strategies for these problems, and has been widely used [118]. Its main concept is that by giving the UAV two consecutive step inputs, the oscillatory response between the two cancels each other out, and resulting in a oscillation free maneuver. However, it as it is open loop, it is sensitive to modeling uncertainties.

The impulse response of the simplest input-shaping filter  $\mathcal{I}$  can be seen in Figure 3.9 as two pulses at times  $t_1 = 0$  and  $t_2$ , of amplitudes  $A_1$  and  $A_2$ , respectively. Given the damped natural frequency  $\omega_d$  and damping ratio  $\zeta$ , the filter coefficients can be calculated as [16]:

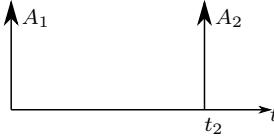
$$t_1 = 0 \quad t_2 = \frac{\pi}{\omega_d} \quad (3.76)$$

$$A_1 = \frac{1}{1 + K} \quad A_2 = \frac{K}{1 + K} \quad (3.77)$$

$$K = \exp\left(\frac{-\zeta\pi}{\sqrt{1 - \zeta^2}}\right) \quad (3.78)$$

For a suspended load, the natural frequency of oscillations is given by  $\omega_n = \sqrt{g/L}$ . Given a reference trajectory  $\bar{\xi}(t) := [\mathbf{r}(t), \dot{\mathbf{r}}(t), \ddot{\mathbf{r}}(t)]$ , the resulting desired trajectory is obtained by the convolution of the shape filter:

$$\xi_I = \bar{\xi} * \mathcal{I} \quad (3.79)$$



**Figure 3.9:** Impulse response of the input shape filter  $\mathcal{I}$ , tuned after the frequency response of the pendulum-motion of the suspended load.

In Figure 3.10, an illustrative example shows the effect of shaping the reference trajectory in this matter. As can be seen, the response is lagging behind the original trajectory, but the residual swings are greatly reduced.

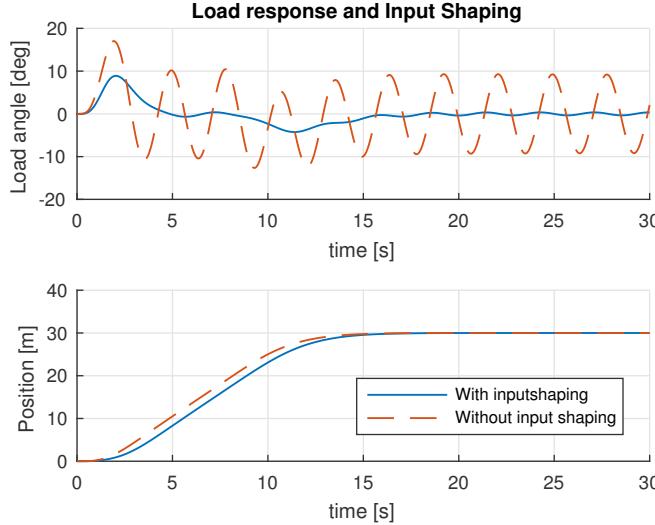
Another technique would be to use feedback from a measured load angle as means to damp the oscillatory motion of the suspended load. In this article, we are utilizing an approach called *Delayed feedback control* [18] to actively damp the swing motion. As illustrated in [18], it is capable of damping out the residual swing during hover or steady transit. The technique is based on adding a feedback-component from the measured deflection angle on the reference signal sent to the controller. Consider the one dimensional auxiliary reference signal

$$x_r(t) = G_d L \sin(\theta_L(t - \tau_d)) \quad (3.80)$$

where  $G_d$  and  $\tau_d$  are design-parameters, designed as to dampen out the swing load. As can be seen, the added reference signal purposefully induces a *delay* in the feedback loop to achieve the damping. While the readers are referred to [18] for

### 3. Single Multirotor UAV with Suspended Load

---



**Figure 3.10:** Illustrative example of the effects of open-loop trajectory generation using input shaping.

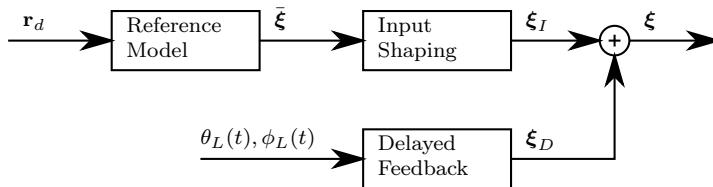
details on the design procedure, for the multicopter UAV we utilize the following parameters:

$$G_d = 0.325, \tau_d = 0.325 \frac{2\pi}{\omega_d} \quad (3.81)$$

To utilize the delayed feedback controller, consider the  $y$ -axis equivalent of (3.80),  $y_r(t) = -G_d L \sin(\phi_L(t - \tau_d))$ , and aggregate the first and second derivatives to

$$\xi_D = [x_r \quad y_r \quad 0 \quad \dot{x}_r \quad \dot{y}_r \quad 0 \quad \ddot{x}_r \quad \ddot{y}_r \quad 0] \quad (3.82)$$

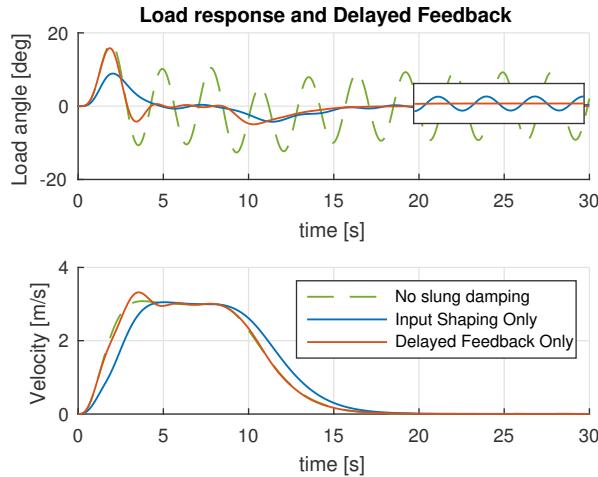
The structure of the total feed forward plus delayed feedback reference generator can then be seen in Figure 3.11. This is a similar structure as presented in [18].



**Figure 3.11:** Traditional overall structure for combining feed-forward input shaping with delayed feedback for trajectory generation.

An example showcasing the delayed feedback controller (compared with the input-shaping feed forward) is seen in Figure 3.12. As can be seen, the feedback

controller is removing all of the residual oscillations. However, during start and stop, it produces unnecessary large oscillations in the load angle and resulting UAV velocity, as can be seen by the peak in Figure 3.12.



**Figure 3.12:** The delayed feedback controller cancels out the residual swing, but produces unwanted peaks in reference velocity.

As can be deducted from this, the feedforward shaping filter does a good job during the transients, but are susceptible to modeling errors and residual swings. The feedback term on the other hand, is overreacting during the transients but does a good job at canceling the lingering oscillations. The direct combination of the two, as illustrated in Figure 3.11, does a fairly decent job at maintaining the benefits of both terms. But, there is still some unwanted velocity peaks and overreacting from the feedback structure during the transients. This is kind of natural, as the job of the feedforward is to allow an initial swing, but then cancel it by the second impulse. By having the feedback term active all of the time, it renders parts of the feedforward terms more inaccurate. Thus, we would like to smoothly disable the feedback term during the transients.

By examining the full dynamic model in (3.29), it can be observed that the UAV only influence the suspended load motion through acceleration. Thus, we propose to introduce a smooth gain-scheduling approach for the delayed feedback, using the recent history of reference acceleration as input. Specifically, let

$$\mathbf{a}_k := [a_k \ a_{k-1} \ a_{k-2} \ \cdots \ a_{k-N}] \quad (3.83)$$

where  $a_k := ||\ddot{r}(t_k)|| \in \mathbb{R}$  is the total reference acceleration at time  $t_k$ , for some constant number of samples  $N$ . Further, let  $p_k$  be the *percentage of values in  $\mathbf{a}_k$  less than a acceleration threshold  $a_t$* . That is;

$$p_k = \frac{1}{N} \sum_{i=0}^N \begin{cases} 1 & \text{if } a_{k-i} < a_t \\ 0 & \text{else} \end{cases} \quad (3.84)$$

### 3. Single Multirotor UAV with Suspended Load

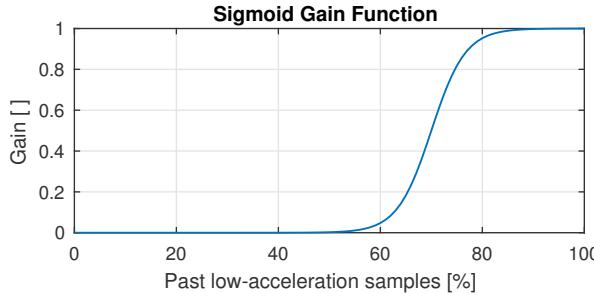
---

Let  $p_k$  be the input to a sigmoid-type function. We utilize the *logistic function*

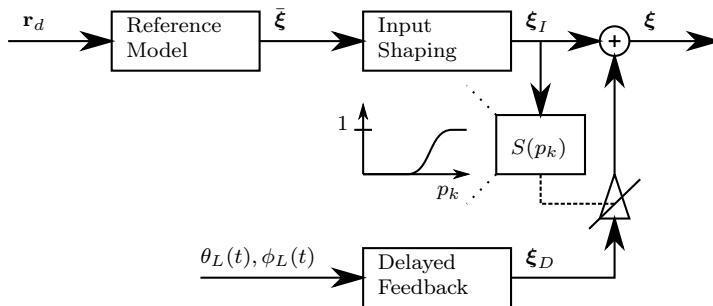
$$S(p_k) := \frac{1}{1 + e^{-k(p_k - p_c)}} \quad (3.85)$$

where  $k, p_c \in \mathbb{R}$  can be tuned to vary the shape of  $S(p_k)$ . Figure 3.13 shows such a sigmoid function for  $k = 30, p_c = 0.7$ . Subsequently, we get the structure depicted in Figure 3.14 where we can see that the sigmoid gain block  $S(p_k)$  adjusts the contribution from the delayed feedback based on the reference acceleration of the unperturbed trajectory  $\xi_I$ . The effect of this scheme can be seen in Figure 3.15, where we can see that the residual oscillations are cancelled out and the UAV avoids the unnecessary velocity peaks.

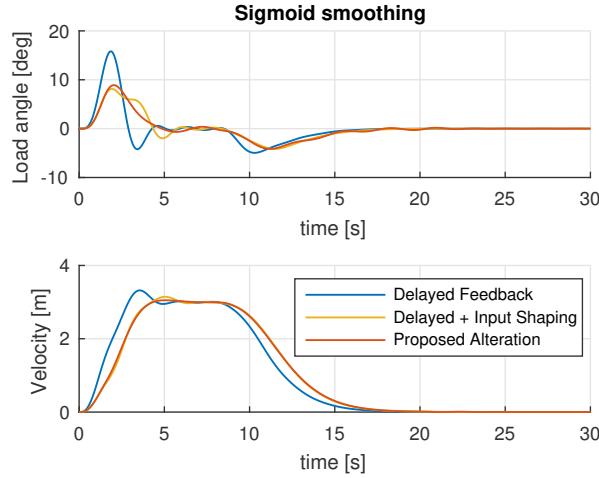
The complete interconnected slung-load controller is now depicted in Figure 3.16.



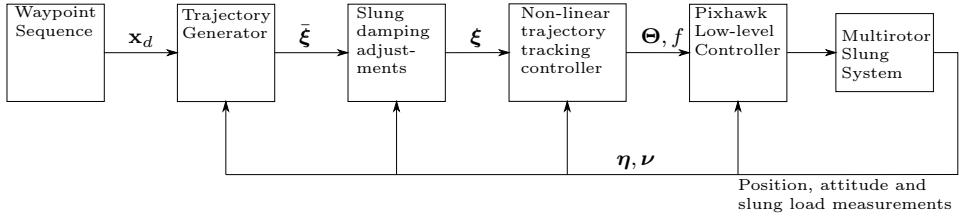
**Figure 3.13:** Illustration of the sigmoid function. By looking at the history of reference accelerations, the gain is increased when we are at steady state for maneuvering.



**Figure 3.14:** Proposed structure for combining the feed forward input shaper with delayed feedback. Using the current and recent reference accelerations from  $\xi_I$ , the block  $S(p_k)$  smoothly activates the feedback path only on constant velocity part of a maneuver (including hover).



**Figure 3.15:** The result when using the proposed sigmoid gain function

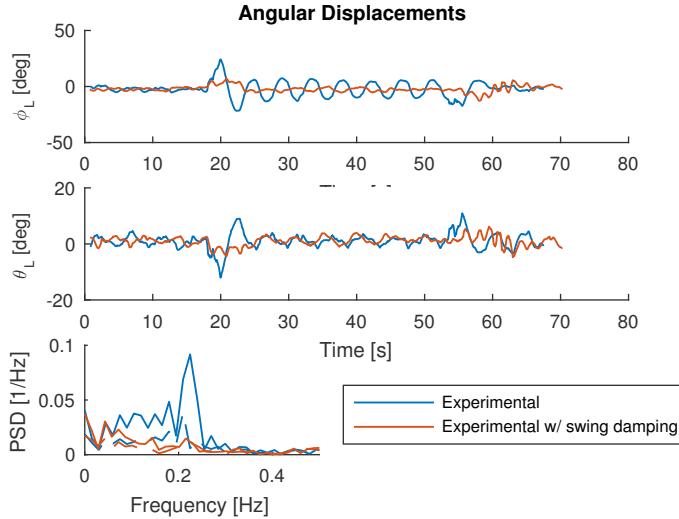


**Figure 3.16:** The complete controller structure, illustrating the signals flowing through the various main components. The physical system, including state estimation and measurement of the slung load deflection angles, is represented in the block *Multirotor Slung System*.

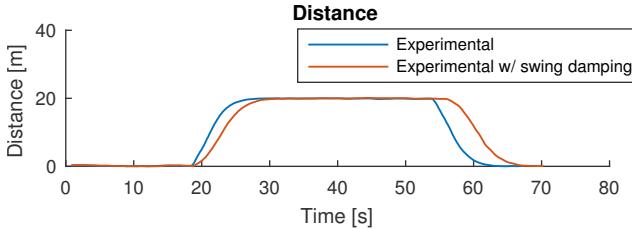
## 3.8 Experimental Validation of Controller with Swing Damping

### 3.8.1 Step-like Trajectory

To verify the performance of the tracking controller with the proposed swing-damping technique, we first do the same step-maneuver as in Section 3.6. Figure 3.17 shows the experimental results with and without swing damping. Clearly, there is substantially less angular motion of the suspended load with the swing damping enabled. This is also visible from the power spectral density in the lower-left of Figure 3.17, where we see that the natural frequencies are now less dominant than secondary oscillatory effects from the UAV motion. Figure 3.18 shows the lag-effect of the input shaper, where the second experiment lags a bit behind the first without the swing damping enabled. For most applications, this is a viable tradeoff when taking the pronounced benefit of the reduced load motion into account.



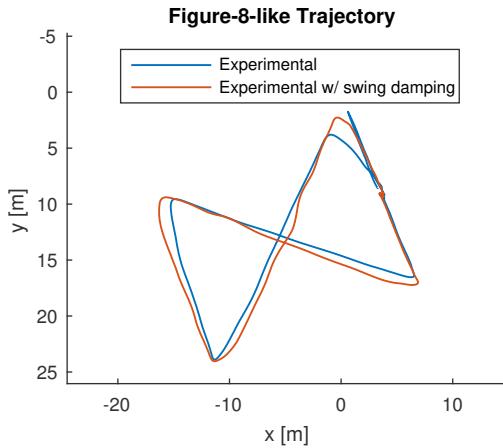
**Figure 3.17:** Angular response during the experiment, running a step-like input. With the swing damping enabled, the oscillations are of lower amplitude and are damped faster. The lower-left plot shows the *power spectral density* of the angular signal.



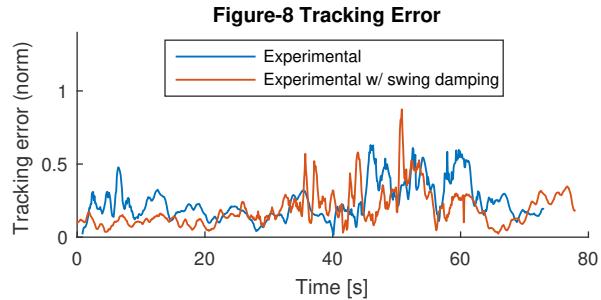
**Figure 3.18:** Distance travelled during the step-like input.

### 3.8.2 Figure-8-like Trajectory

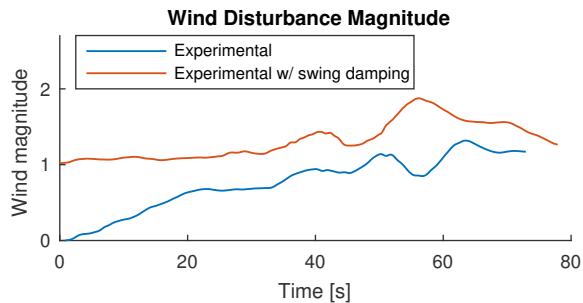
Further, we tested the performance for a more complicated figure-8 like trajectory which we compare with a similar trajectory without any swing damping. For both cases, the trajectory includes a 20 second hover state at the beginning the maneuver. The resulting North-East path can be seen in Figure 3.19. From Figure 3.20, it can be seen that the tracking error is of the same magnitude as without the swing damping, but looking at the angular displacements in Figure 3.22, the difference is more striking. The suggested structure is able to greatly reduce the angular displacements. The wind disturbance estimation can be seen in Figure 3.21, which in both cases varies over the course of the experiment. This is partly due to gusts, but it is also reacting to the somewhat higher average tracking error during the end of the maneuver due to aggressive turning. Also, note that due to the sequence of experiments conducted, the case where the swing damping was enabled had a soft-start of the bias estimation. However, due to the initial hover state, the controller have a reasonable time to reach estimation steady state before the maneuver commences which can be observed in the plot.



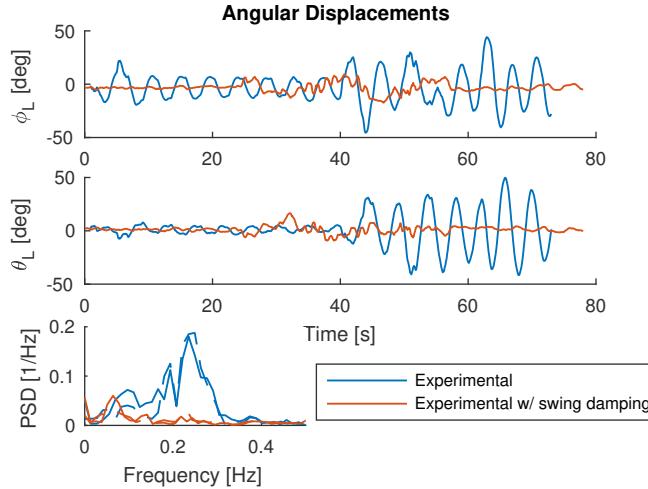
**Figure 3.19:** Trajectories during the figure-8-like maneuver. The discrepancies is due to the altered desired trajectory from the swing-damping.



**Figure 3.20:** Tracking errors along the desired trajectory.



**Figure 3.21:** The estimated wind disturbance magnitude during the figure-8-like maneuver.



**Figure 3.22:** Suspended load displacement angles during the figure-8-like maneuver.

The peak of the *power-spectral-density* graph on the lower left corresponds to the natural oscillation frequency of the undamped load.

### 3.9 Chapter Summary and Conclusions

In this chapter, we have studied a multirotor UAV carrying a suspended load. By utilizing Kane's equation, a nonlinear model of the interconnected dynamics was derived. Further, a trajectory tracking controller based on the backstepping technique was designed. Due to the presence of wind disturbances, the controller includes a wind bias compensator in the form of an added integral effect. The origin of the tracking error was proven UGAS and ULES.

To dampen the deflection angles of the oscillating suspended load, we considered an open-loop and a feedback type trajectory generation approach from the literature. We proposed a new method to combine these two approaches using a gain-scheduling procedure based on recent acceleration reference points.

The controller and proposed swing damping methodology was verified using both numerical simulation and experimental data from a in-house built low-cost UAV platform. All experiments were conducted outside, and implementation was done in a application framework applicable for continuous use and research.

## Chapter 4

# Multibody Operations and Modeling

This chapter works as an introduction to multiple UAVs carrying a suspended payload, which is the topic for Chapters 5–7. The chapter gives an overview of the related work, and gives an introduction to dynamic modeling using the equations of *Udwadia-Kalaba* for constrained multi-body systems. A control design based on measurement of relative load angles instead of global positions is presented in **Chapter 5**, where distributed kinematic control laws are derived to ensure stability of the desired relative UAV formation. Synchronized path-following is introduced and applied to multirotor UAVs carrying a suspended payload in **Chapter 6**, where we consider the multirotor UAVs being able to measure the drag from the load. In **Chapter 7**, a passivity based design procedure is considered to generate cooperative control laws for the multirotor UAVs, where the disturbance from the suspended load is unknown. Chapter 7 also presents experimental results for the proposed controller.

### 4.1 Introduction

As discussed in Chapter 1, the introduction of multiple cooperating UAVs to lift a suspended load can lead to increased payload capacity, operational redundancy and a greater sense of control of the suspended load. In addition, it can open up for all new applications such as net-recovery of fixed wing UAVs using multirotors, as presented in Chapter 8. However, such operations create numerous new challenges in relation to modeling and control, which is the topic of this part of the thesis.

Compared to a single multirotor, several multirotors with a common lifting arrangement (e.g. lines) provides significantly more stability and some added flexibility to position the spatially distributed lifting arrangement in space and attitude. For example, for rendezvous with objects in order to recover and move them, the end effector could be more stably positioned (less swinging due to e.g. winds). Rather than a load at the end of a single line, a load hanging in multiple lines will be more stable and provide more accuracy in e.g. geo-survey applications where the load could include ground-penetrating radar or sensitive instruments to measure magnetic field or gravity to be moved accurately as close to the ground as possible.

For control synthesis and simulation purposes, a sufficiently elaborate dynamic

model of the interconnected UAV-slung load is needed. In the literature, this is called *multi-body*-systems, and several methodologies exists for the dynamic modeling of such systems.

There is a lot of nomenclature surrounding operations and control with multiple vehicles. The following presents the main terms, as used in this thesis and in the literature (e.g. [10, 92, 110]).

- *Coordinated*: Generally considered an umbrella-term for control systems involving multiple agents.
- *Centralized*: Multi-agent systems where a single control law sets the desired location or trajectory of all the agents. Typically, the control law runs on a single computer. The opposite is *decentralized* control, which also can be *distributed*, where each vehicle calculates its own control output based on information from the other agents in the group.
- *Leader-Follower*: One of the vehicles, named the *leader*, uses all available information to set desired setpoints or trajectories to the rest of the vehicles, commonly called *follower* or *slaves*.
- *Cooperative*: Also called *synchronized* control, occurs when all agents in the group tries to achieve the goal simultaneously.

Thus, for practical applications, a distributed cooperative control strategy is often preferred.

#### 4.1.1 Organization of this Chapter

An overview of related work concerning suspended load transportation using multiple UAVs is given in Section 4.2. Next, a brief overview of modeling strategies for complex inter-connected multibody systems is given in Section 4.3, using both *constrained* and *unconstrained* generalized coordinates. Further, the equations of Udwadia-Kalaba is introduced as a suitable tool for generating equations of motion for the coordinated multirotor lift system in Section 4.3.1, intended to be used in a simulation environment. These equations are derived in Section 4.4, using the model of the single multirotor dynamics introduced in Chapter 3.

## 4.2 Related Work

Next, we give an overview on related work on multi-body modeling (where multiple bodies are connected to the same suspended load) and control of cooperative UAVs . In regard to multi-body modeling, the work in [22] provides comprehensive models for several popular attachment configurations. The works includes both elastic and non-elastic cables. Although comprehensive, the methodology is elaborate and hard to generalize for different configurations. See also [127] and references therein.

In [13], Kane's method [59] is utilized to systematically generate equations of motion for  $n$  helicopter UAVs rigidly attached to a suspended payload. The equations are used for simulation and generation of reference trajectories for multi-lift operations. [17] provides a very generalized methodology to model suspended load systems using the equations of Udwadia-Kalaba [137]. This time using *constrained*

coordinates, it allows for a systematic procedure to generate equations for simulation purposes. The work includes the collapse and tensioning of wires, which is neglected in most other references. In addition, the models are experimentally verified on a small-scale helicopter using computer vision to capture the position of the suspended payload.

Experiments on multi-body suspended load systems are presented in [14], where the authors use three small-size helicopters to transport a video-capture device to aid in a simulated search & rescue operation. This work presented a first on such multi-lift operations using UAVs, and more details on the control system and operational aspects is given in [13, 83]. For the coordinated part of the control design, the authors use a centralized scheme, were reference trajectories for each helicopter UAV is generated from a model of the interconnected multi-body system. The additional drag and induced torque due to moment arms on the attachment point is counteracted by feed-forwarding a measurement of the forces along the suspension wire.

*Geometric* control theory has been applied to the multiple multirotor-slung load transportation problem in [44, 73–76]. The authors model the interconnected wires as segmented flexible links, and represents all the dynamics directly on a non-linear manifold in a singularity-free representation. Extensive stability analysis is performed, and some experimental validation is performed on an indoor setup in [45].

Control of attitude and position of a suspended load using multirotors is considered in [35, 86], which also provides experimental validation using external cameras as feedback. In [85], multirotors are equipped with gripping mechanisms to rigidly attach themself to a load, and cooperative control laws for load transportation is developed and verified experimentally on an indoor platform. Coordinated multirotor lifting operations has also been studied in [29, 58, 77, 131].

More generally speaking, coordinated control of multiple agents, often also called *consensus* problems, also has a vast literature. As a very brief overview; The usage of graphs to model communication networks as a basis for distributed control was discussed [34], which derives stability criterions for linear systems. [110] provides a good survey of developments up until early 2000s. For coordinated control, robustness with respect to time delays and link failures are oft discussed, see for instance [92] and references therein. A passivity-based design framework allows for systematic synchronization of nonlinear systems is introduced in [8, 10]. Synchronized path-following is the topic of [2, 55], in which multiple maneuvering controllers is coordinated to achieve formation along a pre-defined path. [55] also discusses robustness with respect to output saturations. Event-based control strategies are discussed in [26, 27], which is especially suited for distributed implementation on sensor systems with low communication and processing capacity. A notable formation control methodology applied to marine vehicles using *Lagrangian multipliers* is presented in [54, 102].

For UAVs carrying a suspended load, of the experimental tests conducted in [14, 35, 45, 85, 86], only [14] is performed outdoors with a UAV size suitable for real-life operations and verified with environmental disturbances.

### 4.3 Multi-Body Modeling

The goal of this section is to model a system consisting of  $m$  UAVs (multirotors), connected with  $k$  links to a suspended payload. This is a generic modeling problem and can be handled in numerous different ways. Each rigid body (the multirotors plus the suspended load) separately has 6 degrees of freedom (DOF). The interconnected constrained system thus has  $6(m + 1) - k$  independent generalized coordinates. If we were to describe the system using these, the equations of motion can be derived using e.g. Kane's method (as in [13], described in [28] or [5]), or by using Lagrange's equation by the Lagrangian function ([28], [42]). These independent generalized coordinates can be difficult to find when the number of constraints are high, and the resulting equations of motion is usually derived by symbolic software such as MotionGenesis [87] (formerly Autolev).

If the generalized coordinates are chosen as dependent coordinates (also called constrained generalized coordinates [42]), the constraint forces of the system (traditionally through the use of Lagrangian multipliers) must be calculated for each step so the constraints are not violated. Various solutions to this exists, which typically involves solving Differential-Algebraic Equations (DAEs) or other *augmented* methods [42, Sec 8.2].

Another way of dealing with constraints on dependent generalized coordinates was proposed by Udwadia and Kalaba [137]. It bears some resemblance to other embedded methods (like [48], as presented in [42, pp. 519]), in that it uses the equations for the constrained acceleration to directly compute the constraint forces.

For the case of suspended payload discussed here, a benefit of working with dependent coordinates is that it is easy to incorporate wire collapse to the model. This can be done by simply removing that constraint from the equation when some conditions are met, as can be seen in [17]. In addition, it allows one to create modular simulators, that can add or remove bodies without needing to re-generate equations of motion. A downside is that analytical expressions in the form  $\ddot{q}_i = f(\mathbf{q}, \dot{\mathbf{q}}, t)$ , where  $\mathbf{q} = [q_1, \dots, q_n]^\top$  are the generalized coordinates, are generally more complex than those for independent coordinates, and are less suitable for control synthesis.

#### 4.3.1 The Udwadia-Kalaba equation

This section follows the derivation in [15]. Consider the unconstrained Newtonian system

$$\mathbf{M}\ddot{\mathbf{q}}_u = \mathbf{Q} \quad (4.1)$$

where  $\mathbf{q}_u \in \mathbb{R}^n$  is the unconstrained generalized coordinates of the system, and  $\mathbf{Q} \in \mathbb{R}^n$  are generalized forces. Now, let the system be subjected to  $p$  constraints in the form

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (4.2)$$

where  $\mathbf{A} \in \mathbb{R}^{p \times n}$  and  $\mathbf{b} \in \mathbb{R}^p$  and  $\mathbf{q} \in \mathbb{R}^n$  the generalized coordinates of the constrained motion of (4.1).

The system (4.1) can be transformed to a constrained system by augmenting it with a constraint force  $\mathbf{Q}_c \in \mathbb{R}^n$  as:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q} + \mathbf{Q}_c \quad (4.3)$$

In [137], (4.3) is solved by applying Gauss's principle of Least Constraints. This principle states that the acceleration of the constrained system follows the vector closest to the unconstrained acceleration, that satisfies the constraints. This leads to a minimization problem, which can be solved using the Moore-Penrose pseudoinverse. Moreover, the constrained system's acceleration  $\ddot{\mathbf{q}}$  can be found from (see [15] and [137]):

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_u + \mathbf{M}^{-1/2}(\mathbf{A}\mathbf{M}^{-1/2})^+(\mathbf{b} - \mathbf{A}\ddot{\mathbf{q}}_u) \quad (4.4)$$

where  $(\cdot)^+$  denotes the Moore-Penrose pseudoinverse. This makes it possible to explicitly identify the constraint force  $\mathbf{Q}_c$  as

$$\mathbf{Q}_c = \mathbf{M}^{1/2}(\mathbf{A}\mathbf{M}^{-1/2})^+(\mathbf{b} - \mathbf{A}\ddot{\mathbf{q}}_u) \quad (4.5)$$

The complete derivation (along with proofs) can be seen in [138], where the author also derives the Lagrange- and Gibbs-Appel equations from (4.4).

### 4.3.2 Illustrative Example

To highlight some of the differences of the above approach to some of the more conventional ones, an illustrative example is provided in this section.

Consider a pendulum of mass  $m$ , rigidly attached to a point, as illustrated in Figure 4.1. To model this, one must first describe the configuration of the system in terms of a choice of coordinates. The first obvious choice is  $\theta$ , which uniquely describes the position of the pendulum. Thus,  $q = \theta$ , and  $q$  then are the independent coordinates of this 1 DOF system. But, we could also choose to describe the system with the variables  $(x, y)$ , which describes the position of the point-mass in cartesian coordinates. But, the point-mass is not free to move about on all values of  $(x, y)$ , it is restricted to be on the circle  $x^2 + y^2 = L^2$ . So, if we chose  $\mathbf{q} = [x, y]^\top$  as our generalized coordinates, we describe our system with 2 coordinates. Since our system has 1DOF, we know that 1 constraint is also present. Such generalized coordinates are thus called *constrained* generalized coordinates, or *dependent* generalized coordinates.

In the following, we illustrate how the dynamics of the pendulum can be modeled using different choices for generalized coordinates, and modeling approaches.

#### Lagrangian Energy formulation - Independent Coordinates

The Lagrangian (kinetic minus potential) energy function of our system is

$$L = \frac{1}{2}mL^2\dot{\theta}^2 - mgL(1 - \cos\theta) \quad (4.6)$$

and the equation of motion are readily found as

$$\ddot{\theta} + \frac{g}{L} \sin\theta = 0 \quad (4.7)$$

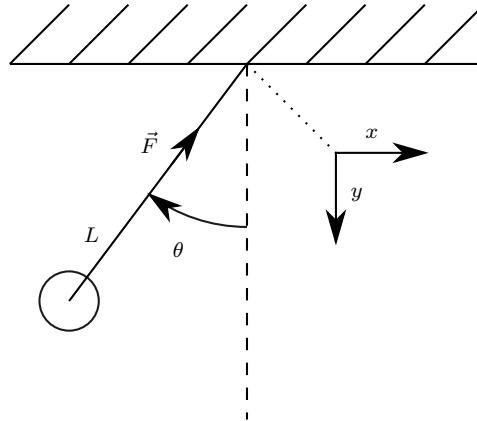


Figure 4.1: Pendulum

### Lagrangian Energy formulation - Constrained Coordinates

Consider now the choice  $\mathbf{q} = [x, y]^\top$ . The constraint are given by

$$x^2 + y^2 = L^2 \quad (4.8)$$

and it can be differentiated to obtain

$$x\dot{x} + y\dot{y} = 0 \quad (4.9)$$

or it's pfaffian form:

$$xdx + ydy = 0 \quad (4.10)$$

In lagrangian mechanics, it is typical to formulate the constraints in the following manner [42]:

$$\sum_{j=1}^N a_{ij}(\mathbf{q}, t)\dot{q}_j + b_i(\mathbf{q}, t) = 0 \quad (4.11)$$

where \$N\$ is the number of generalized coordinates, and \$i\$ is the considered constraint. In our problem, we have a single constraint and \$N = 2\$, which gives

$$a_{11} = a_x = x \quad (4.12)$$

$$a_{12} = a_y = y \quad (4.13)$$

Using the Lagrangian function

$$L = \frac{1}{2}m\mathbf{q}^\top \mathbf{q} - mg(L - y) \quad (4.14)$$

yields the equations

$$m\ddot{x} = \lambda a_x = \lambda x \quad (4.15)$$

$$m\ddot{y} = mg + \lambda a_y = mg + \lambda y \quad (4.16)$$

where  $\lambda$  is the lagrangian associated with our constraint. In vector-form:

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ mg \end{bmatrix} - \lambda \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.17)$$

The identification of  $\lambda$  is in general no trivial task. In this case, we can write down the virtual work done by the force  $F$  as

$$\delta W = F \frac{x}{L} \delta x + F \frac{y}{L} \delta y \quad (4.18)$$

By [42], this lets us identify  $\lambda$  as  $\lambda = F/L$ . But the force  $F$  is still unknown.

### Udwadia-Kalaba equation - Constrained Coordinates

Again, we are using  $\mathbf{q} = [x, y]^\top$  as constrained generalized coordinates, subjected to the constraint  $x^2 + y^2 = L^2$ . According to the procedures listed in Section 4.3.1, the constraint may be twice differentiated to yield

$$\underbrace{\begin{bmatrix} x & y \end{bmatrix}}_{:=\mathbf{A}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \underbrace{-\dot{x}^2 - \dot{y}^2}_{:=b} \quad (4.19)$$

which is of the form (4.2).

Then, we write down the equations of motion for the *unconstrained* system, in terms of it's unconstrained acceleration  $\mathbf{a}$ . Since gravity is the only force applied, this is simply found as

$$\underbrace{\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}}_{:=\mathbf{M}} \mathbf{a} = \begin{bmatrix} 0 \\ mg \end{bmatrix} \quad (4.20)$$

which gives  $\mathbf{a} = [0, g]^\top$ . The Udwadia-Kalaba equation now gives

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ g \end{bmatrix} + \mathbf{M}^{-1/2} (\mathbf{A} \mathbf{M}^{-1/2})^+ \left( b - \mathbf{A} \begin{bmatrix} 0 \\ g \end{bmatrix} \right) \quad (4.21)$$

The More-Penrose pseudo inverse of  $(\mathbf{A} \mathbf{M}^{-1/2})$  can be evaluated to

$$(\mathbf{A} \mathbf{M}^{-1/2})^+ = \frac{1}{m^{-1}x^2 + m^{-1}y^2} \begin{bmatrix} m^{-1/2}x \\ m^{-1/2}y \end{bmatrix} \quad (4.22)$$

which after some algebra results in the explicit equation

$$\mathbf{M} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ mg \end{bmatrix} - m \frac{\dot{x}^2 + \dot{y}^2 + gy}{L^2} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.23)$$

By comparing (4.17) with (4.23), we see that  $\lambda$  can be identified as

$$\lambda = -m \frac{\dot{x}^2 + \dot{y}^2 + gy}{L^2} \quad (4.24)$$

where it can be seen that the constraint force opposes the centrifugal + gravitational acceleration subjected to the pendulum.

### 4.3.3 Discussion

In the foregoing sections, we have seen that choosing constrained generalized coordinates gives more complex equations of motion, as expected. Even though this example is simple, and the task of finding and using independent generalized coordinates are trivial, it is apparent that it might not be that easy when you have 4 interconnected bodies, each with 6DOF, subjected to some  $k$  constraints. In addition, using constrained generalized coordinates lets you evaluate the constraint force (regardless of how the the equations are found) [42]. The general form of the equations of motion with langrangians have the form

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{A}(\mathbf{q}, t)^\top \boldsymbol{\lambda} + \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (4.25)$$

which can be put as

$$\begin{bmatrix} \mathbf{M} & -\mathbf{A}^\top \\ -\mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} \quad (4.26)$$

This is called a differential-algebraic equation as mentioned previously, and special solvers exists to solve these. But they are prone to numerical drift and are computationally complex, and have thus not been widely implemented by dynamics researchers [42]. Refer to [20] for a comprehensive discussion on the topic. The equations are often solved by *augmented* or *embedded* methods as described in [42] and [5].

The equations of Udwadia-Kalaba utilizes the More-Penrose pseudo inverse to embed the solutions of each  $\lambda$  into the equations of motion. This is nothing new, and by asserting certain conditions on the constraints (such as linear independence), other general inverses can be used [42, pp. 519]. However, no such conditions (other than feasibility) of the constraint are needed when using Udwadia-Kalaba. The downside is that the solution of the pseudo inverse is often hard to find by hand. But, there are numerous numerical solvers for this, and most symbolic mathematical software (such as MATLAB) has the capability to give analytical results. Further investigations into the numerical burden of the pseudo-inverse can be seen in [31] and [30]. The latter reference also includes some studies where students are asked which modeling techniques they preferred for various situations. The results showed that the students found the algebraic burden complex for hand-solving the equations of motion using Udwadia-Kalaba, but the numerical implementation was much easier than other methods.

## 4.4 Equations of Motion for Multi-Lift Operations

The goal of this section is to derive the equations of motion for our system by the use of Udwadia-Kalaba's equation (4.4). This section follows a similar procedure as given in [17]. We begin by defining our generalized coordinates. For each body  $b$  (multirotor + payload), let  $\mathbf{p}_b^n \in \mathbb{R}^3$  represent the position of that body in NED, and  $\boldsymbol{\Theta}_b$  the orientation of that body, given in euler angles. The notation  $(\cdot)_{ji}^n$  means that  $(\cdot)$  concerns multirotor  $j$  and wire  $i$ , represented in the frame  $\{n\}$ .  $(\cdot)_l^n$  (subscript  $l$ ) represents the load. We have  $k$  multirotors. The generalized coordinates are thus:

$$\mathbf{q} = [\mathbf{p}_1^n \ \Theta_1 \ \dots \ \mathbf{p}_k^n \ \Theta_k \ \mathbf{p}_l \ \Theta_l]^\top \quad (4.27)$$

We assume that we have the model for each separate (unconstrained) body. This is given in Section 3.2 as

$$\mathbf{M}_b \dot{\nu}_b + \mathbf{C}\nu = \boldsymbol{\tau}_{RB} \quad (4.28)$$

But,  $\dot{\nu}_b$  cannot be used as a generalized acceleration, as it is differentiated in its body-frame. But, we know that the equation of motion can be written

$$\mathbf{M}_b \begin{bmatrix} \mathbf{a}_b^b \\ \boldsymbol{\alpha}_b^b \end{bmatrix} = \boldsymbol{\tau}_{RB} \quad (4.29)$$

where  $\mathbf{a}_b^b$  are the acceleration represented in the body-fixed frame (specific force), and  $\boldsymbol{\alpha}_b^b$  is specific torque. This can be used as generalized acceleration.

The next step is to create the constraint equation, as given in (4.2).

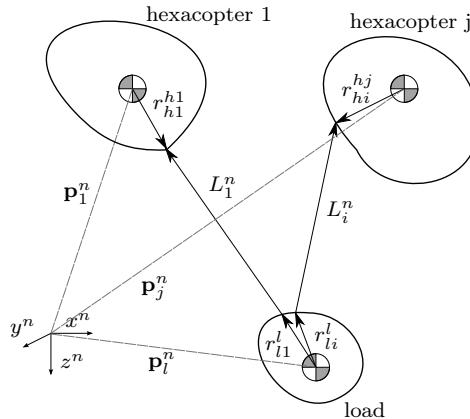
#### 4.4.1 Constraint equation

In this section, the constraint equations for our system will be derived. Let  $i$  represent a wire. The notation  $(\cdot)_{ji}^n$  means that  $(\cdot)$  concerns multirotor  $j$  and wire  $i$ , represented in the frame  $\{n\}$ .  $r_{ji}^j$  is the attachment point of wire  $i$  at multirotor  $j$ , represented in multirotor  $j$ 's body fixed frame.

In the following, let  $k$  multicopters be connected to a single suspended load with a total of  $p$  wires. For each wire  $i$ ,  $j$  represents the multicopter connected through wire  $i$  to the payload (denoted  $l$ ).

$$j(i) : \{1, \dots, p\} \rightarrow \{1, \dots, k\} \quad (4.30)$$

Further, let it be implicit that  $j := j(i)$ .



**Figure 4.2:** Illustration of the system. Several multirotors, located at  $p_j^n$ , are connected to a suspended load. The attachment-points are given as  $r_b^b$ . *Inspired by similar illustrations in [17].*

The constraints are on the length of each wire. Let  $\mathbf{L}_i$  be the wire connecting the payload and a multirotor. See Figure 4.2. The wire-vector  $\mathbf{L}_i^n$  are given by:

$$\mathbf{L}_i^n = \mathbf{p}_{1_{hj}} + \mathbf{r}_{hi}^n - \mathbf{p}_l - \mathbf{r}_{li}^n \quad (4.31)$$

which can be written as

$$\mathbf{L}_i^n = \mathbf{p}_{1_{hj}} + \mathbf{R}_{hj}^n \mathbf{r}_{hi}^{hj} - \mathbf{p}_l - \mathbf{R}_l^n \mathbf{r}_l^l \quad (4.32)$$

$\mathbf{L}_i^n$  is now a function of the position of the involved multirotor ( $\mathbf{p}_{1_{hj}}$ ), the position of it's attachment point in it's body frame ( $\mathbf{r}_{hi}^{hj}$ ), and likewise with the load ( $\mathbf{p}_l$  and  $\mathbf{r}_l^l$ ).

Since the constraint concerns the wire length and distance between the connected objects, let the constraints concern the norm of  $\mathbf{L}_i$ :

$$g_i = \|\mathbf{L}_i^n\|^2 - d_i^2 = (\mathbf{L}_i^n)^\top \mathbf{L}_i^n - d_i^2 = 0 \quad (4.33)$$

where  $d_i$  is the length of wire  $i$ . Now,  $g_i$  can be differentiated twice to achieve:

$$\dot{g}_i = \frac{n}{dt} g_i = 2\dot{\mathbf{L}}_i^\top \mathbf{L}_i = 0 \quad (4.34)$$

$$\ddot{g}_i = 2\ddot{\mathbf{L}}_i^\top \mathbf{L}_i + 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i = 0 \quad (4.35)$$

The derivatives of  $\mathbf{L}_i$  can be found by:

$$\dot{\mathbf{L}}_i = \dot{\mathbf{p}}_{hj} + \dot{\mathbf{R}}_{hj}^n \mathbf{r}_{hi}^{hj} - \dot{\mathbf{p}}_l - \dot{\mathbf{R}}_l^n \mathbf{r}_l^l \quad (4.36)$$

$$\ddot{\mathbf{L}}_i = \ddot{\mathbf{p}}_{hj} + \ddot{\mathbf{R}}_{hj}^n \mathbf{r}_{hi}^{hj} - \ddot{\mathbf{p}}_l - \ddot{\mathbf{R}}_l^n \mathbf{r}_l^l \quad (4.37)$$

The first and second derivative of  $\mathbf{R}_b^n \mathbf{r}_b^b$  can be found as:

$$\dot{\mathbf{R}}_b^n \mathbf{r}_b^b = \frac{n}{dt} \mathbf{R}_b^n \mathbf{r}_b^b = \mathbf{R}_b^n \mathbf{S}(\omega_{b/n}^b) \mathbf{r}_b^b = -\mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \omega_{b/n}^b \quad (4.38)$$

$$\ddot{\mathbf{R}}_b^n \mathbf{r}_b^b = -\dot{\mathbf{R}}_b^n \mathbf{S}(\mathbf{r}_b^b) \omega_{b/n}^b - \mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \frac{n}{dt} \omega_{b/n}^b \quad (4.39)$$

According to [28],

$$\frac{n}{dt} \omega_{b/n}^b = \frac{b}{dt} \omega_{b/n}^b \quad (4.40)$$

Thus,

$$\ddot{\mathbf{R}}_b^n \mathbf{r}_b^b = \mathbf{R}_b^n \mathbf{S}(\omega_{b/n}^b) \mathbf{S}(\omega_{b/n}^b) \mathbf{r}_b^b - \mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \dot{\omega}_{b/n}^b \quad (4.41)$$

This will make  $\mathbf{L}_i$  a function of  $\ddot{\mathbf{p}}_b^n$  and  $\dot{\omega}_{b/n}^b$ . This needs to be converted to generalized coordinates, represented in  $\{b\}$ . This can be done by

$$\mathbf{a}_b^b = \mathbf{R}_b^n \ddot{\mathbf{p}}_b^n \quad (4.42)$$

and

$$\dot{\omega}_{b/n}^b = \underbrace{\mathbf{I}_b^{-1} m_{rb}^b}_{:=\alpha_b^b} - \mathbf{I}_b^{-1} \mathbf{S}(\omega_{b/n}^b) \mathbf{I}_b \omega_{b/n}^b \quad (4.43)$$

Inserting (4.42)–(4.43) into (4.41) yields:

$$\begin{aligned} \ddot{\mathbf{R}}_b^n \mathbf{r}_b^b &= \mathbf{R}_b^n \mathbf{S}(\omega_{b/n}^b) \mathbf{S}(\omega_{b/n}^b) \mathbf{r}_b^b - \mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \alpha_b^b \\ &\quad - \mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \mathbf{I}_b^{-1} \mathbf{S}(\omega_{b/n}^b) \mathbf{I}_b \omega_{b/n}^b \end{aligned} \quad (4.44)$$

This can now be inserted back into (4.35). This can further be restructured to the desired form to yield:

$$\begin{aligned} \ddot{g}_i &= 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i + 2\ddot{\mathbf{L}}_i^\top \mathbf{L}_i \\ &= 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i \\ &\quad + 2\mathbf{L}_i^\top \left( \mathbf{R}_{hj}^n \mathbf{a}_{hj}^{hj} - \mathbf{R}_{hj}^n \mathbf{S}(\mathbf{r}_{hj}^{hj}) \alpha_{hj}^{hj} - \mathbf{R}_l^n \mathbf{a}_l^l + \mathbf{R}_l^n \mathbf{S}(\mathbf{r}_l^l) \alpha_l^l \right) \\ &\quad + 2\mathbf{L}_i^\top (\Upsilon_{hj} - \Upsilon_l) \end{aligned}$$

where

$$\Upsilon_b = \mathbf{R}_b^n \mathbf{S}(\omega_{b/n}^b) \mathbf{S}(\omega_{b/n}^b) \mathbf{r}_b^b - \mathbf{R}_b^n \mathbf{S}(\mathbf{r}_b^b) \mathbf{I}_b^{-1} \mathbf{S}(\omega_{b/n}^b) \mathbf{I}_b \omega_{b/n}^b \quad (4.45)$$

This can be factored as

$$\mathbf{A}_i \ddot{\mathbf{q}} = \mathbf{b}_i \quad (4.46)$$

where the matrices  $\mathbf{A}_i$  and  $\mathbf{b}_i$  can be identified as

$$\mathbf{A}_i = 2\mathbf{L}_i^\top \begin{bmatrix} \mathbf{0}_{3 \times 6(j-1)} & \mathbf{R}_{hj}^n & -\mathbf{R}_{hj}^n \mathbf{S}(\mathbf{r}_{hj}^{hj}) \\ \mathbf{0}_{3 \times 6(m-j)} & -\mathbf{R}_l^n & \mathbf{R}_l^n \mathbf{S}(\mathbf{r}_l^l) \end{bmatrix} \quad (4.47)$$

$$b_i = 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i + 2\mathbf{L}_i^\top (\Upsilon_{hj} - \Upsilon_l) \quad (4.48)$$

The complete constraint equation for all constraints to the form (4.2) can then be created by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_i \\ \vdots \\ \mathbf{A}_p \end{bmatrix}_{p \times 6} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \\ \vdots \\ b_p \end{bmatrix}_{p \times 1} \quad (4.49)$$

#### 4.4.2 Simulation Steps

Let  $\ddot{\mathbf{q}}$  be the vector of generalized accelerations. To calculate this using the Udwadia-Kalaba equation, we need the generalized unconstrained accelerations. These can be obtained by:

$$\ddot{\mathbf{q}}_u = \mathbf{M}^{-1} \boldsymbol{\tau}_{RB} \quad (4.50)$$

where  $\mathbf{M}$  diagonal concatenation of the individual mass-matrices, and  $\boldsymbol{\tau}_{RB}$  contains the stacked forces for all bodies.

$\ddot{\mathbf{q}}_u$  is then inserted into (4.4) as

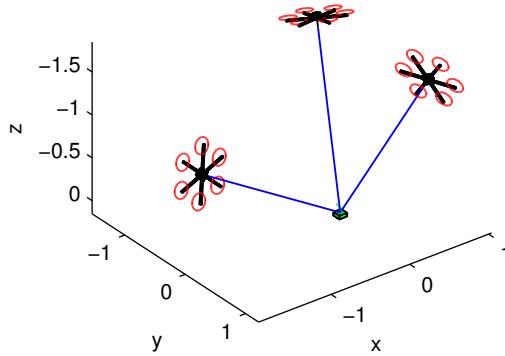
$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_u + \mathbf{M}^{-1/2} (\mathbf{A} \mathbf{M}^{-1/2})^+ (\mathbf{b} - \mathbf{A} \ddot{\mathbf{q}}_u) \quad (4.51)$$

The resulting constrained acceleration must be converted back to our body-differentiated form by:

$$\dot{\boldsymbol{\nu}} = -\mathbf{M}^{-1} \mathbf{C} + \ddot{\mathbf{q}} \quad (4.52)$$

#### 4.4.3 Simulation Example

In this example, a case with three multirotors and a suspended payload is illustrated. The payload is connected to the multirotors at an attachment-point located a downward distance of 0.1, so the load will induce a torque on the bodies. In this example, the multirotors are applied an external force equal to the constrain-forces subjected to each body (so, only artificial control). Thus, the purpose of this section is merely to show that the modeling methodology works in a case where the expected outcome is trivial. The final pose of the three multirotors can be seen in Figure 4.3.

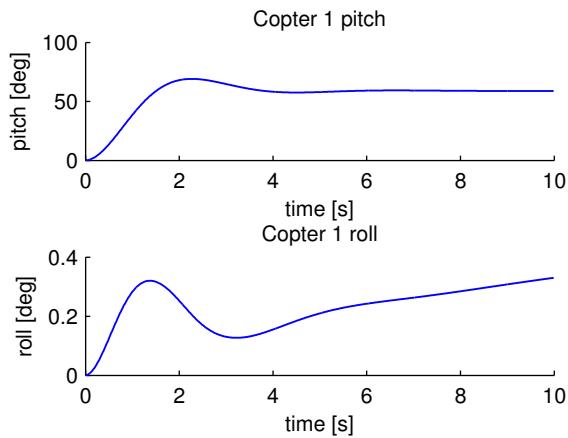


**Figure 4.3:** Final pose of the three multirotors connected to a payload.

The roll and pitch-motion of multirotor 1 can be seen in Figure 4.4. It is seen that the pitch stabilizes to match the vector along the wire, as expected.

#### 4.4.4 Simulation and numerical considerations

The equations of constraints developed here, ensures that the second-order derivative of  $g_j$  equals 0. However, due to numerical error in when integrating,  $g_j = 0$  will generally not be achieved. This will make the length of the wires drift, and is a well recognized problem in the literature [23]. As discussed in [15], several approaches to



**Figure 4.4:** Resulting roll and pitch of the first multirotor during the dynamic simulation example.

counter this problem exists. [15] evaluates two approaches, and yields good results by applying a virtual spring-damper approach. Here, a virtual spring-damper system is added along all wires to ensure that  $g_j = 0$ . Further details can be found in [15] and [17].

#### 4.4.5 Illustration of Dynamic Complexity

Just to make a point of the complexity of the resulting dynamic equations, and to give the reader a sense of why using the equation for model-based control synthesis directly is not necessarily feasible, consider the following example.

Let three multirotors be stationary aligned in a horizontal formation, with the equal distance  $r$  to each other. Further, let a suspended load be hanging in wires of equal lengths from each multirotor, of length  $\rho r$ . Then, in this stationary case, only considering position (that is, the attitude dynamics are disregarded), the pseudo-inverse in (4.5) evaluates to:

$$(\mathbf{AM}^{-1/2})^+ = \begin{bmatrix} \frac{\gamma_1 \gamma_3 \sqrt{m_c m_l}}{2r} & -\frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{2r} & -\frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{2r} \\ 0 & 0 & 0 \\ -\frac{\gamma_1 \gamma_3 \sqrt{m_c m_l} \rho}{2r} & \gamma_2 & \gamma_2 \\ \frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & -\frac{\gamma_1 \gamma_3 \sqrt{m_c m_l}}{4r} & \frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} \\ -\frac{\sqrt{3} \gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & \frac{\sqrt{3} \gamma_1 \gamma_3 \sqrt{m_c m_l}}{4r} & -\frac{\sqrt{3} \gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} \\ \frac{\gamma_2}{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)} & -\frac{\gamma_1 \gamma_3 \sqrt{m_c m_l} \rho}{2r} & \gamma_2 \\ \frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & \frac{\gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & -\frac{\gamma_1 \gamma_3 \sqrt{m_c m_l}}{4r} \\ -\frac{\sqrt{3} \gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & \frac{\sqrt{3} \gamma_1 m_c^{\frac{3}{2}} m_l (2\rho^2 - 1)}{4r} & -\frac{\sqrt{3} \gamma_1 \gamma_3 \sqrt{m_c m_l}}{4r} \\ -\frac{m_c \sqrt{m_l}}{\gamma_4 r} & \frac{m_c \sqrt{m_l}}{2\gamma_4 r} & \frac{m_c \sqrt{m_l}}{2\gamma_4 r} \\ 0 & -\frac{\sqrt{3} m_c \sqrt{m_l}}{2\gamma_4 r} & \frac{\sqrt{3} m_c \sqrt{m_l}}{2\gamma_4 r} \\ \frac{m_c \sqrt{m_l} \rho}{2r(m_l + 3m_c \rho^2 + m_l \rho^2)} & \frac{m_c \sqrt{m_l} \rho}{2r(m_l + 3m_c \rho^2 + m_l \rho^2)} & \frac{m_c \sqrt{m_l} \rho}{2r(m_l + 3m_c \rho^2 + m_l \rho^2)} \end{bmatrix}$$

where

$$\begin{aligned} \gamma_1 &= \frac{1}{9m_c^2 \rho^2 + 6m_c m_l \rho^4 + 9m_c m_l \rho^2 + 3m_c m_l + 2m_l^2 \rho^4 + 4m_l^2 \rho^2 + 2m_l^2} \\ \gamma_2 &= \frac{\gamma_1 m_c^{\frac{3}{2}} m_l \rho (2\rho^2 - 1)}{2r} \\ \gamma_3 &= m_c + 2m_l + 4m_c \rho^2 + 2m_l \rho^2 \\ \gamma_4 &= 2m_l \rho^2 + 3m_c + 2m_l \end{aligned}$$

When including non-symmetric positions and incorporating attitude dynamics, the effect of the dynamic coupling in these multi-lift systems is clear.

## 4.5 Chapter Summary

In this chapter, an introduction to multi-body operations was given. After an overview of related work, the chapter focused on modeling of the complex dynamics of multiple multirotor UAVs connected to a suspended payload. An introduction to different modeling techniques was given, and the equations of Udwadia-Kalaba was chosen as an effective way to create a modularized simulation environment, suitable for numerical verification of the controllers studied in the rest of this thesis.

## Chapter 5

# Distributed Kinematic Control

This chapter is based on **C1** ([63] K. Klausen, T. I. Fossen, and T. A. Johansen. Suspended Load Motion Control using Multicopters. In *Proceedings of the Mediterranean Conference of Control and Automation (MED)*, pages 1371–1376, Palermo, 2014. IEEE. ISBN 9781479958993. doi: 10.1109/MED.2014.6961567).

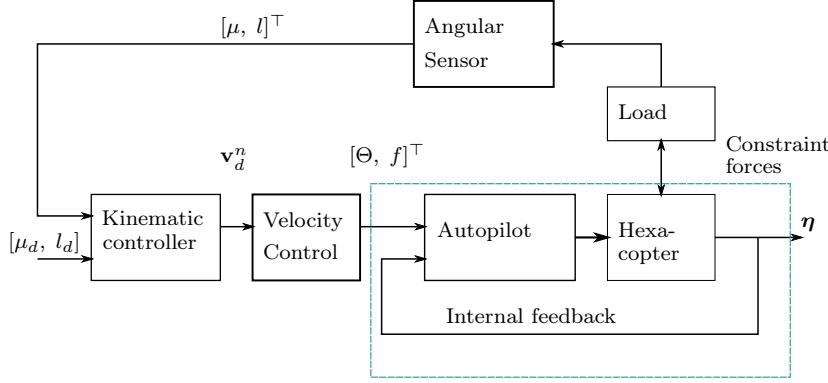
### 5.1 Distributed Kinematic Control

The purpose of this chapter is to derive a kinematic control law to guide a group of multirotors to perform a transport maneuver of a suspended load. The chosen approach is to be used for a commercial multirotor with an autopilot, as seen in Figure 5.1. The autopilot takes desired Euler angles and desired thrust as input signals.

Feedback to the kinematic controller is given by angle sensors, which measures the relative angle between the multirotor and the load. The overall kinematic controller uses an approach that somewhat resembles the approach in [76], but in this chapter we explicitly use available measurements from the angle sensor for feedback control.

#### 5.1.1 Organization

This chapter is organized as follows. First, for each UAV, a velocity-controller is developed that ensures that the multirotor UAV can track an input velocity setpoint. In this chapter, we assume that the drag from the suspended load can be measured and cancelled by the controller. Next, in Section 5.1.3, a distributed motion controller is developed, which by using only local measurements of the angles of the suspension wire, achieves a desired relative formation. In Section 5.2, two simulation studies are conducted. The first studies realization of the desired formation, while the latter also considers the transportation of the load to a pre-defined point. Section 5.3 summarizes and concludes the chapter.



**Figure 5.1:** Block diagram of the system. The boxes “autopilot” and “multirotor” are commercially available systems.  $\mu$  and  $l$  represents relative angle between the multirotor and the suspended load. The velocity and kinematic controllers are discussed in 5.1.2 and 5.1.3, respectively.

### 5.1.2 Kinematic Control

As discussed in Section 3.3, the multirotor autopilot takes the Euler angles  $\Theta$  and a thrust force  $f$  as input. The translational force in  $\{n\}$  generated from the multirotor propellers are given by

$$\tau^n = \mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} \quad (5.1)$$

where  $F = k_f f$ . This can be expanded to:

$$\tau_x^n = -k_f f(\sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta) \quad (5.2)$$

$$\tau_y^n = -k_f f(-\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi) \quad (5.3)$$

$$\tau_z^n = -k_f f(\cos \theta \cos \phi) \quad (5.4)$$

By using (5.2)–(5.4), Algorithm 5.1 summarizes how to generate desired angles based on a desired force in  $\{n\}$ .

**Algorithm 5.1** (Generation of translational forces). *Given desired force  $\mathbf{F}_d^n \in \mathbb{R}^3$  to be applied by the motors. Let  $\phi$ ,  $\theta$ , and  $\psi$  be current roll, pitch and yaw angle, respectively. The goal is to create desired angles  $\phi_d$ ,  $\theta_d$  and  $\psi_d$  to realize the desired force.*

1. Using (5.4), set  $f = -F_{d,z}^n / (k_f \cos \theta \cos \phi)$  to gain upwards thrust and compensate for multirotor tilt.
2. Using the current yaw angle  $\psi$  and  $F$  from Step 1, (5.2)–(5.3) becomes a set of two equations with two unknowns. Setting  $\psi_d = 0$ , gives

$$\tau_x^n = -k_f f(\cos \phi_d \sin \theta_d) \quad (5.5)$$

$$\tau_y^n = k_f f(\sin \phi_d) \quad (5.6)$$

which can be solved for  $\phi_d$  and  $\theta_d$ .

As shown in Section 3.3, the translational motion in  $\{n\}$  can be written as:

$$m\dot{\mathbf{v}}^n = \mathbf{F}_g^n + \mathbf{F}_l^n + \boldsymbol{\tau}^n \quad (5.7)$$

where  $\mathbf{F}_l^n$  and  $\mathbf{F}_g^n$  is pull by gravity and the load. The load disturbs the motion of the helicopter with a force  $|F_l|$  pointed along the wire. The direction of the wire is represented by the angles  $[\mu, l]^\top$ , which is defined by Figure 5.2. The components of this force can be calculated as

$$\mathbf{F}_l^n = \begin{bmatrix} F_{l,x} \\ F_{l,y} \\ F_{l,z} \end{bmatrix} = \begin{bmatrix} -|F_l| \cos l \cos \mu \\ -|F_l| \sin l \cos \mu \\ |F_l| \sin \mu \end{bmatrix} \quad (5.8)$$

The pull from the suspended load needs to be compensated for, in addition to gravity. Let the desired thrust in  $\{n\}$  be:

$$\boldsymbol{\tau}^n = -\mathbf{F}_g^n - \mathbf{F}_l^n + \boldsymbol{\alpha} \quad (5.9)$$

where  $\boldsymbol{\alpha}$  is an additional control force to be designed. Given reference velocities  $\mathbf{v}_d^n$  in  $\{n\}$ , we propose the following control law:

**Theorem 5.1.** *Given a desired velocity  $\mathbf{v}_d^n$ , the controller*

$$\boldsymbol{\alpha} = m\dot{\mathbf{v}}_d^n + \mathbf{K}_p(\mathbf{v}_d^n - \mathbf{R}_b^n \boldsymbol{\nu}_1) + \mathbf{K}_d(\dot{\mathbf{v}}_d^n - \mathbf{R}_b^n(\mathbf{S}(\boldsymbol{\nu}_2)\boldsymbol{\nu}_1 + \dot{\boldsymbol{\nu}}_1)) \quad (5.10)$$

where  $\mathbf{K}_k \in \mathbb{R}^{3 \times 3} = \mathbf{K}_k > 0, k \in \{p, d\}$  will render the equilibrium point  $\mathbf{v}^n = \mathbf{v}_d^n$  of (5.7) exponentially stable (GES).

*Proof.* Let  $\tilde{\mathbf{v}} := \mathbf{v}^n - \mathbf{v}_d^n$ . By using  $\boldsymbol{\nu}_1 = (\mathbf{R}_b^n)^\top \dot{\boldsymbol{\eta}}_1$  and  $\dot{\boldsymbol{\nu}}_1 = (\dot{\mathbf{R}}_b^n)^\top \dot{\boldsymbol{\eta}} + (\mathbf{R}_b^n)^\top \ddot{\boldsymbol{\eta}}_1$ , inserting into (5.10) yields:

$$\boldsymbol{\alpha} = m\dot{\mathbf{v}}_d^n - \mathbf{K}_p \tilde{\mathbf{v}} - \mathbf{K}_d \dot{\tilde{\mathbf{v}}}$$

By using (5.7), we get

$$\dot{\tilde{\mathbf{v}}} = -(m\mathbf{I} + \mathbf{K}_d)^{-1} \mathbf{K}_p \tilde{\mathbf{v}}$$

which proves that the zero-equilibrium  $\tilde{\mathbf{v}} = 0$  is GES.  $\square$

### 5.1.3 Group Coordination

Let the position of multirotor  $j$  relative to the load be expressed in the spherical coordinates  $[l_j, \mu_j, d_j]$ , as described in Figure 5.2. Then, the desired velocity calculated from the following controller will bring the system into a desired configuration:

**Theorem 5.2.** *Let  $l_d \in \mathbb{R}^m$  be evenly distributed in  $(-\pi \dots \pi]$ . Given desired configuration angle  $\mu_d$ ,  $n$  multirotors will evenly distribute themselves with the kinematic control law*

$$\mathbf{v}_{d,j}^n = \mathbf{R}_{l,j}^n [k_1(\mu_{d,j} - \mu_j) \quad -k_2(l_{d,j} - l_j) \quad 0]^\top \quad (5.11)$$

*Proof.* Let there be a coordinate frame  $\{l_j\}$  centered in multirotor  $j$ , which is rotated such that  $z^{l,j}$  points towards the suspended load,  $y^{l,j}$  is directed along  $z^{l,j} \times z^n$ , and  $x^{l,j}$  completes the right-hand rule (see Figure 5.2). Let  $u^{l,j}$ ,  $v^{l,j}$ ,  $w^{l,j}$  be linear velocities along  $x^{l,j}$ ,  $y^{l,j}$ ,  $z^{l,j}$ , respectively.

A coordinate transformation between  $\{l_j\}$  and  $\{n\}$  is given by a principal rotation  $\pi + l_j$  over the  $z^n$ -axis, followed by a rotation  $\pi/2 - \mu_j$  over the  $y^n$ -axis [37].

$$\begin{aligned}\mathbf{R}_{l,j}^n &= \mathbf{R}_{z,\pi+l_j} \mathbf{R}_{y,\frac{\pi}{2}-\mu_j} \\ &= \begin{bmatrix} -\cos l_j \sin \mu_j & \sin l_j & -\cos l_j \cos \mu_j \\ -\sin l_j \sin \mu_j & -\cos l_j & -\sin l_j \cos \mu_j \\ -\cos \mu_j & 0 & \sin \mu_j \end{bmatrix}\end{aligned}\quad (5.12)$$

From the definition of  $\{l_j\}$ , we have

$$\begin{aligned}\dot{\mu}_j &= du^{l,j} \\ \dot{l}_j &= -dv^{l,j}\end{aligned}$$

Let  $(\tilde{\cdot}) := (\cdot)_d - (\cdot)$ . Then, by  $u^{l,j} = k_1 \tilde{\mu}_j$  and  $v^{l,j} = -k_2 \tilde{l}_j$ :

$$\dot{\tilde{\mu}}_j = -dk_1 \tilde{\mu}_j \quad (5.13)$$

$$\dot{\tilde{l}}_j = -dk_2 \tilde{l}_j \quad (5.14)$$

which makes the zero-equilibrium points in (5.13)–(5.14) exponentially stable. Set  $\mathbf{v}_{d,j}^{l,j} = [u^{l,j}, v^{l,j}, 0]^\top$ . Then, the reference velocity in  $\{n\}$  is given by

$$\mathbf{v}_{d,j}^n = \mathbf{R}_{l,j}^n \mathbf{v}_{d,j}^{l,j} \quad (5.15)$$

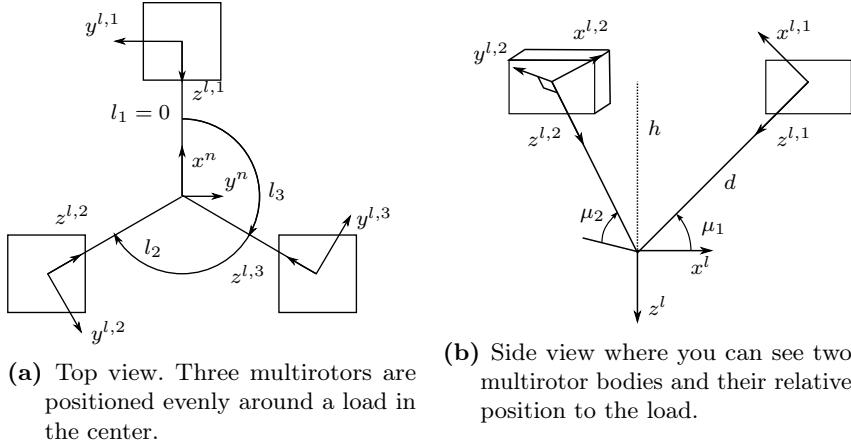
□

The desired velocity  $\mathbf{v}_{d,j}^n$  gained from (5.11) is then used together with (5.10) to generate a desired thrust for each multirotor. The angles  $\mu_j$  and  $l_j$  can be measured by first measuring angles relative to  $\{b\}$  by on-board angular sensors, and supplying with information in  $\Theta$ . More specifically,  $l_j$  requires knowledge of the heading  $\psi$ , while  $\mu_j$  requires the current roll  $\phi$  and pitch  $\theta$  angles.

## 5.2 Simulation Study

### 5.2.1 Simulation 1: Formation stabilization

In this section a simulation with three identical multirotors will be conducted. The three multirotors are connected to a load with three ropes, assumed massless. The system is modeled using the results in Section 4.4. Each multirotor has a mass of 4 kg, while the load weights 0.5 kg. The ropes are connected to the CG of both the load and the multirotors. The autopilot is simulated with a PD-type controller width a bandwidth of 5 rad/s. The controller also has a maximum value for the roll and pitch angles set to 15° to ensure smooth flights.



**Figure 5.2:** Illustration of the  $l$ -frame.

The load is initially placed at  $[0, 0, 0]^\top$ , while the multirotors are placed around the load, each with a distance of  $d_j = 2$ , with the following coordinates:

$$\begin{bmatrix} \mu_1 \\ l_1 \end{bmatrix} = \begin{bmatrix} 30^\circ \\ -180^\circ \end{bmatrix}, \quad \begin{bmatrix} \mu_2 \\ l_2 \end{bmatrix} = \begin{bmatrix} 45^\circ \\ -60^\circ \end{bmatrix}, \quad \begin{bmatrix} \mu_3 \\ l_3 \end{bmatrix} = \begin{bmatrix} 60^\circ \\ 60^\circ \end{bmatrix}$$

The desired  $\mu_d$  for all multirotors are  $\mu_d = 45^\circ$ . Further, the desired  $l_d$ s are

$$l_{d,1} = -120^\circ, \quad l_{d,2} = 0^\circ, \quad l_{d,3} = 120^\circ$$

The numerical integration is done with Runge-Kutta 4, running at 100 Hz. The results can bee seen in Figures 5.3–5.5.

In the simulations, the variables  $\mu$  and  $l$  must be calculated from the current position of the load and each multirotor. Let  $\bar{\mathbf{x}}_j = [\bar{x}_j, \bar{y}_j, \bar{z}_j] := \eta_j - \eta_l$ . Then,  $\mu_j$  and  $l_j$  can be calculated by:

$$\mu_j = \text{atan}\left(\frac{-\bar{z}_j}{\sqrt{\bar{x}_j^2 + \bar{y}_j^2}}\right) \quad (5.16)$$

$$l_j = \text{atan2}(\bar{y}_j, \bar{x}_j) \quad (5.17)$$

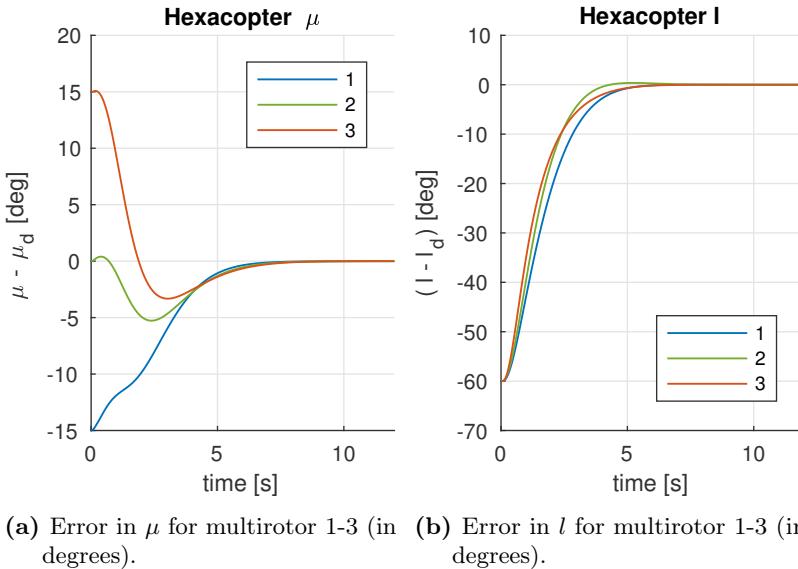
which is valid for  $\bar{x}_j^2 + \bar{y}_j^2 \neq 0$ , that is, when the load is not directly above or below a multirotor.

### 5.2.2 Results and discussions

Figure 5.3 shows the time plot of  $\mu$  and  $l$  for all three multirotors. It it seen that the controller forces  $\mu_j$  and  $l_j$  to their desired values. Further, the constrained acceleration of the load can be seen in Figure 5.4a. It is seen that after the multirotors

reach their position, they stabilize the position of the load. But, as can be seen by Figure 5.4b, the position of the load drifts a bit while they get into formation. This is due to the fact that no attempt to control the position of the load is done in this simulation. This is done in the next section.

Further, the roll and pitch of multirotor 1 can be seen in Figure 5.5. It can be seen that the stationary value is not zero, indicating that the multirotor must be tilted to compensate for the additional pull from the load-wire.



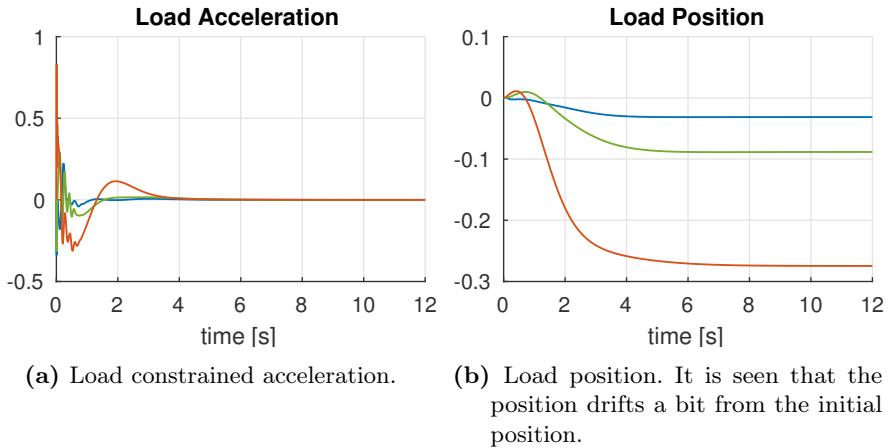
**Figure 5.3:**  $\mu$  and  $l$  for all multirotors. In both 5.3a and 5.3b, the blue line is for multirotor 1, the dashed green for multirotor 2, and the dash-dotted line for multirotor 3.

### 5.2.3 Simulation 2: Load transport

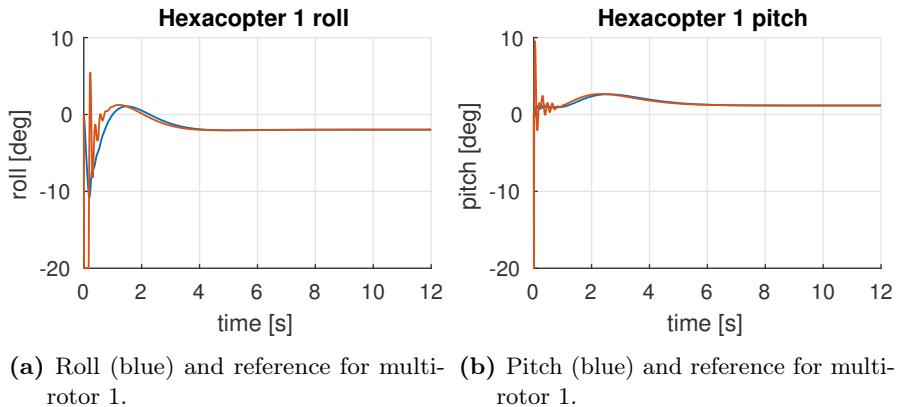
The start of this simulation is identical to that of the previous section, but here, after a stable formation is reached, an additional velocity component is inserted to transport the load to a desired location. Let  $\boldsymbol{\eta}_{l,d} \in \mathbb{R}^3$  be the desired position of the load. Further, let  $\mathbf{v}_{d,p}^n \in \mathbb{R}^3$  be the velocity component to guide the load to its desired position, be

$$\mathbf{v}_{d,p}^n = K_l(\boldsymbol{\eta}_{l,d} - \boldsymbol{\eta}_l)$$

so that after the desired configuration is reached, the total reference velocity is  $\mathbf{v}_{d,p}^n + \mathbf{v}_{d,j}^n$ , where the last term is calculated from (5.15). In this simulation,  $\boldsymbol{\eta}_{l,d} = [1, 5, 0]^\top$  and  $K_l = 1$ . The results of the simulation can be seen in Figures 5.6–5.9.



**Figure 5.4:** Load acceleration (5.4a) and position (5.4b). In both figures, blue, green and red represents the directions  $x^n$ ,  $y^n$  and  $z^n$ , respectively.



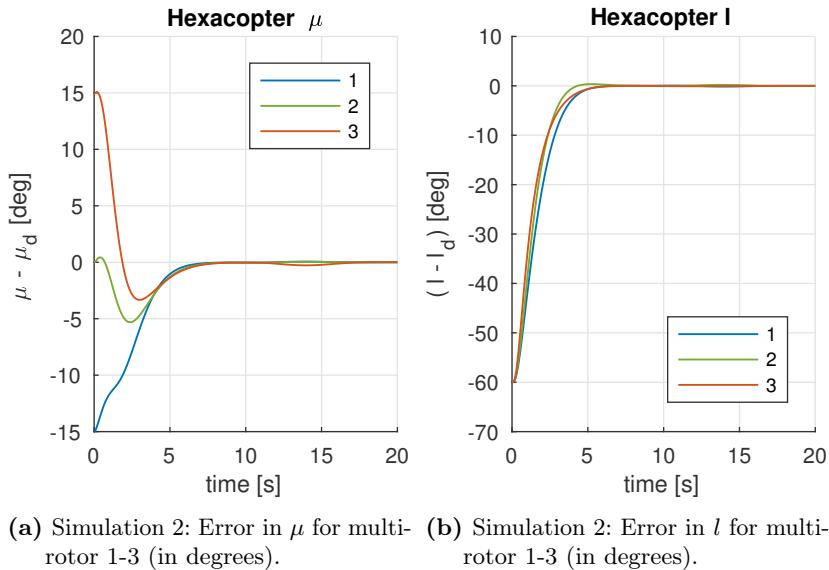
**Figure 5.5:** Roll (5.5a) and pitch (5.5b) for multirotor 1.

#### 5.2.4 Results and discussions

In Figure 5.6, it is seen that the error in  $\mu_j$  and  $l_j$  stays at zero during the transport manoeuvre, which shows the feasibility of the proposed controller. Further, Figure 5.7b shows that the load is transported to the desired location  $[1, 5, 0]^\top$ . From Figure 5.8, it is seen that the velocity-controller for multirotor 1 follows the desired velocities. A time-lapse of the maneuver can be seen in Figure 5.9, at the times indicated in the figure.

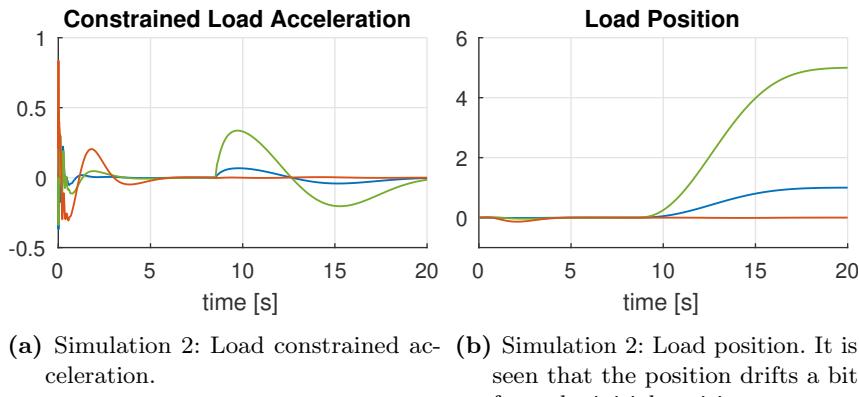
### 5.3 Chapter Summary and Conclusions

In this chapter, we have derived a nonlinear kinematic control law for cooperative load transport using multiple multirotors. The proposed control law can be used as a



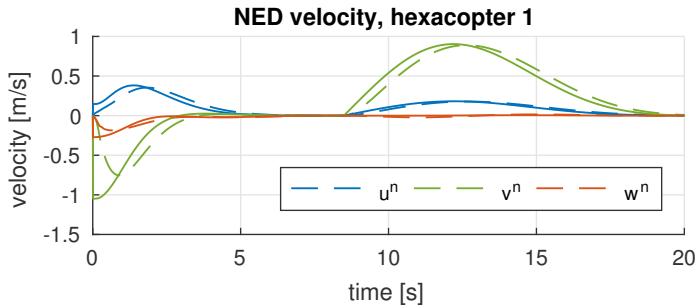
(a) Simulation 2: Error in  $\mu$  for multi-  
rotor 1-3 (in degrees). (b) Simulation 2: Error in  $l$  for multi-  
rotor 1-3 (in degrees).

**Figure 5.6:**  $\mu$  and  $l$  for all multirotors in Simulation 2. In both 5.6a and 5.6b, the blue line is for multirotor 1, the dashed green for multirotor 2, and the dash-dotted line for multirotor 3.

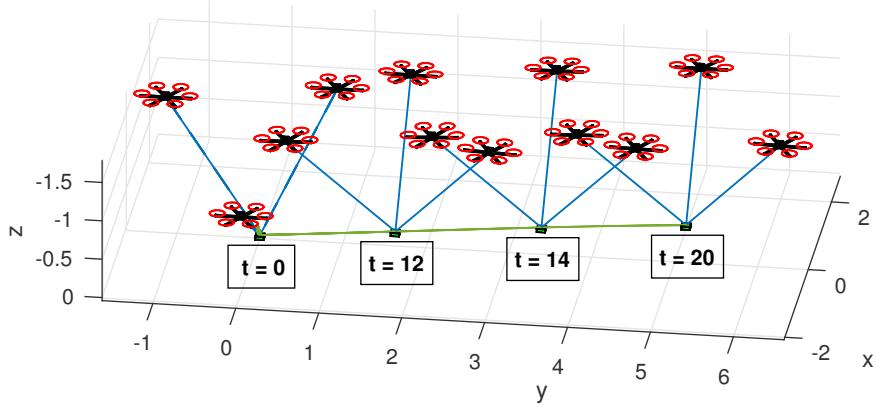


(a) Simulation 2: Load constrained ac-  
celeration. (b) Simulation 2: Load position. It is  
seen that the position drifts a bit  
from the initial position.

**Figure 5.7:** Simulation 2: Load acceleration (5.7a) and position (5.7b). In both figures, blue, green and red represents the directions  $x^n$ ,  $y^n$  and  $z^n$ , respectively.



**Figure 5.8:** Simulation 2: NED velocities for multirotor 1.



**Figure 5.9:** Simulation 2: Trajectory overview.

guidance law for a multirotors with autopilot, and illustrates how to achieve relative formation without external positioning systems accurate enough for formation control, such as RTK GPS or external camera systems. The control-design relied on measurement of the tensile-force from the suspended load. Numerical simulations illustrates the performance of the controller.



## Chapter 6

# Cooperative Path Following

This chapter is based on **C3** ([65] K. Klausen, T. I. Fossen, T. A. Johansen, and A. P. Aguiar. Cooperative path-following for multirotor UAVs with a suspended payload. *Proceedings of the 2015 IEEE Conference on Control Applications (CCA)*, pages 1354–1360, 2015. doi: 10.1109/CCA.2015.7320800).

### 6.1 Introduction

This chapter considers the use of multirotor UAVs to transport a suspended payload. This load can be a variety of things, such as cargo transportation or sensor deployment. Due to the inherent capacity limitations of small multirotor UAVs, we illustrate how a team of UAVs can cooperatively transport a single payload to distribute and minimize the load on each vehicle, thus increasing the total lift capacity and operation time.

Similar problems are addressed for scale helicopters in [13], which were also validated in experiments. [35] considers motion planning and control for a team of indoor multirotors to manipulate a suspended payload, and [73] utilizes geometric control to generate coordinate-free controllers to control a suspended load in 6 degrees of freedom (DOF). In addition to distributed control, this chapter also considers synchronization between the vehicles in the mission over a communication network. Fundamental consensus algorithms for synchronization are discussed in detail in [110], utilizing graph theory to create distributed control laws under communication constraints. Passivity-based approaches are discussed in [8], [10]. See also [101] and references therein.

In [120], the authors introduce a method to solve the formation maneuvering problem. By solving individual maneuvering problems ([121], [51]), it is shown that by distributing and synchronizing their path parameterization variable, the formation problem can be solved. Utilizing graph theory ([34]), a similar framework (coordinated path-following) is introduced in [2] and applied to a team of under-actuated autonomous underwater vehicles. In [3] and [41], the coordinated path-following (CPF) problem is addressed in the presence of time delays and discrete logic-based communications, respectively.

This chapter applies the results from the above citations to the team of multirotors transporting the suspended payload. By synchronizing each UAV along a certain path, the position of the suspended load can be manipulated.

The chapter is organized as follows. First, the problem is described in more detail. The dynamical system is modeled using a combination of Newton-Euler mechanics for each individual vehicle, while the interconnected system with dynamical constraints imposed by transportation wires is modeled by the Fundamental Equation [138]. In Section 6.2, a non-linear path-following controller is designed for the multirotors by the back-stepping technique. Further in Section 6.2.1, a controller to synchronize the vehicles along the path using communication links is presented. Section 6.3 illustrates an algorithm for spatial path generation for each vehicle, that minimizes the load drag variation among the multirotors as they move along the desired load path. A speed profile is designed next, followed by numerical simulations in Section 6.4.

### 6.1.1 Problem Statement

Inspired by [2], we design the overall controller in two parts. First, a local controller that solves the *path-following problem* is employed on each vehicle:

**Definition 6.1. Path-following problem ([2]):**

Let  $\mathbf{p}_{d_i}(\gamma) \in \mathbb{R}^3$  be a desired path parameterized by a continuous variable  $\gamma_i \in \mathbb{R}$ , and  $v_L(\gamma_i) \in \mathbb{R}$  a desired reference speed for vehicle  $i$ . Design feedback control laws for  $\tau_i$ , such that all the closed-loop signals are bounded, the position of the vehicle converges to and remains inside a tube centered around the desired path, and the vehicle travels at a desired speed assignment  $v_L$ , that is,  $\dot{\gamma} - v_L \rightarrow 0$  as  $t \rightarrow \infty$ .

The path-following controller is designed in such a way, that it can take an extra input,  $\tilde{v}_i$ , as a coordination variable to speed up or slow down along the path, to synchronize its position along the path with the other vehicles. This is the *coordination problem*:

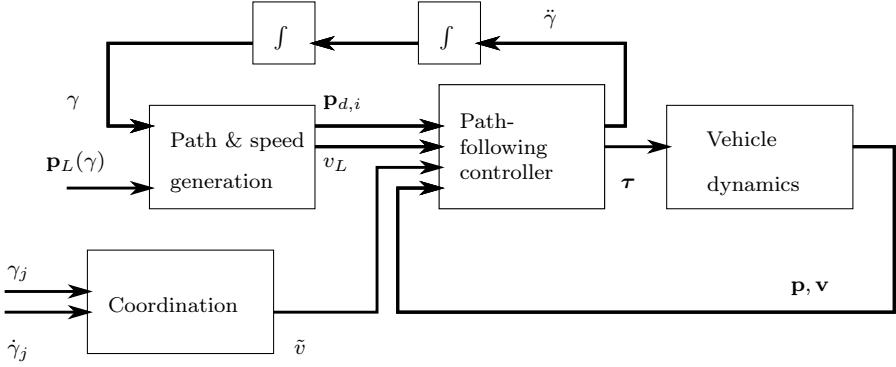
**Definition 6.2. Coordination problem ([2]):**

For each vehicle  $i \in \mathcal{I} := 1, \dots, n$  derive a control law for the speed command as a function of  $\gamma_i$  and  $\gamma_j$ ,  $j \in \mathcal{N}$ , such that  $\gamma_i - \gamma_j, \forall i, j \in \mathcal{I}$  approach zero as  $t \rightarrow \infty$  and the formation travels at an assigned speed  $v_L$ , that is,  $|\dot{\gamma}_i - v_L|$  tends to zero.

The coordinated path-following (CPF) problem is the combination of the two previous stated problems. An overview of the control architecture can be seen in Figure 6.1.

### 6.1.2 Dynamic modeling

In this section, a dynamical model of a single UAV is first presented. The dynamic model for a the multirotor is slightly different than introduced in Chapter 3 to better fit the control-structure derived in this chapter. This is a well-studied system in the literature, and the presentation is thus quite brief. As in Chapter 4, the



**Figure 6.1:** An overview of the controller structure.

equations of Udwadia and Kalaba is used to model the forces of constraint acting on each vehicle due to the interconnecting wires. This is based on the work in [15].

Let the NED frame  $\{n\}$  be an inertial frame. The position of each UAV in this inertial frame is given by  $\mathbf{p}_i \in \mathbb{R}^3$ . Further, let  $\{b_i\}$  be a body-aligned frame at UAV  $i$ , whose orientation wrt.  $\{n\}$  is given by the transformation matrix  $\mathbf{R}_{b,i}^n \in \mathcal{SO}^3$ . Thus, the kinematics are given by

$$\dot{\mathbf{p}}_i = \mathbf{R}_{b,i}^n \mathbf{v} \quad (6.1)$$

$$\dot{\mathbf{R}}_{b,i}^n = \mathbf{S}(\boldsymbol{\omega}_{b,i}^n) \mathbf{R}_{b,i}^n \quad (6.2)$$

where  $\mathbf{v} \in \mathbb{R}^3$  is the translational velocity in  $\{b_i\}$ ,  $\boldsymbol{\omega}_{b,i}^n$  is the rotational velocity of  $\{b_i\}$  with respect to  $\{n\}$ , and  $\mathbf{S}(\cdot)$  is the skew-symmetric matrix operator defined by

$$\mathbf{S}(\lambda) := \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \quad (6.3)$$

for  $\lambda \in \mathbb{R}^3$ .

The kinetics are given from standard Newton-Euler rigid-body dynamics:

$$\mathbf{M}_i \ddot{\mathbf{v}}_i = -\mathbf{S}(\boldsymbol{\omega}) \mathbf{M}_i + \boldsymbol{\tau}_{a,i} + \Pi \boldsymbol{\tau}_{c,i} + \mathbf{g}_i + \boldsymbol{\tau}_{L,i} \quad (6.4)$$

$$\mathbf{J}_i \dot{\boldsymbol{\omega}}_{b,i}^n = \mathbf{S}(\mathbf{J}_i \boldsymbol{\omega}_{b,i}^n) \boldsymbol{\omega}_{b,i}^n + \boldsymbol{\tau}_{M,i} \quad (6.5)$$

where  $\mathbf{M}_i > 0, \in \mathbb{R}^{3 \times 3}$  is the mass matrix of the vehicle, and  $\mathbf{J}_i > 0$  its inertia dyadic in the vehicle body fixed frame. As a multirotor is only capable of producing translational force along its  $z$ -axis, we have  $\Pi = [0, 0, 1]^\top$  and the control force  $\boldsymbol{\tau}_{a,i} \in \mathbb{R}$ . The effect of gravity is  $\mathbf{g}_i = \mathbf{R}_{b,i}^n [0, 0, m_i g]^\top$ .  $\boldsymbol{\tau}_{a,i} \in \mathbb{R}^3$  encompasses all aerodynamical forces acting on the body.  $\boldsymbol{\tau}_{M,i} \in \mathbb{R}^3$  represents control momentum.  $\boldsymbol{\tau}_{L,i} \in \mathbb{R}^3$  are the forces of constraint from the suspended payload.

In this work, we consider the suspended payload as a point-mass with position  $\mathbf{p}_l$  and mass  $m_l$ , which gives the dynamics:

$$m_l \ddot{\mathbf{p}}_l = \begin{bmatrix} 0 \\ 0 \\ m_l g \end{bmatrix} - \sum_{i=0}^N \boldsymbol{\tau}_{L,i} \quad (6.6)$$

Next, we utilize the results in [139] to find the forces of constraint  $\boldsymbol{\tau}_{L,i}$ .

### 6.1.3 Multi-body modeling

This section is based on the work by [15] and [139], and is a simplified version of that presented in Chapter 4 in that it only considers wires attached to the Centre of Gravity of the vehicles.

Consider  $N$  multirotors, connected by  $N$  wires from the body centre of gravity to a suspended load. The goal is to find the constraint force  $\boldsymbol{\tau}_{L,i}$ . Note that as the suspension is done in the vehicle Centre of Gravity (CoG), the attitude dynamics are unaffected. Define the concatenated vectors  $\mathbf{p}$ ,  $\mathbf{v}$  and  $\boldsymbol{\tau}_L$  as

$$\mathbf{p} := \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_N \\ \mathbf{p}_l \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \\ \dot{\mathbf{p}}_l \end{bmatrix}, \quad \boldsymbol{\tau}_L = \begin{bmatrix} \boldsymbol{\tau}_{L,1} \\ \boldsymbol{\tau}_{L,2} \\ \vdots \\ \boldsymbol{\tau}_{L,N} \\ \boldsymbol{\tau}_l \end{bmatrix} \quad (6.7)$$

where  $\boldsymbol{\tau}_l = -\sum_{i=0}^N \boldsymbol{\tau}_{L,i}$

Consider wire  $i$ , connecting multirotor  $i$  and the load. The line along the wire is given by

$$\mathbf{L}_i^n = \mathbf{p}_i - \mathbf{p}_l \quad (6.8)$$

A constraint  $g_i$  acting on each body is now given by

$$g_i = \|\mathbf{L}_i\|^2 - d_i^2 = 0 \quad (6.9)$$

where  $d_i$  is the nominal length of wire  $i$ . It's derivatives are given by

$$\dot{g}_i = 2\dot{\mathbf{L}}_i^\top \mathbf{L}_i \quad (6.10)$$

$$\ddot{g}_i = 2\ddot{\mathbf{L}}_i^\top \mathbf{L}_i + 2\dot{\mathbf{L}}_i^\top \dot{\mathbf{L}}_i = 0 \quad (6.11)$$

The constraint (6.11) can now be formulated in the standard form:

$$\mathbf{A}_i(\mathbf{p}, \mathbf{v}) \dot{\mathbf{v}} = \mathbf{b}_i(\mathbf{p}, \mathbf{v}) \quad (6.12)$$

where

$$\mathbf{A}_i = 2\mathbf{L}_i^\top \begin{bmatrix} \mathbf{0}_{3 \times 3(i-1)} & \mathbf{R}_{b,i}^n & \mathbf{0}_{3 \times 3(N-i)} & -\mathbf{I} \end{bmatrix} \quad (6.13)$$

and

$$\begin{aligned}\mathbf{b}_i = & -2\mathbf{L}_i^\top (\mathbf{R}_{b,i}^n \mathbf{S}(\boldsymbol{\omega}_{b,i}) \mathbf{v}_i - \dot{\mathbf{p}}_l) \\ & - \dot{\mathbf{L}}_i^\top \mathbf{L}_i\end{aligned}\quad (6.14)$$

According to [139], the constraint force  $\boldsymbol{\tau}_L$  are now given by:

$$\boldsymbol{\tau}_L = \mathbf{M}^{1/2} (\mathbf{A} \mathbf{M}^{-1/2})^+ (\mathbf{b} - \mathbf{A} \dot{\mathbf{v}}) \quad (6.15)$$

where  $\mathbf{M}$ ,  $\mathbf{A}$  and  $\mathbf{b}$  are concatenations of  $\mathbf{M}_i$ ,  $\mathbf{A}_i$  and  $\mathbf{b}_i$ , respectively, and  $(\cdot)^+$  denotes the Moore-Penrose pseudo inverse.

## 6.2 Control Design

In this section, a path following controller for each UAV is designed. For notational simplicity, we skip the subscript  $i$  in this part, and it is understood that this concerns each single UAV.

Low-level control of multirotors has been extensively studied in the past, see e.g. [80] for a good review of recent advances. Thus, in this work we consider the case when each UAV is equipped with a low-level controller, handling attitude control. That is, there exists a control law for  $\boldsymbol{\tau}_M$  that takes input desired roll and pitch angle, along with a desired yaw rate  $r$ . In addition, as described in [80], an algorithm is embedded in the low-level controller that translates between a desired force  $\boldsymbol{\tau}$  in  $\{b_i\}$  and attitude, thus giving the following dynamics:

$$\dot{\mathbf{p}} = \mathbf{R}(\psi) \mathbf{v} \quad (6.16)$$

$$\dot{\psi} = r \quad (6.17)$$

$$\mathbf{M} \dot{\mathbf{v}} + \mathbf{D} \mathbf{v} + \mathbf{g} = \boldsymbol{\tau} + \boldsymbol{\tau}_L \quad (6.18)$$

where  $\mathbf{D} > 0$  and the term  $\mathbf{D} \mathbf{v}$  represents air drag at low speeds,  $\mathbf{R}(\psi)$  is a principal rotation about the  $\{n\}$  z-axis, and control inputs  $\mathbf{u} = [\boldsymbol{\tau} \ r]^\top$ .

### Path following controller

This section will synthesize a path-following controller for the system from the previous section. Let  $\mathbf{p}_d(\gamma)$  be a desired path in  $\mathbb{R}^3$  to follow, and  $\psi_d$  a corresponding desired yaw angle. Define the error variables as:

$$\mathbf{z}_1 := \mathbf{p} - \mathbf{p}_d(\gamma) \quad (6.19)$$

$$z_\psi := \psi - \psi_d(\gamma) \quad (6.20)$$

whose derivatives are

$$\dot{\mathbf{z}}_1 = \mathbf{R}(\psi) \mathbf{v} - \mathbf{p}_d^\gamma(\gamma) \dot{\gamma} \quad (6.21)$$

$$\dot{z}_\psi = r - \psi_d^\gamma(\gamma) \dot{\gamma} \quad (6.22)$$

Next, define the along-track speed error as:

$$\eta := \dot{\gamma} - v_L(\gamma(t), t) \quad (6.23)$$

where  $v_L$  is a mission velocity known to all vehicles. Thus,

$$\dot{\mathbf{z}}_1 = \mathbf{R}(\psi)\mathbf{v} - \mathbf{p}_d^\gamma(\gamma)v_L(\gamma(t), t) - \mathbf{p}_d^\gamma(\gamma)\eta \quad (6.24)$$

$$\dot{z}_\psi = r - \psi_d^\gamma(\gamma)v_L(\gamma(t), t) + \psi_d^\gamma(\gamma)\eta \quad (6.25)$$

For notational simplicity, we will drop the arguments on  $v_L$  and the reference signals  $\mathbf{p}_d$  and  $\psi_d$ . Further, define the virtual control  $\boldsymbol{\alpha}$  as

$$\mathbf{z}_2 := \mathbf{v} - \boldsymbol{\alpha} \quad (6.26)$$

which gives

$$\mathbf{M}\dot{\mathbf{z}}_2 = \boldsymbol{\tau} + \boldsymbol{\tau}_L - \mathbf{g} - \mathbf{D}\mathbf{v} - \mathbf{M}\dot{\boldsymbol{\alpha}} \quad (6.27)$$

### Step 1

Let a Control Lyapunov Function (CLF) be

$$V_1(\mathbf{z}_1, z_\psi) = \frac{1}{2}\mathbf{z}_1^\top \mathbf{z}_1 + \frac{1}{2}z_\psi^2 \quad (6.28)$$

whose derivative along the solutions are

$$\begin{aligned} \dot{V}_1 &= \mathbf{z}_1^\top (\mathbf{R}(\psi)\mathbf{z}_2 + \mathbf{R}(\psi)\boldsymbol{\alpha} - \mathbf{p}_d^\gamma v_L - \mathbf{p}_d^\gamma \eta) \\ &\quad + z_\psi(r - \psi_d^\gamma v_L + \psi_d^\gamma \eta) \end{aligned} \quad (6.29)$$

Next, choose the virtual control and yaw reference as

$$\mathbf{R}(\psi)\boldsymbol{\alpha} = -\mathbf{K}_p \mathbf{z}_1 + \mathbf{p}_d^\gamma v_L \quad (6.30)$$

$$r = -k_\psi z_\psi + \psi_d^\gamma v_L \quad (6.31)$$

which gives

$$\dot{V}_1 = -\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1 - k_\psi z_\psi^2 + \mathbf{z}_1^\top \mathbf{R}(\psi)\mathbf{z}_2 + h_1 \eta \quad (6.32)$$

where

$$h_1 = -\mathbf{z}_1^\top \mathbf{p}_d^\gamma - z_\psi \psi_d^\gamma \quad (6.33)$$

### Step 2

Let

$$V_2(\mathbf{z}_1, z_\psi, \mathbf{z}_2) = V_1 + \frac{1}{2}\mathbf{z}_2^\top \mathbf{M}\mathbf{z}_2 \quad (6.34)$$

whose derivative is

$$\begin{aligned} \dot{V}_2 &= -\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1 - k_\psi z_\psi^2 \\ &\quad + \mathbf{z}_2^\top (\mathbf{R}(\psi)\mathbf{z}_1 + \boldsymbol{\tau} + \boldsymbol{\tau}_L - \mathbf{D}\mathbf{v} - \mathbf{g} - \mathbf{M}\dot{\boldsymbol{\alpha}}) \\ &\quad + h_1 \eta \end{aligned} \quad (6.35)$$

We can express  $\dot{\alpha}$  as

$$\dot{\alpha} = \dot{\alpha} + h_2\eta \quad (6.36)$$

where

$$\begin{aligned} \dot{\alpha} = & (-\mathbf{S}(r)\mathbf{R}^\top \mathbf{p}_d^\gamma + \mathbf{R}^\top \mathbf{p}_d^{\gamma^2} v_L + \mathbf{R}^\top \mathbf{p}^\gamma v_L^\gamma) v_L \\ & + \mathbf{S}(r)\mathbf{R}^\top \mathbf{K}_p \mathbf{z}_1 - \mathbf{R}^\top \mathbf{K}_p \mathbf{R} \mathbf{z}_2 + \mathbf{R}^\top \mathbf{K}_p \mathbf{K}_p \mathbf{z}_1 \end{aligned} \quad (6.37)$$

$$h_2 = \mathbf{R}^\top \mathbf{p}_d^{\gamma^2} + \mathbf{R}^\top \mathbf{p}_d^\gamma v_L^\gamma + \mathbf{R}^\top \mathbf{K}_p \mathbf{p}^\gamma \quad (6.38)$$

Assuming we know  $\tau_L$ , let the control  $\tau$  be

$$\tau = -\tau_L - \mathbf{R}(\psi) \mathbf{z}_1 + \mathbf{D}\alpha + \mathbf{g} + \mathbf{M}\dot{\alpha} - \mathbf{K}_d \mathbf{z}_2 \quad (6.39)$$

which gives

$$\dot{V}_2 = -\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1 - k_\psi z_\psi^2 - \mathbf{z}_2^\top \mathbf{K}_d \mathbf{z}_2 - \mathbf{z}_2 D \mathbf{z}_2 + h_3 \eta \quad (6.40)$$

where

$$h_3 = h_1 - \mathbf{z}_2^\top \mathbf{M} h_2. \quad (6.41)$$

### Step 3

Finally, consider

$$V_3 = V_2 + \frac{1}{2}\eta^2 \quad (6.42)$$

which gives

$$\begin{aligned} \dot{V}_3 = & -\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1 - k_\psi z_\psi^2 - \mathbf{z}_2^\top \mathbf{K}_d \mathbf{z}_2 - \mathbf{z}_2 D \mathbf{z}_2 \\ & + \eta(\ddot{\gamma} - v_L^\gamma \dot{\gamma} + h_3) \end{aligned} \quad (6.43)$$

where we have used the fact that  $\dot{\eta} = \ddot{\gamma} - v_L^\gamma \dot{\gamma}$ . Inspired by [2], we chose  $\ddot{\gamma}$  as

$$\ddot{\gamma} = -k_\eta \eta + h_3 + v_L^\gamma \dot{\gamma} + \tilde{v} \quad (6.44)$$

which leads to

$$\dot{V}_3 = -\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1 - k_\psi z_\psi^2 - \mathbf{z}_2^\top \mathbf{K}_d \mathbf{z}_2 - \mathbf{z}_2 D \mathbf{z}_2 - k_\eta \eta^2 + \eta \tilde{v} \quad (6.45)$$

We are now ready to state the main result of this section.

**Theorem 6.1.** *The controllers given by (6.39) and (6.44) robustly solves the path-following problem. The closed-loop state, including the path following error  $|\mathbf{p} - \mathbf{p}_d|$  and the speed error  $|\dot{\gamma}_i - v_L(\gamma_i)|$  are bounded and input-to-state stable (ISS) from an input  $\tilde{v}$ .*

*Proof.* By applying Youngs Inequality to (6.45), one can show that there exists  $\lambda, \rho > 0$  s.t.

$$\dot{V}_3 \leq -\lambda V_3 + \rho |\tilde{v}|. \quad (6.46)$$

Further, the zero-equilibrium of the unforced system ( $\tilde{v} = 0$ ) is globally exponentially stable according to [61, Th. 4.10], and by direct application of [61, Th. 4.19] the system is ISS from input  $\tilde{v}$ .  $\square$

The path-following controller here ensures that a single vehicle will follow a path  $p_d(\gamma)$  along a given speed profile  $v_L(\gamma)$ .

### 6.2.1 Coordination controller

The coordination controller utilizes graph theory to represent communication links, and a brief introduction is given here.

Let  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  be a graph representing the interconnections of the vehicles in the networks. The set of nodes  $\mathcal{V}$  represents a vehicle, while the set of edges  $\mathcal{E}$  is a bidirectional communication link. Nodes  $i$  and  $j$  are *adjacent* if an edge exists between them. A graph  $\mathcal{G}$  is said to be *connected* if there exists a path between every two nodes in the graph. Let  $N_i$  be the set of edges originating in node  $i$ . The adjacency matrix  $A$  is a square matrix with rows and columns indexed by the nodes such that the  $i, j$ -entry of  $A$  is defined by

$$A_{i,j} = \begin{cases} 1 & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases} \quad (6.47)$$

Further, the degree matrix  $D$  of a graph is a diagonal matrix of the cardinality of  $\mathcal{G}$ , that is

$$D_{i,i} = |N_i| \quad (6.48)$$

The laplacian  $L$  of a graph is defined as  $L := D - A$ . As an example, consider the case where a graph consists of three nodes, all connected to each other. This is a graph with three nodes and three edges, and can be represented by

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (6.49)$$

### 6.2.2 Coordination

Now, let the  $n$  multirotors be connected by a bi-directional communication link. Let  $\mathcal{G}$  be the graph describing the connectivity map, and  $L$  its laplacian matrix. We assume that the graph  $\mathcal{G}$  is *connected*. Since the update law (6.44) has the same structure as introduced in [3], and the path-following controller is ISS by Theorem 1, we apply [2, Theorem 3] to solve the overall CPF problem.

**Lemma 6.2.** *By applying the distributed feedback law*

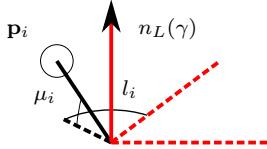
$$\tilde{\mathbf{v}} = -\mathbf{K}_1 L \dot{\gamma} - \mathbf{K}_2 (\dot{\gamma} - v_L \mathbf{1} + \mathbf{K}_1 L \gamma) \quad (6.50)$$

where  $\tilde{\mathbf{v}} := [\tilde{v}_i]_{i \in \mathcal{I}}$ ,  $\gamma := [\gamma_i]_{i \in \mathcal{I}}$ ,  $\dot{\gamma} := [\dot{\gamma}_i]_{i \in \mathcal{I}}$ ,  $\mathbf{K}_1, \mathbf{K}_2$  are positive diagonal matrices, and given sufficiently large path-following gains or coordination gains, the overall closed-loop system consisting of the path following error  $[\mathbf{z}_{1,i}]_{i \in \mathcal{I}}$ , coordination error  $[\gamma_i - \gamma_j]_{i \in \mathcal{I}, j \in N_i}$  and speed error are Input-Output-practically Stable (IOpS).

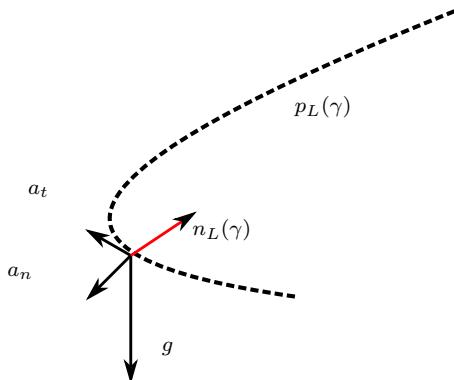
*Sketch of proof.* Direct application of [2, Theorem 3] for continuous communication. The proof involves showing IOpS for the coordination error subsystem, and further application of the small-gain theorem for the interconnected system with the path-following controller.  $\square$

### 6.3 Path generation

This section illustrates how a path can be generated for each UAV, to ensure that the inter-vehicle formation is kept. In addition, the formation is designed in such a way as to minimize the difference of the load drag among each multicopter. There are two parts of this path generation. The first part describes how each vehicle  $i$  should be located within the formation. The second describes how the entire formation should rotate as the load traverses the path.



**Figure 6.2:**  $\{l\}$ -frame. The position  $\mathbf{p}_i$  of each multicopter is parameterized by  $\mu_i$  and  $l_i$ .



**Figure 6.3:** Along-path frame. Here,  $a_t$  is along-path acceleration,  $a_n$  the normal component of the acceleration, and  $g$  is influence on gravity.

First, consider a load-centered load frame  $\{l\}$ . The orientation of this frame will be designed later, but this frame moves along a desired load path  $\mathbf{p}_L(\gamma)$ . Each UAV should follow a path that has a fixed offset in this frame, parameterized by two consecutive rotations  $l_i$  and  $\mu_i$  about the load frame  $z$  and  $y$  axis, respectively (See Figure 6.2). Thus, each vehicle position in  $\{l\}$  is given by

$$\rho_i^l := \mathbf{R}_z(l_i) \mathbf{R}_y(\mu_i) \begin{bmatrix} L_i \\ 0 \\ 0 \end{bmatrix} \quad (6.51)$$

where  $\mathbf{R}_\alpha(\cdot)$  is a principal rotation about an axis  $\alpha \in \{x, y, z\}$ , and  $L_i$  is the nominal length of wire  $i$ . Next, we describe the  $\{l\}$  frame. Let  $n_L(\gamma)$  be a unit vector that is designed to be parallel and in opposite direction to the total acceleration of the load as it travels along  $\mathbf{p}_L(\gamma)$ , as in Figure 6.3. In addition, we require that the

frame rotates along the tangential angle of the path. Let  $\mathbf{R}_l^n \in \mathcal{SO}^3$  describing the orientation of  $\{l\}$  with respect to  $\{n\}$  be parameterized by three consecutive rotations  $\psi_L, \phi_L$  and  $\theta_L$ :

$$\mathbf{R}_l^n(\Theta_L(\gamma)) := \mathbf{R}_z(\psi_L)\mathbf{R}_x(\phi_L)\mathbf{R}_y(\theta_L) \quad (6.52)$$

where  $\Theta_L := [\theta_L, \phi_L, \psi_L]^\top$  and

$$\psi_d(\gamma) = \text{atan2}(\mathbf{p}_{L,y}^\gamma, \mathbf{p}_{L,x}^\gamma) \quad (6.53)$$

$$\phi_L(\gamma) = \arctan \frac{\kappa(\gamma)v_L^2(\gamma)}{g} \quad (6.54)$$

$$\theta_L(\gamma) = \arctan \frac{v_L^\gamma(\gamma)}{g} \quad (6.55)$$

in which we require the path  $\mathbf{p}_L(\gamma)$  to be parameterized by its path length,  $v_L(\gamma)$  is the speed profile along the path and  $\kappa(\gamma)$  is the curvature.

The path of UAV  $i$  can thus be calculated from:

$$\mathbf{p}_i(\gamma) = \mathbf{p}_L(\gamma) + \mathbf{R}_i^n(\gamma)\rho_i^l \quad (6.56)$$

**Remark 6.1.** Note that the first and second derivatives of  $\mathbf{p}_{d,i}$  are required for path-following, which results in large and computationally expensive expressions. Depending on application, practical simplification such as assuming slowly varying speed reference  $v_L$  so we have  $v_L^{\gamma^2} \approx 0 \rightarrow [\phi_L^{\gamma^2}, \theta_L^{\gamma^2}] \approx 0$  will greatly simplify the computations.

### 6.3.1 Speed Profile

In addition to the spatial path introduced above, a speed profile to be tracked while the vehicles traverse the path is needed. The speed-profile should be  $C^2$  continuous and provide smooth velocity signals. In this work, we utilize a fifth order polynomial represented by Bernstein polynomials [32]. Further, we have four distinct phases, characterized by acceleration, constant speed, deceleration and stop, respectively. In the following, we will describe how to compute  $v_L(\gamma)$  and its derivatives in the acceleration phase. In the constant speed and stop phase, the values are trivial. The speed profile in the deceleration phase is computed in a similar manner as in the acceleration phase.

The acceleration phase is characterized by an initial speed  $V_0$ , a desired final speed  $V_f$ , and a rise time  $T$  (that is,  $v_L(0) = V_0$  and  $v_L(T) = V_f$ ). Note that when combining the speed profile with a spatial path,  $T$  will be a certain value of the path parameterization  $\gamma$ , that is, a speed at a certain point on the path. Then, we have

$$v_L(\gamma) = \sum_{k=0}^2 V_0 b_k^5 \left( \frac{\gamma}{T} \right) + \sum_{k=3}^5 V_f b_k^5 \left( \frac{\gamma}{T} \right) \quad (6.57)$$

where  $b_k^5$  is the fifth-order Bernstein basis defined by [32]:

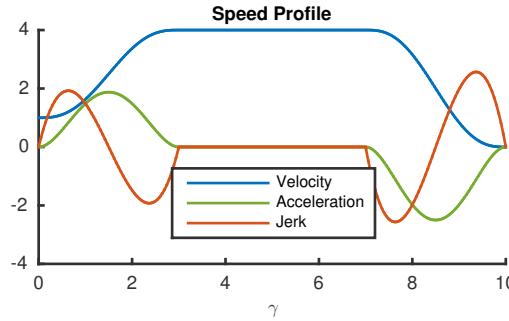
$$b_k^n(\tau) = \binom{n}{k} (1 - \tau)^{n-k} \tau^k \text{ for } k = 0, \dots, n \quad (6.58)$$

Its derivatives are readily given by

$$\frac{\partial v_L}{\partial \gamma} = \frac{5}{T} (V_f - V_0) b_2^4 \left( \frac{\gamma}{T} \right) \quad (6.59)$$

$$\frac{\partial^2 v_L}{\partial \gamma^2} = \frac{20}{T^2} (V_f - V_0) \left( b_1^3 \left( \frac{\gamma}{T} \right) - b_2^3 \left( \frac{\gamma}{T} \right) \right) \quad (6.60)$$

An example of a complete speed profile is given in Figure 6.4. Here, we have an initial speed of  $v_L(0) = 1$ , and require that  $v_L(3) = 4$ . The deceleration phase starts at  $\gamma = 7$ , before we have a stop at  $\gamma = 10$ .



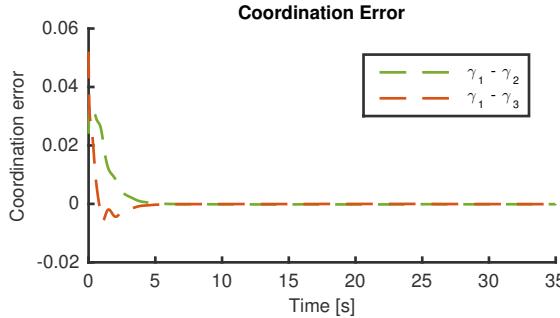
**Figure 6.4:** An example speed profile during acceleration, constant speed and deceleration phase.

## 6.4 Simulation Study

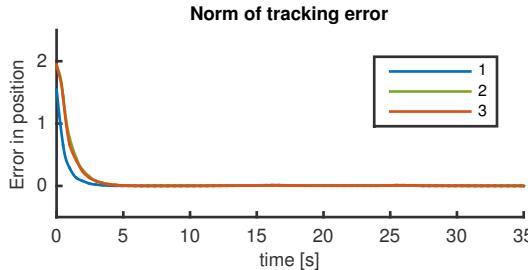
In this section, we apply the controllers developed in this chapter to a team of three multirotors transporting a suspended load along a circular path of radius  $R = 10$ . The paths for each vehicle is generated as in Section 6.3, with  $\mu_{1,2,3} = 60^\circ$ , and  $l_1 = 0, l_2 = 120^\circ, l_3 = -120^\circ$ . The simulations are conducted with the full nonlinear dynamics, including the constraint force from the payload, as developed in Section 6.1.2. Further, each UAV is equipped with a low-level attitude controller, as discussed in Section 6.2. In this simulation, the attitude controller is a simple PD structure. The maximum roll- and pitch-angle have been set to 30 degrees, to force reasonable realistic maneuvers. Each vehicle has a mass  $m = 4\text{kg}$ , transporting a payload of mass  $m_L = 1\text{kg}$ . Further, the speed profile is designed as in Section 6.3.1, with a maximum speed of 3 m/s, accelerating from 0.5 m/s at  $\gamma_i = 0$  to  $\gamma_i = 2R\pi/3$ , a third part around the circle. The control parameters was chosen as:  $\mathbf{K}_p = 5\mathbf{I}$ ,  $\mathbf{K}_d = 5\mathbf{I}$ ,  $k_\psi = 1$ ,  $k_\eta = 20$ ,  $\mathbf{K}_1 = \mathbf{K}_2 = 30\mathbf{I}$ . Integration was conducted by the *Runge-Kutta 4* method, at 100 Hz. Again, note that we assume that each multirotor can measure the drag from the suspended load to cancel it. Realistically, this can be achieved by an angular sensor combined with a force sensor, as demonstrated in

[13]. Further, we assume continuous communication links. However, the framework used in this chapter can easily be extended using the results in [109].

The results of the simulation can be seen in Figures 6.5–6.9. In Figure 6.5, the coordination error is illustrated. Clearly, it is seen that the error quickly converges to zero, and we have achieved coordination. Likewise, the path-following error is depicted in Figure 6.6 and shows convergence to a neighborhood around zero. The load drag on each multirotor is illustrated in Figure 6.7, and it is seen that the distribution is relatively uniform, except during the acceleration and deceleration phases. Further, in Figure 6.8, the total speed of the load as it traverses its path is shown. As can be seen, it reaches its desired speed of 3 m/s. Figure 6.9 provides snapshots of the simulation as it unfolds.



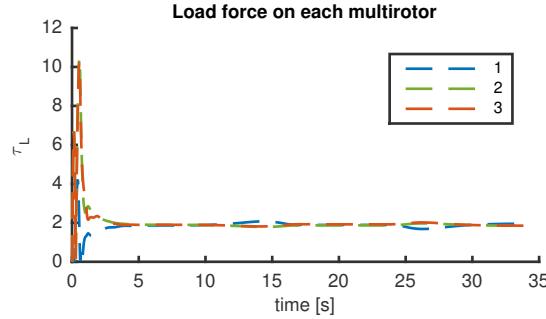
**Figure 6.5:** The coordination error during the simulation.



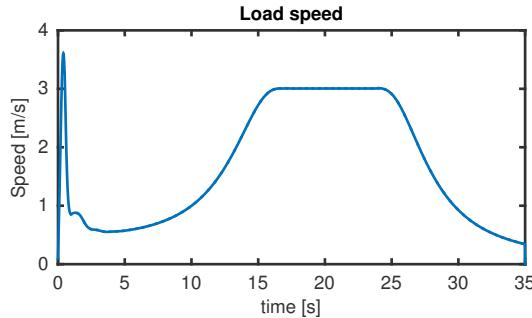
**Figure 6.6:** Path-following error of each multirotor. A small error is induced when the formation changes speed quickly; this is due to the performance of the low-level controllers.

## 6.5 Chapter Summary and Conclusions

In this chapter, we have designed a cooperative control law to transport a suspended load with a group of coordinating multirotors. The local path-following controller was analyzed by Lyapunov theory, and proven to solve the *path-following* problem. Coordination was achieved by applying results from the literature, most notably [3] and [41]. Further, we illustrated how to generate paths for each vehicle to minimize the variation of load drag among the vehicles to ensure smooth operations.

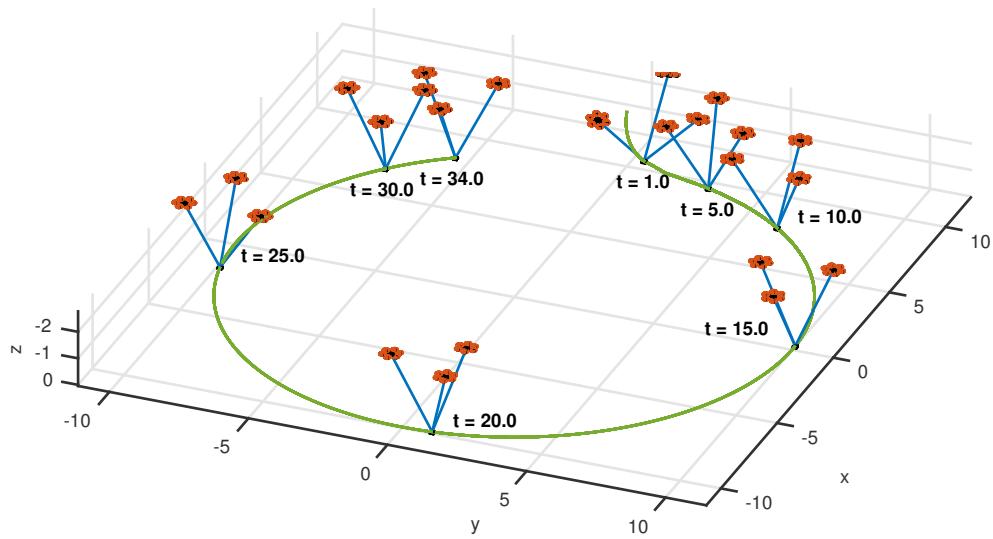


**Figure 6.7:** Load drag on each multirotor. The high initial values are due to initial movements of the vehicles, before they reach synchronization.



**Figure 6.8:** The total speed of the suspended load. As can be seen, it reaches its designated max speed at 3 m/s, before decelerating during the end of the circle movement.

Numerical results verified the results, where we showed convergence of the path-following- and coordination error. Robustness-properties with respect to external disturbances needs further investigation. Our suggested methodology to minimize the load variation showed decent results, but future work should look at improving this.



**Figure 6.9:** Snapshots of the simulation at different times, as marked in the figure.  
As can be seen, the multirotors achieve their desired formations.

## Chapter 7

# Passivity-Based Cooperative Control

This chapter is based on **C5** ([84] C. Meissen, K. Klausen, M. Arcak, T. I. Fossen, and A. Packard. Passivity-based Formation Control for UAVs with a Suspended Load. In *Proceedings of the IFAC World Congress*, 2017)

and **J3** ([69] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen. Cooperative Control for Multirotors Transporting an Unknown Suspended Load with Environmental Disturbances. *IEEE Transactions on Control Systems Technology (submitted)*, 2017).

### 7.1 Introduction

In Chapter 6, a cooperative path-controller was designed that let the group of multirotors synchronize their position along a predefined path. A practical challenge with this procedure is the necessity of a parameterized path, which is a non-trivial task to generate for complex operations. In addition, due to the method used for the technical proof, robust control with respect to environmental disturbances is hard to incorporate (however, certain types of adaptive compensation can be applied, see [55]).

In this chapter, a *position-based* formation control scheme is derived. In stead of synchronizing their location along a pre-defined path, the involved multirotor UAVs use directly the measured position of each other to achieve a desired relative formation. The design procedure is based on a *passivity-based* framework [10], which allows for adaptive extensions to unknown payload weight and disturbances.

The chapter is organized as follows. After some preliminaries in Section 7.1.1, two control laws are presented in Sections 7.3 and 7.5. In the first, we consider the suspended load being connected to a group of multirotor UAVs by means of flexible ropes modeled as springs. Integral action is considered to compensate for the unknown payload. The dynamics of the load through the spring wires onto the multirotors are considered in a technical stability proof, which establishes stability of the origin of the link errors. The next control law improves the first result showing convergence of the link errors, by adding a feed-forward term. In this design, the total disturbance from the suspended load and environmental forces such as wind are considered unknown, but the dynamics of the suspended load are

not considered in the technical stability proof. While both controllers are illustrated through numerical simulations, experimental trials was conducted for the latter in Section 7.7, where three multirotors (each weighing 2.2 kg) cooperatively lift a weight of 2 kg.

### 7.1.1 Preliminaries

This section defines the kronecker product, and introduces the concept of equilibrium-independent passivity. Let the kronecker product  $A \otimes B$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{p \times q}$  be a matrix  $\in \mathbb{R}^{mp \times nq}$  defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}. \quad (7.1)$$

The matrix satisfies  $(A \otimes B)^\top = A^\top \otimes B^\top$  and  $(A \otimes I_p)(C \otimes I_p) = (AC) \otimes I_p$ .

In order to prove stability of the systems described in this chapter we decompose it into subsystems and characterize the input-output properties of each individual subsystem by showing they are equilibrium independent dissipative (EID) [21, 52]. Consider a system of the form

$$\dot{x}(t) = f(x(t), u(t)) \quad y(t) = h(x(t), u(t)) \quad (7.2)$$

where there exists a nonempty set  $\mathcal{X} \subseteq \mathbb{R}^n$  such that for each  $\bar{x} \in \mathcal{X}$  there exists a unique  $\bar{u} \in \mathbb{R}^m$  satisfying  $f(\bar{x}, \bar{u}) = 0$ .

**Definition 7.1.** The system (7.2) is *EID* on an open set  $\mathcal{D} \subset \mathbb{R}^n$  with respect to a supply rate  $w$  if there exists a nonnegative storage function  $V : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_+$  such that  $V(\bar{x}, \bar{x}) = 0$  and

$$\dot{V}(x, \bar{x}) \triangleq \nabla_x V(x, \bar{x})^\top f(x, u) \leq w(u - \bar{u}, y - \bar{y}) \quad (7.3)$$

for all  $x \in \mathcal{D}$ ,  $u \in \mathbb{R}^m$  and for all  $\bar{x} \in \mathcal{D}$  and  $\bar{u}$  that satisfy  $f(\bar{x}, \bar{u}) = 0$  where  $y = h(x, u)$  and  $\bar{y} = h(\bar{x}, \bar{u})$ .

This definition ensures dissipativity with respect to any possible equilibrium point rather than a particular point. This is advantageous for compositional analysis, since the equilibrium of an interconnected system may be hard to compute.

Furthermore, a system is *equilibrium independent passive* if it is EID with respect to the supply rate

$$w(u - \bar{u}, y - \bar{y}) = (u - \bar{u})^\top (y - \bar{y})$$

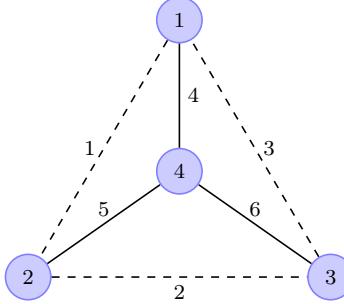
and it is *equilibrium independent output strictly passive* if it is EID with respect to the supply rate

$$w(u - \bar{u}, y - \bar{y}) = (u - \bar{u})^\top (y - \bar{y}) - \epsilon(y - \bar{y})^\top (y - \bar{y})$$

with  $\epsilon > 0$ .

## 7.2 System Description

We consider  $N$  UAVs that are cooperatively carrying a suspended load. The configuration of the UAVs and the load in the  $x - y$  plane are represented by an undirected graph as in Figure 7.1. We let  $\eta_i \in \mathbb{R}^3$ ,  $i = 1, \dots, N$  and  $\eta_{N+1}$  be the position of the UAVs and the suspended load, respectively.



**Figure 7.1:** Undirected graph of UAVs (1, 2, 3) and the suspended load 4.

For each dotted edge  $\ell = 1, \dots, T$  in the graph between UAVs we assign one vertex to be the head if it is clockwise from the other vertex. Therefore, for the configuration in Figure 7.1 vertex 1 is the head and vertex 2 is the tail along edge 1. Solid edges  $\ell = T + 1, \dots, E$  represent the cable between the UAV and the load. We assign the UAV vertex to be the head and the load vertex the tail. The incidence matrix, given by

$$M_{il} = \begin{cases} 1 & \text{if vertex } i \text{ is the head of edge } \ell \\ -1 & \text{if vertex } i \text{ is the tail of edge } \ell \\ 0 & \text{otherwise,} \end{cases} \quad (7.4)$$

will be used to characterize the interconnection topology.

Along each edge  $\ell = 1, \dots, E$  we define the relative position as  $r_\ell(t) := \eta_i(t) - \eta_j(t) \in \mathbb{R}^3$  where  $i$  and  $j$  correspond to the head and tail vertices, respectively, of the  $\ell$ -th edge.

Since the input and output of the UAVs and the suspended load is in three dimensions the matrix  $D := M \otimes I_3$  maps the position of the UAVs to the relative positions  $r_\ell(t) \in \mathbb{R}^3$  along each edge  $\ell = 1, \dots, E$  by

$$r = D^\top \eta. \quad (7.5)$$

As an example for the formation described in Figure 6.2 we have

$$M = \begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$$

and

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{bmatrix} = D^\top \eta = \begin{bmatrix} \eta_1 - \eta_2 \\ \eta_2 - \eta_3 \\ \eta_3 - \eta_1 \\ \eta_1 - \eta_4 \\ \eta_2 - \eta_4 \\ \eta_3 - \eta_4 \end{bmatrix}.$$

## 7.3 Design 1: Suspended Load Connected by Springs

### 7.3.1 System Dynamics

As described in Section 3.3, we assume a low-level attitude controller is present, and we can model the translational dynamics as:

$$m_i \dot{v}_i(t) = -m_i g + u_i^L(t) + \tau_i(t) \quad i = 1, \dots, N \quad (7.6)$$

with state  $v_i(t) \in \mathbb{R}^3$ , control input  $\tau_i(t) \in \mathbb{R}^3$ , mass  $m_i$ , and acceleration due to gravity  $g \in \mathbb{R}^3$ . The force applied to the UAV through the cable connected to the suspended load is  $u_i^L(t) \in \mathbb{R}^3$ . For each UAV an internal feedback controller will be used to compensate for this unknown force.

The dynamics of the suspended load are

$$m_L \dot{v}_{N+1}(t) = -m_L g + u_{N+1}(t) \quad (7.7)$$

with state  $v_{N+1}(t) \in \mathbb{R}^3$ , mass  $m_L$ , acceleration due to gravity  $g$ , and input  $u_{N+1}(t) \in \mathbb{R}^3$  which is the sum of the forces applied to the load by the UAVs through the cables.

The edges  $\ell = 1, \dots, T$  between UAVs do not represent a physical connection so in Section 8.3.2 a control law is proposed that acts as virtual springs oriented along the edges between the UAVs. The edges  $\ell = T + 1, \dots, E$  between the UAVs and the load represent a flexible cable. The force transferred along this cable is modeled by a function  $h_\ell : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  which takes the form

$$h_\ell(r_\ell) = \sigma_\ell(\|r_\ell\|) \frac{1}{\|r_\ell\|} r_\ell \quad (7.8)$$

where we assume  $\sigma_\ell : \mathbb{R}_+ \rightarrow \mathbb{R}$  is strictly increasing and onto for  $\ell = T + 1, \dots, E$ . This function can be interpreted as a spring acting between the UAV and the suspended load.

### 7.3.2 Control Strategy

In this section, we describe an internal feedback control for each UAV that renders it EID and compensates for the vertical force applied to it by the suspended load. Then, a formation control strategy is presented that regulates the relative position of the UAVs in the  $x$ ,  $y$ , and  $z$  coordinates independently.

### Internal Feedback Control

We propose a passivity based design where the internal feedback for each UAV is

$$\tau_i(t) = m_i g + v^d - v_i(t) - \delta_i(t) + u_i^f(t) \quad (7.9)$$

where  $m_i g$  compensates for the effect of gravity on the UAV,  $v^d$  is the desired velocity of the formation, and  $u_i^f$  is the formation control force which will regulate the relative positions of the UAVs as described in the next section. The component  $\delta_i(t)$  is updated by

$$\dot{\delta}_i(t) = v_i(t) - v^d \quad \delta_i(0) = \hat{\delta}_i \quad (7.10)$$

where  $\hat{\delta}_i$  is an estimate of the bias force applied to each UAV. The purpose of  $\delta$  is to compensate for the vertical force applied to each UAV by the suspended load as well as compensate for other unmodeled bias forces like wind.

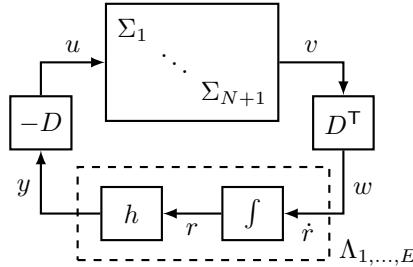
With this control strategy the UAV dynamics are

$$\begin{aligned} m_i \dot{v}_i(t) &= -v_i(t) + v^d - \delta_i(t) + u_i(t) \\ \dot{\delta}_i(t) &= v_i(t) - v^d \end{aligned} \quad (7.11)$$

where  $u_i = u_i^f + u_i^L$  is the sum of the formation control force  $u_i^f$  and the force from the suspended load  $u_i^L$ . In Section 7.3.3 we show that this choice of  $\tau$  guarantees that the system (7.11) is equilibrium independent output strictly passive from the input  $u_i$  to the output  $v_i$ .

### Formation Control

The interconnected system can be represented as the block diagram in Figure 7.2. The  $\Sigma_i$  subsystems mapping  $u_i$  to  $v_i$  are the UAV ( $i = 1, \dots, N$ ) and suspended load ( $i = N + 1$ ) subsystems and the  $\Lambda_\ell$  subsystems mapping  $w_\ell$  to  $y_\ell = h(r_\ell)$  for  $\ell = 1, \dots, E$  are the edge subsystems.



**Figure 7.2:** Block diagram of the vehicle platoon dynamics.

As depicted in Figure 7.2 we express the edge subsystems as

$$\begin{aligned} \dot{r}_\ell &= w_\ell \\ y_\ell &= h_\ell(r_\ell) \end{aligned} \quad (7.12)$$

where  $w \triangleq D^\top v$  is the input to the edge subsystems and  $y$  is the output. For the edges  $\ell = 1, \dots, T$  the functions  $h_\ell$  characterize the formation control strategy, while for the edges  $\ell = T + 1, \dots, E$  they model the cable connecting the UAVs and the suspended load.

The formation control strategy, which requires measurement of the relative positions of the UAVs, is described by the functions  $h_\ell : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  of the form

$$h_\ell(r_\ell) = \begin{bmatrix} \sigma_\ell^x(r_\ell^x) \\ \sigma_\ell^y(r_\ell^y) \\ \sigma_\ell^z(r_\ell^z) \end{bmatrix} \quad (7.13)$$

for each edge  $\ell = 1, \dots, T$  between UAVs. Similarly to (7.8) we assume that  $\sigma_\ell^x, \sigma_\ell^y, \sigma_\ell^z : \mathbb{R} \rightarrow \mathbb{R}$  are strictly increasing and onto for  $\ell = 1, \dots, T$ . This control law can be interpreted as three virtual springs for each edge connecting UAVs that act independently in each coordinate.

The input to the UAVs is then

$$u = -D \begin{bmatrix} h_1(r_1) \\ \vdots \\ h_E(r_E) \end{bmatrix} \quad (7.14)$$

where  $h_\ell$  are given by (7.13) for  $\ell = 1, \dots, T$  and by (7.8) for  $\ell = T + 1, \dots, E$ . Therefore, the input applied to the  $i$ -th subsystem is

$$u_i = - \sum_{\ell=1}^E D_{il} h_\ell(r_\ell) \quad (7.15)$$

which depends only on locally available measurements because  $D_{il} \neq 0$  only when vertex  $i$  is the head or tail of edge  $\ell$ .

### 7.3.3 Stability Analysis

We analyze the stability properties of the system and proposed control laws using a compositional approach. Specifically, for each subsystem (i.e. UAV, load, or edge) we find a storage function certifying that it is EID. From these storage functions we then obtain a Lyapunov function for the interconnected system.

The stability theorem from **C5** is repeated here, for details see Appendix B.

**Theorem 7.1.** *Any equilibrium point  $(\bar{v}, \bar{\delta}, \bar{r}) \in \mathcal{E}$  of the interconnected system in Figure 7.2 that satisfies  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$  for  $\ell = T + 1, \dots, E$  is stable.*

*Proof.* See Appendix B. □

## 7.4 Simulation Study

As an example, consider the configuration in Figure 7.1 with  $N = 3$  UAVs carrying the suspended load. Let each UAV have a mass  $m = 2$  kg, and the load have mass

$m_L = 3$  kg. In addition to the reaction force from the load, the UAVs are also affected by a constant wind in the  $x$ -direction with a magnitude of 4 m/s.

For  $N = 3$ , a fully connected graph between all UAVs consists of three links and the resulting control incidence matrix  $M_c$  is given by

$$M_c = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad (7.16)$$

Suppose we want the three UAVs to form a triangle with each side having length  $\Delta \in \mathbb{R}$ . We let  $\eta_i^d \in \mathbb{R}^3$  for  $i = 1, \dots, N$  represent the desired positions of the UAVs and assume that

$$\eta_1^d = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Then a set of possible desired positions of the UAVs are described by

$$\eta_2^d = \begin{bmatrix} \Delta \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \eta_3^d = \begin{bmatrix} \Delta/2 \\ \Delta\sqrt{3}/2 \\ 0 \end{bmatrix}.$$

The desired relative positions  $r_\ell^d \in \mathbb{R}^3$  are then given by

$$\begin{bmatrix} r_1^d \\ r_2^d \\ r_3^d \end{bmatrix} = (M_c \otimes I_3)^\top \begin{bmatrix} \eta_1^d \\ \eta_2^d \\ \eta_3^d \end{bmatrix}. \quad (7.17)$$

We then design  $h_\ell$  as described in Section 7.3.2. Specifically, we choose the formation control feedback function  $h_\ell$  for  $\ell = 1, \dots, T$  to be

$$h_\ell(r_\ell) = k(r_\ell - r_\ell^d) \quad (7.18)$$

with  $k = 8$ . Since  $k$  is positive  $h_\ell$  is strictly increasing and onto.

Note that without the suspended load the relative positions  $r_\ell$  of the UAVs would converge to the desired relative positions  $r_\ell^d$ . However, the force from the suspended load will pull the UAVs slightly closer together, so the equilibrium  $\bar{r}_\ell$  for the links between the UAVs will be slightly different from  $r_\ell^d$ .

The force between the load and the UAVs are modeled by Hooke's law as

$$\sigma_\ell(\|r_\ell\|) = \gamma(\|r_\ell\| - L_\ell) \quad (7.19)$$

for  $\ell = T + 1, \dots, E$ , where  $L_\ell = 2$  is the nominal length of the wire at link  $\ell$ . The wires are modeled as relatively stiff springs with  $\gamma = 100$ .

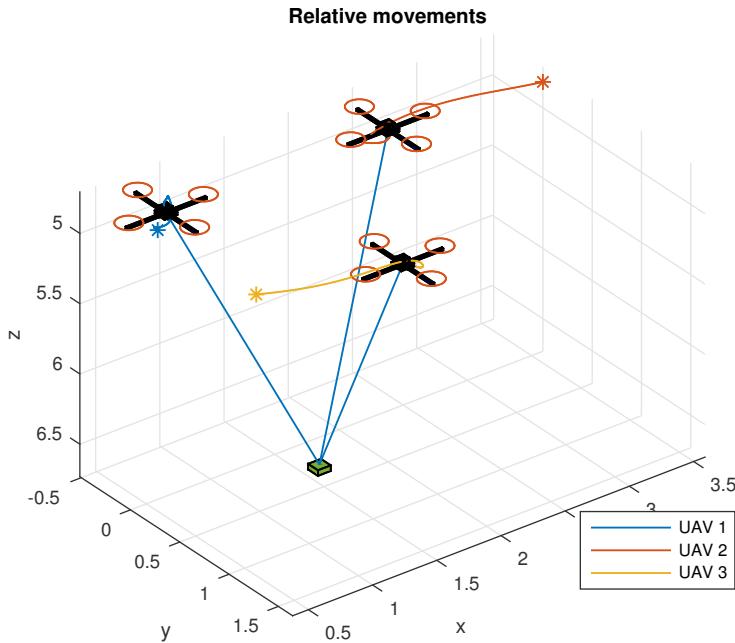
### 7.4.1 Results

We let the initial positions of the UAVs be

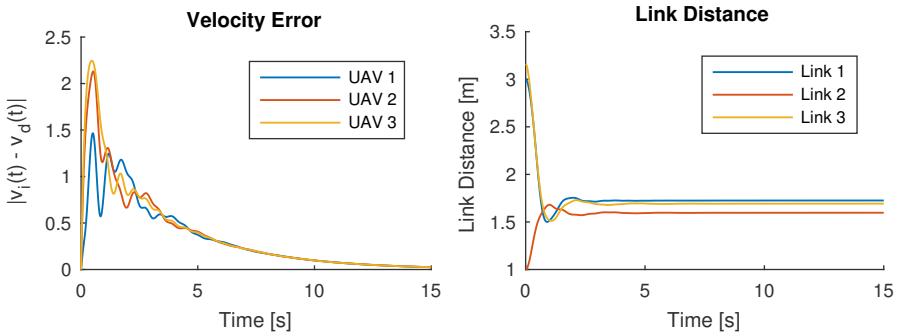
$$\eta_1^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \eta_2^0 = \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}, \quad \eta_3^0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

the desired formation be given by (7.17) with  $\Delta = 2$ , and the desired velocity be  $v^d = [0 \ 0 \ 0]^\top$ . The results of the simulation can be seen in Figures 7.3–7.6. The relative movements of the three UAVs can be seen in Figure 7.3, where the star represents the initial position, and the UAV drawing is the final position. As can be seen, they approach the desired relative formation, and the velocity error  $|v_i(t) - v^d(t)|$  converges to zero in Figure 7.4a. From Figure 7.4b we see that the distance between the UAVs converge to a constant value, but as expected, it is slightly less than  $\Delta$ .

Figure 7.6 shows estimates of the load mass and wind bias, which converge to the true values. The estimated load mass is calculated by summing the  $z$ -components of  $\delta_i$ , while the estimated wind bias is the average of the  $xy$ -components.

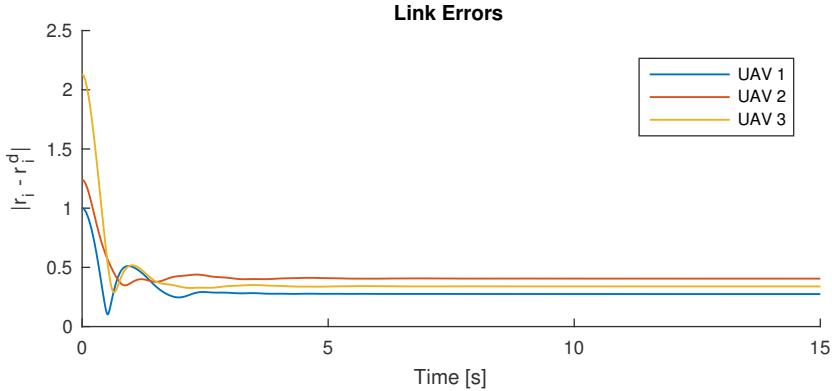


**Figure 7.3:** Position of the three UAVs and the suspended load.

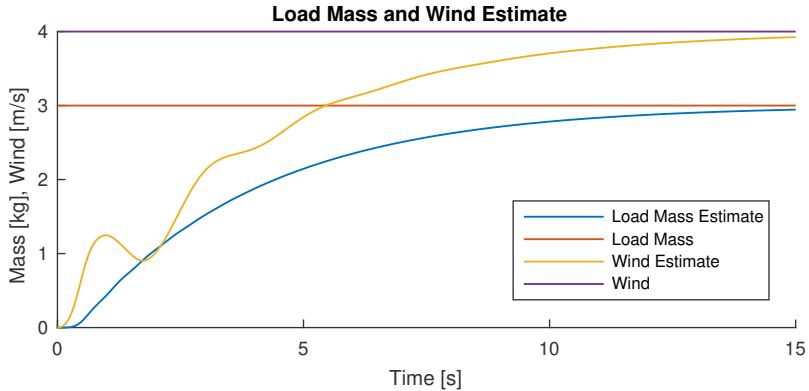


(a) The velocity error relative to the desired velocity  $v^d$  for each UAV.  
 (b) Distance between each UAV.

**Figure 7.4:** Velocity error and link distance from Design 1.



**Figure 7.5:** The link error between the UAVs from Design 1.



**Figure 7.6:** Load and bias estimates, as calculated from the integral terms  $\delta_i$  of each UAV.

## 7.5 Design 2: Applying Formation Feed Forward

In this design, we modify the control-structure presented in Section 7.3 by feeding forward the output from the formation controller,  $u$ , into the internal controller. In addition, we relax some of the conditions posed in the previous design:

- We allow for a time-varying mission velocity  $v(t)$
- A broader set of parametric uncertainties can be handled.

The cost of this is that we do not consider the dynamic of the load in the stability proofs, but rather consider the effect it has on the multirotors as an unknown bias.

### 7.5.1 System Dynamics

As in Section 7.3.1, we let the multirotor dynamics be governed by:

$$m_i \dot{v}_i = -m_i g + \tau_i + b_i^L + b_i^W \quad (7.20)$$

where  $x_i \in \mathbb{R}^3$  is the position of UAV  $i$ ,  $m_i$  its mass and acceleration due to gravity  $g \in \mathbb{R}^3$ . The suspended load affects multirotor  $i$  with an unknown drag  $b_i^L$ . Further, the vehicle is affected by a slowly varying wind with  $b_i^W$ .

In this design, we write the dynamics in the form

$$m_i \ddot{x}_i = \tau_i + Y_i \theta_i \quad (7.21)$$

where

$$Y_i := I \quad (7.22)$$

$$\theta_i := b_i^L + b_i^W \quad (7.23)$$

It is noted that due to the structure of the disturbances, the wind and load drag is lumped together in  $\theta_i$ . The system is written in this form to emphasize that the same design procedure listed here could be expanded to other types of parametric uncertainties of the form  $Y_i \theta_i$ , using the procedures illustrated in [10, Ch. 6].

Note however that in the following adaptive design, the disturbance from the load is considered a constant bias. However, as shown in Chapter 4, the magnitude of the disturbance changes as a function of the acceleration of the involved multirotors. Nonetheless, the assumption is valid for constant velocities, and only minor deviations occur when accelerating.

As in the previous design, we assume the UAVs are communicating with each other, where the communication topology is represented by a graph  $G$ . However, as we in this design do not consider the load dynamics explicitly, we only use the links from  $M$  that concerns the links between the UAVs,  $M^C$ . We similarly define  $D^C := M^C \otimes I_3$ . As an example, from the communication topology illustrated in Figure 7.1, the definition of  $M^C$  can be seen as:

$$M = \underbrace{\begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}}_{= M^C}$$

### 7.5.2 Control Strategy

The goal is to design a *decentralized* control structure for each UAV, where they utilize relative position and velocity from their neighbor to converge to a desired relative formation while maintaining a common mission velocity  $v(t) \in \mathbb{R}^3$  and correcting for the unknown disturbance from the wind and the suspended load. We are inspired by the *passivity based approach* introduced in [10], which defines the following desired group behaviors

*A1) Each agent achieves in the limit the common mission velocity vector as*

$$\lim_{t \rightarrow \infty} |\dot{x}_i - v(t)| = 0, \quad i = 1, \dots, N \quad (7.24)$$

*A2) If vehicles  $j$  and  $j'$  are connected by link  $k$ , then the difference variable  $z_k$*

$$z_k := \sum_{l=1}^N M_{lk}^C x_l = \begin{cases} x_i - x_j & \text{if vehicle } i \text{ is the head of edge } k \\ x_j - x_i & \text{if vehicle } i \text{ is the tail of edge } k \end{cases} \quad (7.25)$$

*converges asymptotically to  $z_k^d$  defined below.*

Let

$$\begin{aligned} x &:= [x_1^\top, \dots, x_n^\top]^\top \in \mathbb{R}^{3N} \\ z &:= [z_1^\top, \dots, z_\ell^\top]^\top \in \mathbb{R}^{3\ell} \end{aligned}$$

we can re-write (7.25) as

$$z = D^{c\top} x \quad (7.26)$$

Let now  $x^d \in \mathbb{R}^{3N}$  describe the desired relative positions between the UAVs.  $z^d$  is now given by

$$z^d = D^{c\top} x^d \quad (7.27)$$

### 7.5.3 Nominal case: Known disturbances

To illustrate the control methodology, consider first the case where the external disturbances (wind and force from the suspended load) is exactly known. They can subsequently be canceled by the controller. The method is similar to [10, Ch. 6].

#### Internal Feedback Control

We first design an *internal feedback* controller, designed to render the system *strictly passive* from an external input  $u_i$  to the velocity error

$$y_i := \dot{x}_i - v(t) - \Gamma_i u_i \quad (7.28)$$

where  $\Gamma_i = \Gamma_i^\top > 0 \in \mathbb{R}^3$  is a tuning matrix. In this design, we utilize the availability of relative velocities gathered from each agents, through the feed-forward of  $u_i$ .

This can be achieved with the control law

$$\begin{aligned} \tau_i^{nominal} &= -k_i(\dot{x}_i - v(t) - \Gamma_i u_i) \\ &\quad + m_i(\dot{v}(t) + \Gamma_i \dot{u}_i) \\ &\quad + m_i g + u_i - Y_i \theta_i \end{aligned} \quad (7.29)$$

where  $k_i > 0 \in \mathbb{R}$ , and by introducing

$$\xi_i = \dot{x}_i - v(t) - \Gamma_i u_i \quad (7.30)$$

yields the system to the form

$$\dot{x}_i = y_i + v(t) + \Gamma_i u_i \quad (7.31)$$

$$\mathcal{H}_i : \begin{cases} m_i \dot{\xi}_i = -k_i \xi_i + u_i \\ y_i = \xi_i \end{cases} \quad (7.32)$$

The strict passivity from  $u_i$  to  $y_i$  can be shown with the storage function

$$S_i(\xi) = \frac{1}{2} m_i \xi_i^\top \xi_i \quad (7.33)$$

### Formation Control

Next, the external control signal  $u_i$  is designed to stabilize the formation. Consider

$$u_i = - \sum_{k=1}^{\ell} M_{ik}^c \Psi_k(z_k) \quad (7.34)$$

where

$$\Psi_k(z_k) = \kappa_k(z_k - z_k^d) \quad (7.35)$$

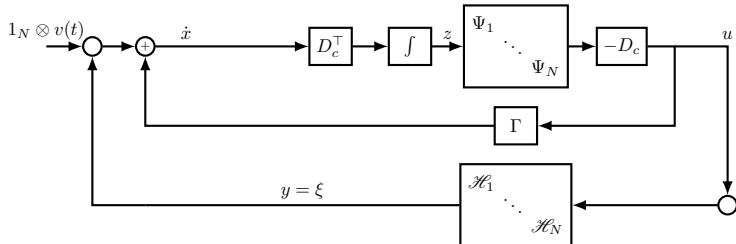
where  $\kappa_k = \kappa_k^\top > 0 \in \mathbb{R}^3$  is a diagonal matrix of gains.  $\kappa$  represents the diagonal concatenation of all  $\kappa_k$ . It should be noted that the definition of  $u_i$  in (7.34) is of the same format as used in the simulation study of Section 7.4 in the first design.

To clarify notations, let  $\tilde{z}_k$ , and its corresponding stacked vector  $\tilde{z}$ , be defined as

$$\tilde{z}_k := z_k - z_k^d, \quad \tilde{z} := z - z^d \quad (7.36)$$

$u$  can now be re-written as

$$u = -D^c \kappa \tilde{z} \quad (7.37)$$



**Figure 7.7:** Nominal system, with known disturbances.

The interconnected system can be seen in Figure 7.7. To prove the stability, we utilize the passivity-properties of the feedforward and the feedback path in the following lemma.

**Lemma 7.2.** *The path from  $y$  to  $-u$  is passive, and the feedback is passive from  $u$  to  $y$ . Due to the feedback connection of two passive subsystems as seen in Figure 7.7, the connected system is passive and the origin of  $(\tilde{z}, \xi)$  is globally asymptotically stable.*

*Proof.* i) Feedforward path: We first prove passivity from  $y$  to  $-u$ . Let  $V_f$  be a storage function:

$$V_f(\tilde{z}, t) = \frac{1}{2} \sum_{\ell} \tilde{z}_k^\top \kappa_k \tilde{z}_k \quad (7.38)$$

whose derivative along the solutions of

$$\dot{\tilde{z}} = D^{c\top} \dot{x} \quad (7.39)$$

is

$$\dot{V}_f(\tilde{z}, t) = \tilde{z}^\top \kappa D^{c\top} (\dot{x}) \quad (7.40)$$

$$= \tilde{z}^\top \kappa D^{c\top} (y + 1_N \otimes v(t) + \Gamma u) \quad (7.41)$$

$$= \tilde{z}^\top \kappa D^{c\top} (y + \Gamma u) \quad (7.42)$$

$$= -u^\top \Gamma u - u^\top y \quad (7.43)$$

which shows strict passivity from  $y$  to  $-u$ .

ii) Feedback path: Next we use the storage function

$$V_b(\xi, t) = \sum_N S_i(\xi_i) = \sum_N \frac{1}{2} m_i \xi_i^\top \xi_i \quad (7.44)$$

to show passivity from an input  $-u$  to  $y$ :

$$\dot{V}_b(\xi, t) = \sum_N -k_i \xi_i^\top \xi_i + u^\top y \quad (7.45)$$

$$= -\xi^\top K \xi + u^\top y \quad (7.46)$$

where  $K = (k_i \otimes I_3) \otimes I_N$ , which shows passivity from  $u$  to  $y$ .

iii) To show asymptotic stability of the interconnected path, consider

$$V_n(\tilde{z}, \xi) = V_f(\tilde{z}) + V_b(\xi) \quad (7.47)$$

which gives

$$\dot{V}_n(\tilde{z}, \xi) = -u^\top \Gamma u - \xi^\top K \xi \quad (7.48)$$

$$= -\tilde{z}^\top \kappa D^{c\top} \Gamma D^c \kappa \tilde{z} - \xi^\top K \xi \leq 0 \quad (7.49)$$

which is negative semi-definite since  $D^{c\top} \Gamma D^c \leq 0$ . Note that  $\dot{V}_n = 0$  when  $D^c \kappa \tilde{z} = 0$ . This is when  $\kappa \tilde{z}$  is in the null space of  $D^c$ . By construction,  $\tilde{z}$  is in the range space of  $D^{c\top}$ . Thus,  $\tilde{z}$  is orthogonal to  $\kappa \tilde{z}$  since the range space of  $D^{c\top}$  is orthogonal to the null space of  $D^c$ , and  $\tilde{z}^\top \kappa \tilde{z} = 0$  which implies  $\tilde{z} = 0$ . We then apply the Invariance principle [61, Th. 4.4], and since  $V_n(\tilde{z}, \xi)$  is unbounded and satisfies  $V_n(0, 0) = 0$ , we have shown Global Asymptotic Stability (GAS) of the nominal system.

□

### 7.5.4 With disturbances

#### Internal Feedback Control

Now, consider the case when  $\theta_i$  is unknown. In our controller, we now supply our estimate  $\hat{\theta}$ :

$$\begin{aligned}\tau_i = & -k_i(\dot{x}_i - v(t) - \Gamma_i u_i) \\ & + m_i(\dot{v}(t) + \Gamma_i \dot{u}_i) \\ & + m_i g + u_i - Y_i \hat{\theta}_i\end{aligned}\quad (7.50)$$

and let the update-law for  $\hat{\theta}_i$  be governed by:

$$\dot{\hat{\theta}}_i = \Lambda_i Y_i \xi_i \quad (7.51)$$

where  $\Lambda_i = \Lambda_i^\top > 0 \in \mathbb{R}^3$ , which brings the system to the form

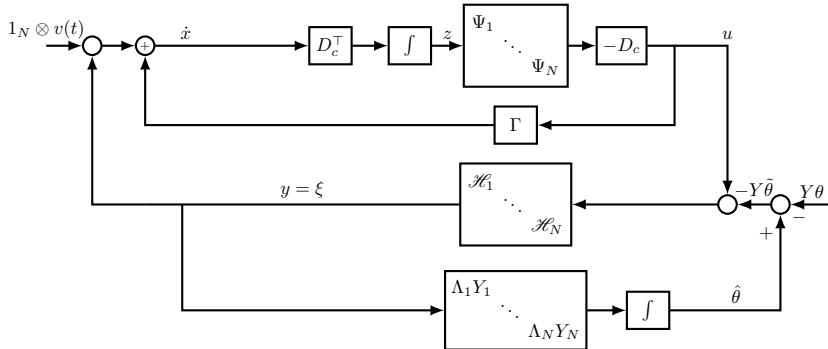
$$\dot{x}_i = y_i + v(t) + \Gamma_i u_i \quad (7.52)$$

$$\mathcal{H}_i : \begin{cases} m_i \dot{\xi}_i = -k_i \xi_i + u_i + Y_i \tilde{\theta}_i \\ y_i = \xi_i \end{cases} \quad (7.53)$$

where

$$\tilde{\theta}_i = \hat{\theta}_i - \theta_i \quad (7.54)$$

The *formation control* is identical to the nominal known disturbance case, and the final system can be seen in Figure 7.8.



**Figure 7.8:** System interconnection with unknown disturbances  $\theta$  acting on the system.

### 7.5.5 Stability Analysis

We first illustrate that the initial subsystem is passive from an input  $Y \tilde{\theta}$  to  $y$  (see Figure 7.8), and then conclude asymptotic stability of the interconnected system:

**Lemma 7.3.** *The nominal system is passive from an input  $Y\tilde{\theta}$  to  $y$ .*

*Proof.* Using  $V_a(\tilde{z}, \xi) = V_f(\tilde{z}) + V_b(\xi)$  as a storage function, we obtain

$$\dot{V}_a = -\tilde{z}^\top \boldsymbol{\kappa} D^c \Gamma D^c \boldsymbol{\kappa} \tilde{z} - \xi^\top K \xi - \xi^\top Y \tilde{\theta} \quad (7.55)$$

which shows strict passivity from  $Y\tilde{\theta}$  to  $y$  [10].  $\square$

Now, with the interconnected adaptive term, we state the main theoretical result of this chapter:

**Theorem 7.4.** *Consider  $N$  multirotors carrying a suspended load, modeled as (7.21), with the control input (7.50) and  $\tilde{\theta}_i$  is updated as in (7.51). Then, the origin of  $(\tilde{z}, \xi, \tilde{\theta})$  is stable, and the trajectories of  $(\tilde{z}, \xi, \tilde{\theta})$  are bounded. Furthermore,  $|\dot{x}_i - v(t)| \rightarrow 0$  and  $\tilde{z}_k \rightarrow 0$ .*

*Proof.* Let  $V(\tilde{z}, \xi, \tilde{\theta})$  be a Lyapunov function, defined by

$$V(\tilde{z}, \xi, \tilde{\theta}) = V_a(\tilde{z}, \xi) + \frac{1}{2} \sum_N \tilde{\theta}_i^\top \Lambda_i^{-1} \tilde{\theta}_i \quad (7.56)$$

which satisfies  $V(\tilde{z}, \xi, \tilde{\theta}) > 0$  and  $V(0) = 0$ , whose derivate along the solutions of the system are

$$\dot{V}(\tilde{z}, \xi, \tilde{\theta}) = -\tilde{z}^\top \boldsymbol{\kappa} D^c \Gamma D^c \boldsymbol{\kappa} \tilde{z} - \xi^\top K \xi \leq 0 \quad (7.57)$$

which implies stability of the origin and boundedness of  $(\tilde{z}, \xi, \tilde{\theta})$ . From (7.53) we have boundedness of  $\dot{\xi}_i$ , and boundedness of  $\dot{\tilde{z}}$  is concluded from (7.37) and (7.52). Thus, we conclude that  $\ddot{V}$  is bounded. Finally, we apply Barbalat's lemma [61, Th. 8.4], and from (7.52) arguments similar to Lemma 7.2, conclude  $|\dot{x}_i - v(t)| \rightarrow 0$  and  $\tilde{z}_k \rightarrow 0$ .  $\square$

Note that Theorem 7.4 is a special case of [10, Th. 6.3]. Because of the case studied in this chapter, convergence of  $\theta_i$  can be shown since  $Y_i$  in (7.22) is persistently exciting [57].

### 7.5.6 Design Comparisons and Discussions

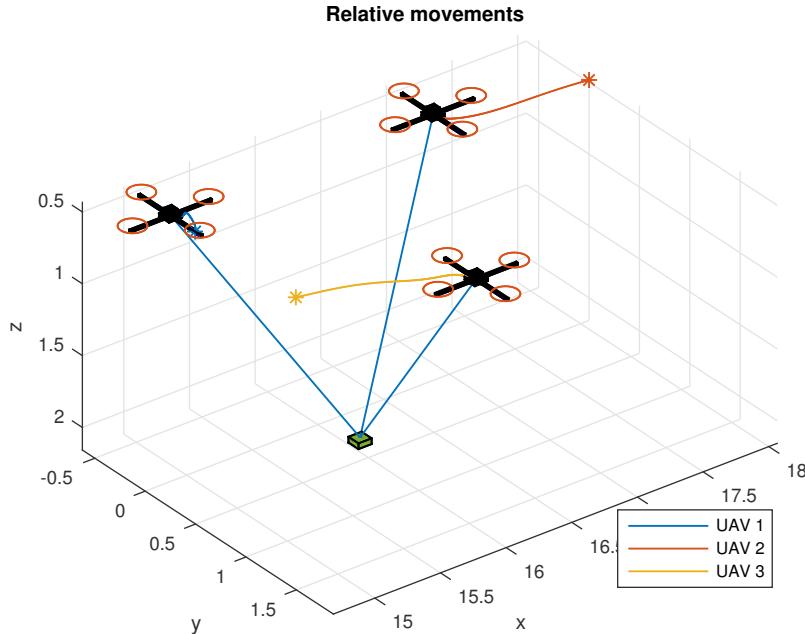
It should be obvious to the reader that Design 2 is, practically speaking, superior to that of Design 1 as it actually shows convergence of the link errors. However, Design 1 illustrates an important theoretical exercise in that it also includes the dynamics of the load in the analysis. A similar analysis for Design 2 has not yet been completed, but preliminary work suggests that similar results can be shown.

As was seen in Chapter 4, the dynamic coupling between the suspended load and the coordinating multirotors means that the acceleration of any vehicle has a direct impact on the resulting wire-tension on all the other multirotors. Therefore, the assumption that the load can be viewed as a constant disturbance is technically only valid for constant velocity segments, even though the controller itself shows stability in the case of a non-constant mission velocity.

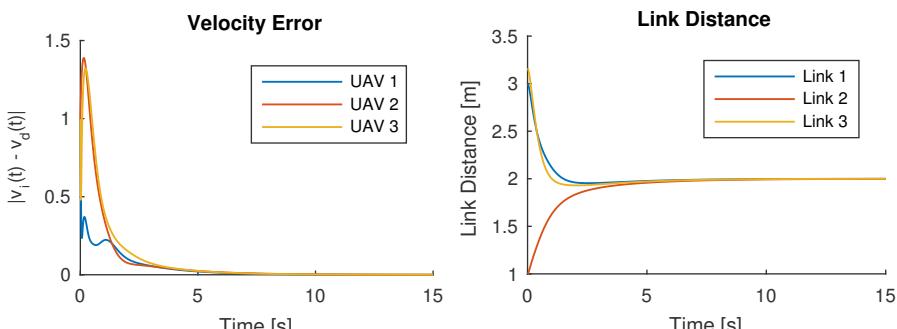
In the next section, a numerical simulation study illustrates the proposed controller.

## 7.6 Simulation Results

We re-do the experiments from Section 7.4, where we also include a constant desired mission velocity, that tends the multirotors to carry the load at 1 m/s in the positive x-direction. As can be seen in Figure 7.9 and Figure 7.10b, the multirotors tends towards the desired formation. Relative to Figure 7.4b in Design 1, we see that the links converge to their actual desired values in Figure 7.11. Further, in Figure 7.12 we see as before that the wind and load estimates converge to their true values.



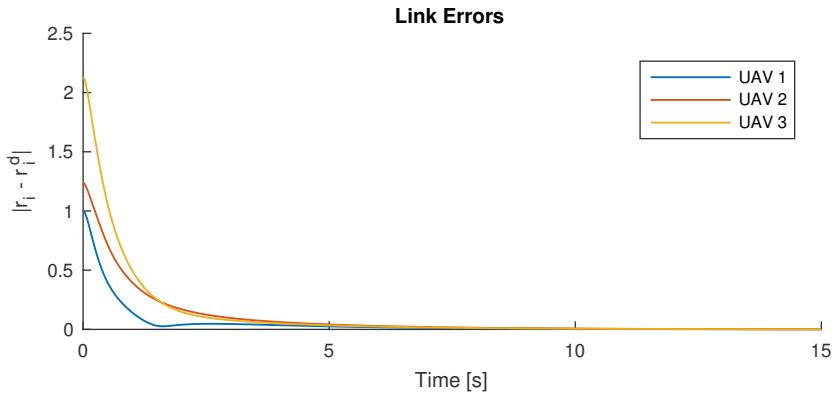
**Figure 7.9:** Position of the three UAVs and the suspended load.



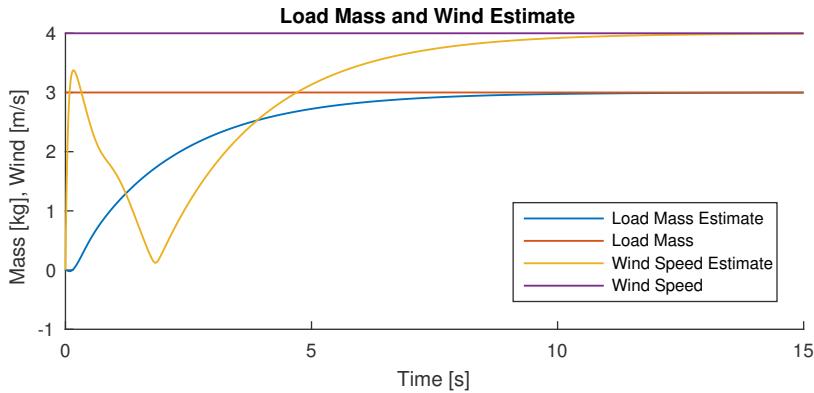
(a) The velocity error relative to the desired velocity  $v^d$  for each UAV.

(b) Distance between each UAV.

**Figure 7.10:** Velocity error and link distance from Design 2.



**Figure 7.11:** The link error between the UAVs from Design 2.



**Figure 7.12:** Load and bias estimates, as calculated from the integral terms  $\delta_i$  of each UAV for Design 2.

## 7.7 Experimental Validation

To further validate the proposed controller, experimental trials with three multirotors carrying a suspended load was conducted. A detailed description of the hardware-platform was given in Chapter 2.

For the experiment, Design 2 was implemented. The desired relative position was given by the initial position on the ground, which was set to an equal-sided triangle, with sides of 5 m. This is illustrated in Appendix A, Figure A.1. The control software was implemented such that a single pilot can perform coordinated takeoff, and use a remote controller to set reference velocities and accelerations during the test. An image of the three multirotors successfully transporting a suspended load is given in Figure 7.13.

It should be noted that as it is not possible to attach the load at the centre of gravity of the UAVs, the frame is subjected to a slight moment-arm from the load. However, the integral effect of the internal attitude controller on the autopilot is capable of suppressing this disturbance.

Two main tests was conducted. In the first, a lighter load of  $\sim 1$  kg was used,

and in the second a heavier load of  $\sim 2.2$  kg.



**Figure 7.13:** Image of the experiment underway. From left to right, we see multirotors 1, 3 and 2. The wind was heavily blowing from right to the left, so it is visible that the rightmost multirotor got more drag from the load.

In Figures 7.14–7.15 a summary of the two experiments are illustrated. In the figures, the upper plot shows the sum of link-error norms ( $\sum |\tilde{z}_k|$ ), the middle shows the norm of acceleration setpoint ( $|\dot{v}^d(t)|$ ), and the lower shows the norm of the bias term for all multirotors. It can be seen that in both cases the controllers are able to maintain a relative close formation, but during heavy accelerations a somewhat larger error is seen. However, it is quickly recovered once the acceleration-phase is over.

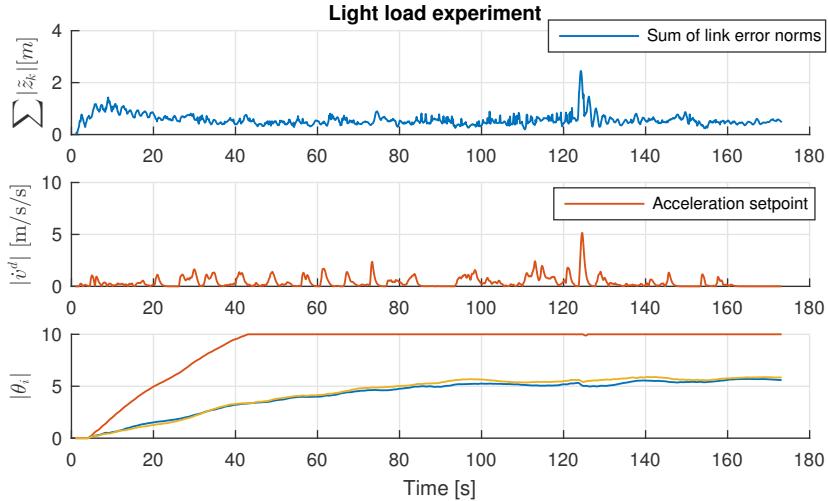
Due to heavy winds on the test-day (estimated 4-6 m/s from the right to the left as shown in Figure 7.13), the rightmost multirotor (UAV 2) has an increased load due to air drag on the suspended load. This is also seen by the lower plots in Figures 7.14–7.15. As can also be seen, the bias term saturates at a maximum level, which unfortunately was set to a too low value during the experiments.

**Table 7.1:** Mean link error over the two experiments.

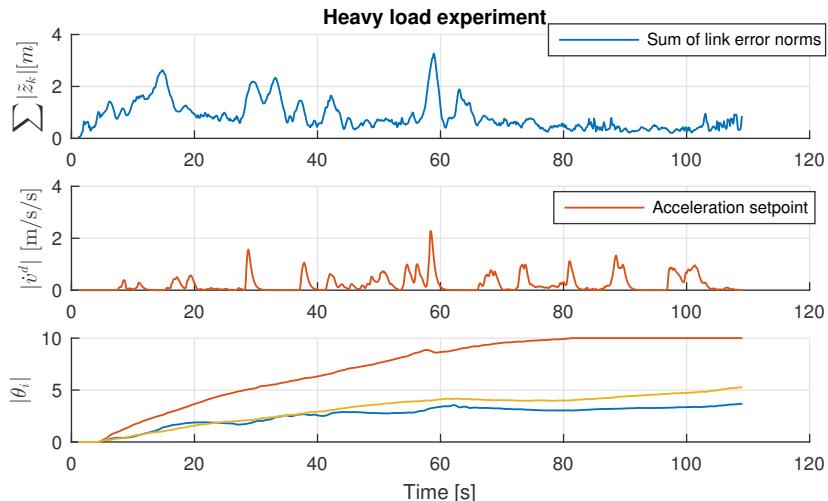
Mean Link Error [m]	Light load	Heavy load
Link 1	0.38	0.57
Link 2	0.37	0.34
Link 3	0.16	0.46

Nonetheless, as can be seen in Figure 7.16, the link errors remain relatively small during the entire experiment. A summary of the *mean link errors* are provided in

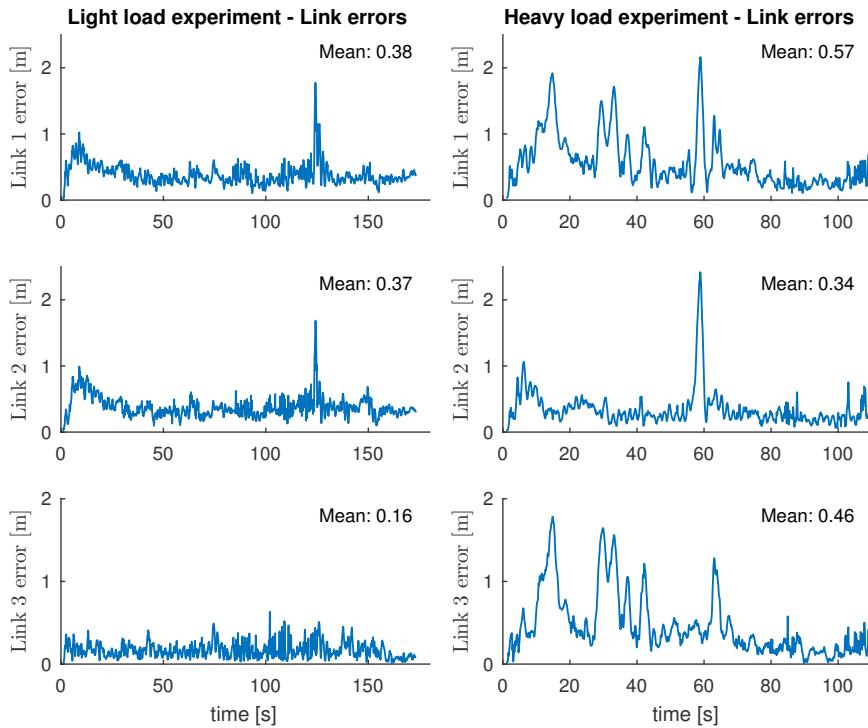
Table 7.1. It is seen that for the light experiment, the mean errors are below 0.4 m, while for the heavier load they are slightly higher.



**Figure 7.14:** Summary of the first (light) experiment. The lower plots shows the norm of bias estimation for the first (blue), second (red) and third (green) multirotor UAV.



**Figure 7.15:** Summary of the second (heavy) experiment. The lower plots shows the norm of bias estimation for the first (blue), second (red) and third (green) multirotor UAV.



- (a) Norm of the link errors for links 1-3 for the experiment with the lighter load.  
 (b) Norm of the link errors for links 1-3 for the experiment with the heavier load.

**Figure 7.16:** The position error (link error) between the multirotors for the light load (a) and heavy load (b) experiment.

## 7.8 Chapter Summary and Conclusions

In this chapter, we have looked at a position-based cooperative control strategy inspired by a passivity framework. Two designs were discussed, where the former explicitly considered the load dynamics in the analysis while the latter considered the load an unknown disturbance. Stronger convergence properties were established for Design 2 (asymptotic stability), due to an additional feed-forward term of the formation control output into the internal controller.

The latter design was further verified by two experiments, where three multirotor UAVs cooperatively lifted an unknown payload of up to 2.2 kg. Throughout the experiment, the average link errors remained below 0.6 m for the heavier load, and below 0.4 m for the lighter.

## Chapter 8

# Net-Recovery of a Fixed-Wing UAV using Multirotors

This chapter is based on **C4** ([66] K. Klausen, J. B. Moe, J. C. Van Den Hoorn, A. Gomola, T. I. Fossen, and T. A. Johansen. Coordinated Control Concept for Recovery of a Fixed-Wing UAV on a Ship using a Net Carried by Multirotor UAVs. In *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 964–973, 2016. ISBN 9781467393331. doi: 10.1109/ICUAS.2016.7502640)

and **J2** ([68] K. Klausen, T. I. Fossen, and T. A. Johansen. Autonomous Recovery of a Fixed-Wing UAV Using a Net Suspended by Two Multirotor UAVs. *Journal of Field Robotics (submitted)*, 2017).

### 8.1 Introduction

Fixed-Wing Unmanned Aerial Vehicles have a vast range of use-cases. Mostly, these encompass surveillance or other form of remote sensing. The fixed-wing UAV is the natural choice for a UAV in many of these cases, due to their long range, high endurance and flexible payload capacity.

The maritime sector has a need to operate these types of UAVs. They would like to use UAVs for ice surveying and monitoring, scan for unidentified vessels, search and rescue, and to provide extended range for survey-missions normally conducted from shore. Key challenges that needs to be addressed for autonomous operations are takeoff and landing (recovery) of the UAV. For takeoff, the use of catapults, powered by pneumatics, springs or rubber bands, has proven to be a useful tool that is applicable to many types of airframes and operation-types. Autonomous recovery is, however, an active area of research.

Several aspects of autonomous landing of fixed-wing UAVs are discussed in the literature. In [112], a navigation system with emphasis on robustness and accuracy is presented by utilizing dual-frequency, carrier phase differential GNSS. The authors also proposes solutions to some of the challenges associated with local antenna arrays for jamming prevention and fault monitoring strategies. Autonomous landing in a fixed-net is demonstrated in [122], using lower cost single frequency carrier-phase

RTK GPS with successful experimental validation. [89] uses a combination of GNSS and vision systems to reliably land a fixed-wing UAV on a moving car, intended to be used as a recovery platform for larger high altitude, long endurance UAVs without heavy landing gear. Pure vision-based navigation systems for autonomous fixed-wing UAV landing is tried in [134], [71] and [62].

Recovery of Fixed-Wing UAVs from ships require different methodologies from those performed on shore. There are several challenges associated with such operations, including heave-induced motion from waves, turbulence near the ship, and the safety of the crew operating on deck. In addition, there is no room for a runway unless it is a ship dedicated to such operations. Instead, most operations rely on the dissipative effects of a fixed net or wires and hooks to stop the incoming UAV. Vertical Take-Off and Landing (VTOL) UAVs does not require such infrastructure, but still requires accurate positioning and tracking to be able to reliably land on moving targets such as ships. Such operations are addressed in [111], [140] and [115], using vision system for navigational aid during the final descent.



**Figure 8.1:** Image from the experimental validation. The fixed-wing UAV is approaching the net, which is being carried by two multirotor UAVs. On the right, the fixed-wing UAV is arrested by the net, and stays attached due to hooks on the nose of the UAV.

In this chapter, we present a novel approach for landing a small fixed-wing UAV for maritime missions. By suspending a net below two multirotor UAVs, the entire recovery operation can be moved off deck. The fixed-wing UAV is equipped with hooks so that after impact with the net, it will be arrested by the net to be transported back to the ship. Figure 8.1 shows the recovery system in action, and Figure 8.2 illustrates the sequence of events when recovering a fixed-wing UAV from a ship.

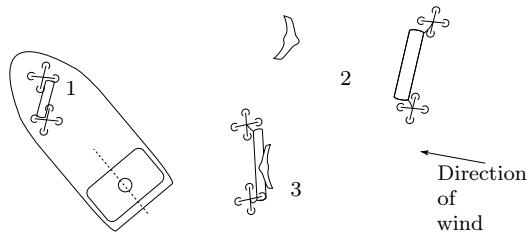
Recovery of UAVs with nets suspended by multirotors have been attempted in various settings. Due to the popularity of consumer-type multirotor UAVs, there is an increased interest in the ability to safely remove such vehicles from restricted airspaces. In [25], a multirotor is equipped with a net gun, capable of incapacitating smaller multirotors by shooting a net at them to disable the rotors on the target. A similar experiment was conducted in [88], where the target multirotor stays attached to the larger multirotor after the net is fired, see also [132], [90]. To the best of the

authors knowledge, no attempts at cooperative recovery of fixed-wing UAVs in a net suspended between multirotor UAVs have yet been published.

The contributions of this chapter are the experimental validation and detailed functional description of the presented net-recovery strategy. The experiments were conducted in December 2016, using a small fixed-wing UAV (1.5 kg). An image of the experiment under-way can be seen in Figure 8.1.

### 8.1.1 Organization

The chapter is organized as follows. In Section 8.2, the autonomous net recovery is introduced, and key aspects and advantages are discussed. Further, the definition of the *virtual runway* is given in Section 8.2.4. Section 8.3 presents the proposed control design, which includes coordination controllers between the multirotor UAVs and the incoming fixed-wing UAV. The experimental platform is given in Section 8.4, which in addition to introducing the hardware and software, also details the operational procedure and safety considerations considered when performing the experiments. The results of the trials are presented in Section 8.5, followed by concluding remarks in Section 3.9.



**Figure 8.2:** The figure illustrates the recovery of a fixed-wing UAV. (1), the multirotors take off from the ship. (2), the fixed-wing UAV moves against the wind direction, while the multirotors position the net along a virtual runway (coinciding with the fixed-wing UAV trajectory) and accelerate to a prescribed velocity in order to catch the incoming fixed-wing UAV. (3), the multirotors are transporting the fixed-wing UAV back to the ship.

## 8.2 Autonomous Net Recovery

In this section, the concept of UAV recovery using a net suspended below multirotors is discussed in further details. As described in [66] in the context of ship-based UAV operations, key benefits of the proposed approach are

- *Operational flexibility:* When recovering a fixed-wing UAV, it is usually beneficial to travel against the wind to minimize the ground speed, and thus a fixed net should be aligned with this path. Even in vessels equipped with Dynamic Positioning (DP) systems, turning the ship can be undesirable as it may interfere with operations. The multirotors can however quickly react to changing wind conditions, and align the net against the wind without the need to coordinate with other ship operations.

- *Not affected by waves and turbulence:* Since the net is suspended away from the ship, heave motion induced by waves on the ship will not affect the landing. Also, there is no impact from turbulence caused by the ship super-structure.
- *Safety:* By having the net suspended by two multirotor UAVs, the recovery operation can be moved off ship. Thus, no operators or staff risk coming in contact with the incoming UAV.
- *Smaller impact force:* By having the two multirotors move away from the fixed-wing UAV, the relative speed difference between it and the net can be made smaller, thus decreasing the structural load on the fixed-wing body during impact.
- *Smaller footprint:* By moving the landing operation off ship, operations with UAVs can be conducted from smaller ships, not needing a large open deck with a net to support the mission. Launch and recovery of the multirotors are still required.

These benefits are useful also in other recovery scenarios where there is limited space available.

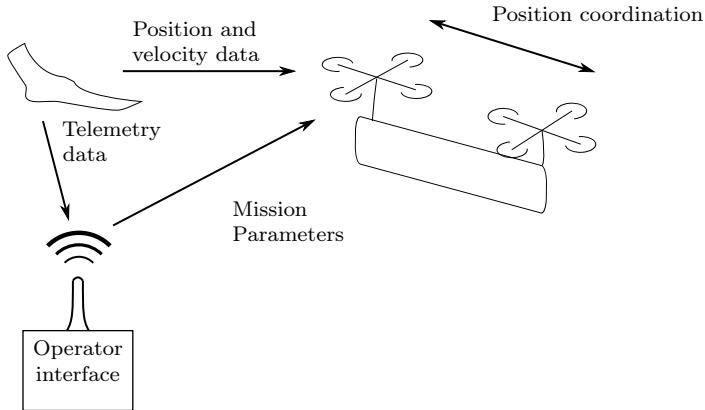
Autonomous recovery of a fixed-wing UAV in a suspended net is a complex task, so the functionality is split into several components. The fixed-wing UAV is commanded to follow a path against the wind, with the minimal airspeed required for safe flying. This path is called the *virtual runway*, and the path is transmitted to the multirotor UAVs. Both multirotors are equipped with coordinated controllers that keep the inter-formation of the two intact, while lifting the suspended net. The current position and the velocity of the fixed-wing UAV is transmitted in real-time at regular intervals to a coordination controller in one of the multirotors, which sends desired setpoints to the formation controllers according to the phases of the mission, as to catch the fixed-wing UAV. An overview of the components and their communication can be seen in Figure 8.3.

Although using two multirotor UAVs instead of one increases the complexity of the system, it has several practical advantages. First, by distributing the load, each multirotor can be physically smaller than a single multirotor with the combined lift capacity. This can lead to simplified operations, and makes it easier to have redundancy in the equipment. Further, although not utilized in this research, the two multirotors can spread the net without a support structure (top beam), giving reduced weight and high stability of the net.

### 8.2.1 The Recovery Maneuver

The maneuver is composed by the following key steps:

1. **Initialization:** The operator specifies the location and direction of the desired recovery point by defining the *Virtual Runway* (VR). This runway defines the desired path of the incoming fixed-wing UAV. The operator defines the safety area around the runway (that is, how far away from the center line can the multirotors go while tracking the fixed-wing UAV), and other parameters such as desired relative recovery velocity.



**Figure 8.3:** Simplified overall overview of the system. The operator specifies the location of the *Virtual Runway*, based on weather conditions, air space regulations and the surroundings. This instructs the multirotors to wait at the start of the virtual runway, while monitoring the position of the incoming fixed-wing UAV to initiate the recovery maneuver. Low-level controllers synchronize the relative formation of the multirotors, to position the net at the correct position and orientation.

2. **Standby:** The multirotors take off, and lift the net toward the start of the virtual runway. They are now waiting for the incoming fixed-wing UAV to approach.
3. **Approach:** When the fixed-wing UAV is approaching, the multirotors starts to align themselves with the location of the fixed-wing UAV in the cross-track plane of the virtual runway. This is typically 10 s prior to impact, and the tracking is in a *low-gain mode* to not over-react unnecessarily to the movements of the fixed-wing UAV as it is approaching.
4. **Start:** Based on the speed on the incoming fixed-wing UAV and desired relative speed, the multirotors starts to move along the virtual runway with a prescribed velocity and acceleration profile as to intercept the fixed-wing UAV at the desired recovery point. The multirotors are able to accelerate quite fast, so the desired speed is achieved quickly. At this point, the multirotors track the location of the fixed-wing UAV with a *higher gain* and agility, typically 3-5 s before the recovery time.
5. **Catch:** The multirotors are continuously monitoring the state of the incoming fixed-wing UAV, and tracks it's location until a successful catch is detected. If at any point the tracking is off or communication is lost, the multirotors can initiate a evasive maneuver and restart the recovery phase by instructing the fixed-wing to loop back and do another attempt.
6. **Return:** After the catch, the fixed-wing UAV stays attached to the net due to hooks and other edged structures, and is safely transported back by the multirotors which land individually after releasing the net.

### 8.2.2 Navigation

Precise navigation is crucial for precision landing of UAVs. In this work, we utilize *Real-Time Kinematic (RTK) Global Navigation Satellite System (GNSS)*. This is a navigation technique using the carrier wave of the incoming signals from the satellites, and comparing the signals to that received by a base station. By computing the phase shift between the signals at the UAV (rover) and the base, the location can be locked in at centimeter-level accuracy. Such a system was used in [122] for landing a fixed-wing UAV in a stationary net, which also contains more detailed information about RTK GNSS systems.

### 8.2.3 Emergency Plans

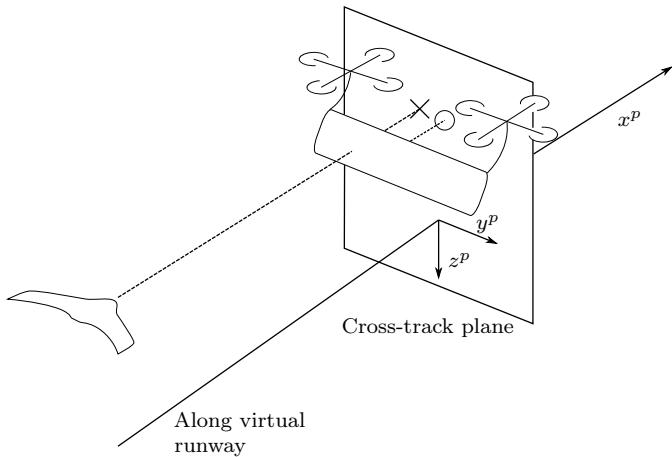
The design of the recovery maneuver readily makes it possible to design various emergency plans to be executed at different phases of the maneuver, as discussed in Section 8.2.1. The maneuver can be halted at any point without risk of damage, even near the point of recovery. By continuously monitor the available data, emergency plans can be executed in the case of communication errors or loss of navigation solutions. Such plans can include repositions of the multirotors and waiting-patterns for the fixed-wing. In severe error cases, for instance if the multirotor UAVs are unable to maintain their desired formation positions due to mechanical faults or complete loss of satellite reception coverage, the suspended net can be released from the suspension points.

### 8.2.4 Virtual runway

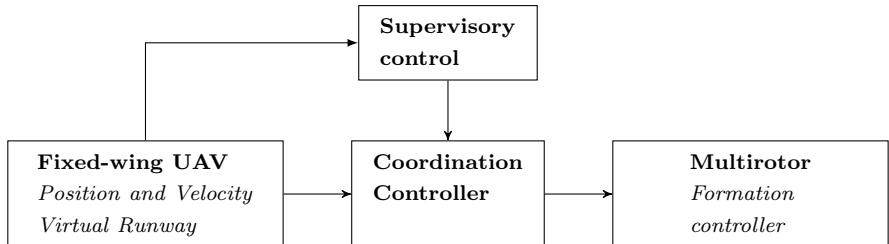
Figure 8.4 illustrates the *virtual runway* (VR). The virtual runway is used by the operator to set a designated area for the recovery maneuver. In addition, the operator can specify boundaries in the vertical and lateral direction of the runway, working as a virtual fence for the location of the multirotor UAVs. The virtual runway defines a path-frame, which is again divided into a *cross-track* plane and an *along-track* distance, so we can design controllers for each part separately.

## 8.3 Control Design

This section introduces the control design, and gives details about each of the different parts. The overall structure can be seen in Figure 8.5. There are three distinct control modules, namely a *Coordination controller*, which uses information from the incoming fixed-wing UAV to intercept and recover it, by sending setpoints to the underlying *multirotor formation controller*. A *supervisory controller* tracks the current overall state of the maneuver. All three are further discussed in the next sections. The *Virtual Runway* formally defines a path coordinate frame  $\{p\}$  at constant altitude, which is defined by an origin  $\mathbf{p}_{p/n}^n$  and a rotation  $\psi$  around the  $\{n\}$  z-axis such that  $\mathbf{R}_p^n = \mathbf{R}_z(\psi)$ . Then a position  $\mathbf{p}^n$  can be decomposed in  $\{p\}$  by the transformation  $\mathbf{p}^p = (\mathbf{R}_p^n)^\top (\mathbf{p}^n - \mathbf{p}_{p/n}^n)$ . Further, the frame is divided into a *cross-track* plane and an *along-track* distance, for which coordination controllers are developed individually.



**Figure 8.4:** Illustration of the *virtual runway*. The runway defines a *path coordinate frame*  $\{p\}$ , and can be divided into a cross-track plane ( $y^p z^p$ ) and an along-track distance  $x^p$ . The center position of the net on the cross-track plane is marked with a circle, while the intersection of the cross-track plane and the path of the fixed-wing UAV is marked with a cross.



**Figure 8.5:** Information flow in the controller structure. Based on the current position of the fixed-wing UAV, the supervisor starts the net-recovery maneuver. The coordination controller guides the two multicopters along a virtual runway to intercept the fixed-wing UAV.

### 8.3.1 The Fixed-Wing UAV Autopilot

The fixed-wing is equipped with an autopilot, taking care of all local navigation and control tasks with the accuracy of a standard code-based GNSS solution. For the recovery maneuver, the fixed-wing UAV receives information about the virtual runway from the operator, specifying altitude and reference air speed. Information about position and velocity is continuously transmitted to both the operator as telemetry data and to the multicopter autopilots. The detailed aspects of guidance and control of a fixed-wing UAV is thoroughly addressed in the literature (for instance [12]) and is not within the scope of this paper.

### 8.3.2 Multirotor Control

Modeling and control of a multirotor UAV has been thoroughly studied in the literature. For a detailed survey, refer to [80] and references therein. In this context, we are interested in the translational dynamics, and assume that an autopilot with a low-level controller is present on the UAV.

In the following, let the subscript  $(\cdot)_i$  denote the  $i$ 'th multirotor. It's dynamics is governed by

$$m_i \dot{\mathbf{v}}_i = m_i \mathbf{g} + \mathbf{R}_i f_i + \boldsymbol{\tau}_{L,i} \quad (8.1)$$

where  $m_i$  is the vehicle mass,  $\mathbf{v}_i \in \mathbb{R}^3$  the velocity in a local North-East-Down (NED) frame  $\{n\}$ , assumed inertial,  $\mathbf{g}$  is the gravitation acceleration vector  $[0 \ 0 \ g]^\top$ , where  $g$  is the gravitational constant. The orientation or the multirotor, specified by the body-aligned reference frame  $\{b\}$ , is represented with  $\mathbf{R}_i \in \mathcal{SO}^3$ . This orientation is crucial to the behavior of the multirotor, as it is can only apply thrust,  $f_i \in \mathbb{R}$ , along the  $z$ -axis. External forces, such as the load from the suspended net and environmental effects, is given in  $\boldsymbol{\tau}_{L,i} \in \mathbb{R}^3$ .

The direction of applied force is given by  $\mathbf{R}_i$ , which corresponds to specifying the roll- and pitch-angle of the vehicle. Since we assume a sufficiently fast low-level attitude controller is present on the multirotor UAV is present (examples of such is given in [80] and [63]), the term  $\mathbf{R}_i f_i$  can be replaced by an inertial control force  $\mathbf{F}_i \in \mathbb{R}^3$ , resulting in the dynamics

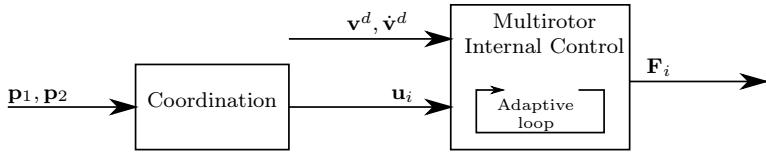
$$m_i \dot{\mathbf{v}} = m_i \mathbf{g} + \mathbf{F}_i + \boldsymbol{\tau}_{L,i} \quad (8.2)$$

#### Formation control

The design of the formation controller between the multirotor UAVs is based on a passivity design as presented in [8] and [10]. Each of the copters are *synchronizing* their relative position, meaning that both multirotors actively maintain their desired configuration by using position and velocity measurement from themselves and the other vehicle. Both vehicles receive a common *mission velocity*, which drives the position of the multirotors in the inertial coordinate frame. In addition, each multirotor has an internal adaptive scheme to compensate for the unknown disturbance  $\boldsymbol{\tau}_{L,i}$  from the suspended net. The structure of the formation controller can be seen in Figure 8.6. More details on the this controller, including Lyapunov-based stability proofs, can be found in [84], [66] and Chapter 7.

The mission velocity  $\mathbf{v}^d$  provided to the formation controller must be a smooth, differentiable signal. Formally, let a signal have  $\mathcal{C}^n$  continuity if the  $n$ th derivative are continuous. To account for the time it takes for the multirotor to realize a desired force output by adjusting its roll- and pitch angle, a  $\mathcal{C}^2$  signal is desirable.

Next, the coordination controller, which supplies the mission velocity to the multirotors, is discussed.



**Figure 8.6:** The overall structure of the relative *synchronization* between the multirotors. By using available positions from each other  $\mathbf{p}_1, \mathbf{p}_2$ , the multirotors cooperatively reach their desired relative position. Internally, each multirotor is applying an *adaptive control* scheme to counter-act the unknown disturbances from the suspended net. An external controller supply the common *mission velocity* signal  $\mathbf{v}^d$ .

### 8.3.3 Fixed-Wing and Multirotor Coordination

To be able to recover the fixed-wing UAV with the net suspended below the multirotors, some level of coordination is needed between the vehicles. Various control strategies were considered, where on one end the fixed-wing UAV should control itself as to hit the net, and on the other the fixed-wing UAV simply followed its nominal flight path and the multirotors did all the tracking. These two strategies could also be combined to a synchronized maneuver.

In the end, we settled on letting the agile multirotors do most of the coordination. This has the advantage of letting the fixed-wing UAV keep a stable, clean flight path rather than correcting minor deviations from the net position. It simplified the control and hardware design, and issues such as time delays in the communication are less pronounced. Thus, from a control-design point of view, the fixed-wing UAV acts as a reference generator, or master, to the coordination controllers on the multirotor UAVs.

In the next sections, let  $\mathbf{p}_i^n \in \mathbb{R}^3, i \in \{1, 2\}$  be the position of multirotor  $i$  in the inertial coordinate frame  $\{n\}$ . Further, we define the position  $\bar{\mathbf{p}}^n := (\mathbf{p}_1^n + \mathbf{p}_2^n)/2 + \mathbf{p}_{offset}^n$  as the centroid of the two multirotors plus an height offset to compensate for the position of the net. Further, the states of the fixed-wing UAV is denoted with subscript  $\cdot_f$ .

#### Coordination - Cross-track

The position of the net is controlled according to the fixed-wing UAV position in the cross-track plane along the virtual runway. A cross-track frame  $\{p_*\}$  is defined as the  $yz$ -plane in the path frame  $\{p\}$ , such that there exist a mapping from a position  $\mathbf{p}^p = [p_x \ p_y \ p_z]^\top$  to  $\mathbf{p}^{p_*} = \mathbf{p}_{2:3}^p = [p_y \ p_z]^\top \in \mathbb{R}^2$ .

To generate smooth  $\mathcal{C}^2$  signals to the low-level multirotor controllers, where a signal has  $\mathcal{C}^n$  continuity if the  $n$ th derivative are continuous, the navigation data from the fixed-wing is fed through a tracking generator. This ensures continuity and smoothness of the reference signals. Let the generator be governed by the dynamics of a third-order filter:

$$\mathbf{x}^{(3)} + (2\zeta + 1)\omega_0 \ddot{\mathbf{x}} + (2\zeta + 1)\omega_0^2 \dot{\mathbf{x}} + \omega_0^3 \mathbf{x} = \omega_0^3 \mathbf{p}_f^{p_*} \quad (8.3)$$

where  $\omega_0, \zeta \in \mathbb{R}$  are tuning parameters. To further enhance the tracking results, the measured velocity from the fixed-wing UAV  $\mathbf{v}_f^p$  is used as a feed forward in the trajectory generation. In addition, to be able to prescribe maximum setpoint velocities and accelerations, (8.3) is re-written as a third-order integrator with saturating elements as illustrated in [67]:

$$\mathbf{x}^{(3)} = \mathbf{u} \quad (8.4)$$

$$\tau_1 = \text{sat}(k_1(\mathbf{p}_f^{p*} - \mathbf{x}), v_{\max}) \quad (8.5)$$

$$\tau_2 = \text{sat}(k_2(\tau_1 + \mathbf{v}_f^{p*} - \mathbf{x}^{(1)}), a_{\max}) \quad (8.6)$$

$$\mathbf{u} = k_3(\tau_2 - \mathbf{x}^{(2)}) \quad (8.7)$$

The parameters  $k_i, i \in \{1, \dots, 3\}$  are found by inspection of (8.3) as

$$k_3 = (2\zeta + 1)\omega_0 \quad k_2 = \frac{(2\zeta + 1)\omega_0^2}{k_3} \quad k_1 = \frac{\omega_0^3}{k_3 k_2}$$

Additionally, the virtual runway defines a boundary that the net should be within. The boundary fulfills a safety requirement in terms of defining the area where the landing will take place, and also gives the operator greater control. For the tracking, when the fixed-wing is outside of the boundary, the tracking generator uses the closest boundary to the virtual runway as the target position. In this case, the feed-forward from the measured fixed-wing velocity  $\mathbf{v}_f$  is disabled. Figure 8.7 sums up the tracking generator.

Smooth  $C^3$  signals are preferable to the low-level formation controller, but to get adequate performance at the final parts of the recovery maneuver it is possible to reduce the model above to a second order one, leading to faster tracking performance. This is achieved by setting  $k_3 = 0$  above and replacing (8.4) with  $\ddot{\mathbf{x}} = \tau_2$ , as illustrated by the dashed box in Figure 8.7.

The desired velocity  $\mathbf{v}_d^{p*}$  for the multirotors are now calculated by

$$\mathbf{v}_d^{p*} = \dot{\mathbf{x}} + \mathbf{K}_p(\mathbf{p}^{p*} - \mathbf{x}) \quad (8.8)$$

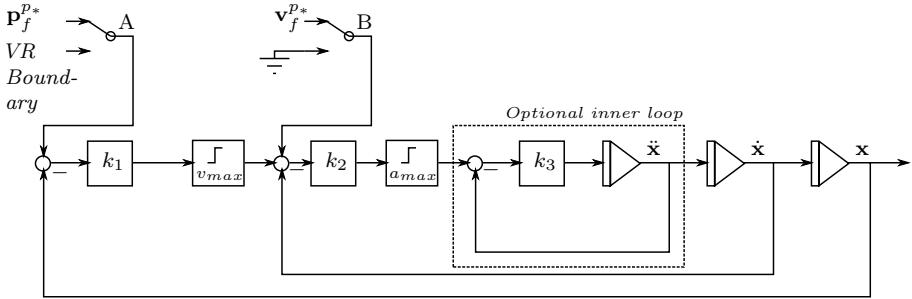
where also  $\ddot{\mathbf{x}}$  is used directly as a feed-forward in the underlying controller.

It should be noted that the net position is not measured explicitly, and furthermore it is not a desirable control target as the net will swing during the transit. Therefore we seek to control the position  $\bar{\mathbf{p}}$  as illustrated in Figure 8.4 as the circle in the cross-track plane. Hence,  $\mathbf{p}^{p*} = \bar{\mathbf{p}}_{2:3}$ .

### Coordination - Along-track

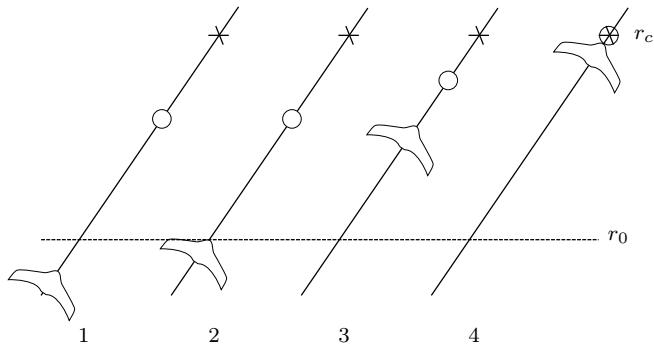
The relative velocity between the net and the fixed-wing UAV is reduced by accelerating the net to a desired velocity. In order to control the point of impact an, open loop scheme is proposed.

For the final recovery phase, the along-track velocity of the fixed-wing UAV is assumed constant. The virtual runway defines a point  $r_c$  along the runway as the



**Figure 8.7:** Block-diagram of the trajectory generator. The measured signals from the incoming fixed-wing UAV is filtered through the tracking algorithm to generate smooth reference signals to the controller. Based on the location of the fixed-wing UAV, the switches  $A$  and  $B$  can be set to adjust for the situation when the fixed-wing UAV is outside of the virtual runway. Note that the saturation blocks are not in between the integrators, ensuring consistent  $\mathcal{C}^3$  signals. However, if faster performance is needed and  $\mathcal{C}^2$  is sufficient at certain stages of the maneuver, the inner loop can be removed.

designated recovery point. While waiting at the start of the virtual runway, the multirotor UAVs should monitor the location of the fixed-wing UAV. Based on a operator-defined relative speed to be achieved by the multirotors at the point of recovery  $r_c$ , the multirotor UAVs will start a pre-defined velocity profile along the virtual runway, to intercept the incoming fixed-wing UAV at  $r_c$ . By knowing the type of velocity profile used, the distance to the fixed-wing UAV,  $r_0$ , can be calculated based on the desired relative speed and along-track velocity of the fixed-wing UAV. This is summarized in Figure 8.8.



**Figure 8.8:** The figure shows the timing of the along-track velocity, where the current position of the net is marked with a circle at different instances of time (1)-(4). When the fixed-wing UAV reaches  $r_0$ , the multirotors starts the velocity profile for forward flight as to recover the fixed-wing UAV at  $r_c$ .

Different methods can be used to create a feasible velocity profile as the ultimate goal is to be able to calculate  $r_0$ . For the experimental setup conducted in this paper, a linear velocity profile with constant acceleration is utilized. As the acceleration

phase of the along-track profile is rather quick for the agile multirotors, this proved more than sufficient. More elaborate profile could obviously be used, for instance the  $C^4$  polynomial profile proposed in [66].

### 8.3.4 Supervisor

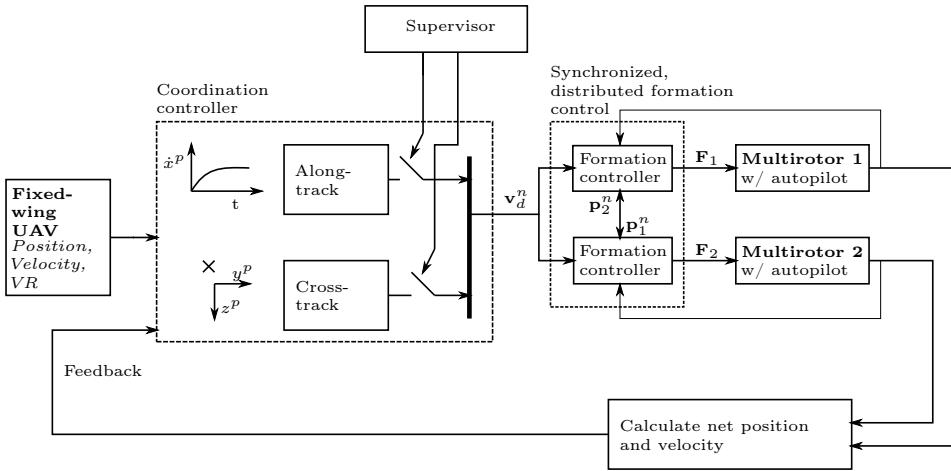
The supervisor monitors the position and velocity of the fixed-wing UAV relative to the virtual runway in order to switch between the different modes in the maneuver, as described in Section 8.2.1. Each mode enables a certain controller and reference which gives a desired velocity setpoint. The overall controller structure is thus summarized in Figure 8.9, where we see how the supervisor controls when to activate the various parts of the coordination controllers.

In addition, the supervisor monitors the maneuver as it is progressing. If, because of wind or other factors, the fixed-wing UAV misses the net, it instructs the vehicles to try the maneuver again. Furthermore, if the projected proximity of the fixed-wing UAV and multirotors are too small, the supervisor can abort the operation. Depending on the situation, an abort can involve the multirotors to climb and reposition for a retry, or releasing the net and abort the mission entirely.

Let  $v_{d,x}^p(t)$  be the along-track velocity profile, designed as discussed in Section 8.3.3. By combining this with the desired velocity from the cross-track control, we get  $\mathbf{v}_d^p = [v_{d,x}^p(t) \quad (\mathbf{v}_d^{p*})^\top]^\top$ , and the resulting desired velocity in  $\{n\}$  can be found by the following transformation

$$\mathbf{v}_d^n = (\mathbf{R}_n^p)^\top \mathbf{v}_d^p \quad (8.9)$$

which gives the desired mission velocity for the two multirotors as discussed in Section 8.3.2. The complete controller structure is illustrated in Figure 8.9.



**Figure 8.9:** Illustration of the control-structure. The cross-track controller gets feedback from the position of the fixed-wing UAV in a plane orthogonal to the virtual runway. The along-track controller is an open-loop controller, initiated by the supervisor when the fixed-wing UAV reaches the virtual runway.

## 8.4 Experimental Validation

To validate the concept illustrated above, a series of experiments were conducted with a platform of relatively small size vehicles. The fixed-wing UAV used for experiments are a modified hobby-grade RC plane, with a takeoff-weight of approx. 1.5 kg, and a cruise-speed of around 15 m/s. The multirotors weigh approx. 2.2 kg, and have enough lift capacity to lift the fixed-wing UAV, the net and necessary payloads. The following sections will discuss the various practical aspects with the implementation, such as the overall system architecture, choice of net and structural support systems, and the sensors needed. Finally, we will go over the operational aspect and the safety mechanisms built into the system.

Note that even though the UAVs in question can be considered relatively small, it is important to realize that all the functionality and solutions discussed in this section can be transferred to larger vehicles. The simulation study conducted in [66] used a fixed-wing UAV with a maximum take-off weight of 4.2 kg, and multirotors with a total of 11 kg lifting capacity.

### 8.4.1 Overall System Architecture

The system consist of three main components; the fixed-wing UAV, the multirotors carrying the net, and a ground control segment. The ground control segment is used by the operators to configure the mission, and for monitoring and control during execution as discussed in Section 8.4.4. The fixed-wing UAV continuously sends position and velocity data to the multirotors, which calculates the path for interception when initiated by the operator. Even though the multirotors synchronize their position to achieve relative formation between themselves, the tracking- and coordination controllers described above runs on one of the multirotors, acting as a master.

Further, the system is designed to have a minimal impact on the normal operation of the fixed-wing UAV. Only a small module, consisting of an RTK GPS antenna, RTK GPS receiver and a wireless transmitter is required to be placed in the UAV. These components enable the accurate position and velocity of the fixed-wing UAV to be available on the controllers running on the multirotors.

### 8.4.2 Software and Hardware Platform

Each of the UAVs (both the fixed-wing and multirotors) are equipped with a Pixhawk [107] autopilot running the Ardupilot [9] software stack. The autopilot handles navigation and low-level control of the vehicles, and for the fixed-wing UAV it also handles the guidance tasks. These autopilot's are based on highly flexible open-source software, which makes them ideal for the trials described in this paper. The software on the pixhawks in the multirotors have been slightly modified with regards to controller inputs. The modifications are available online.<sup>1</sup>

The onboard computer is a Beaglebone Black [11] running the LSTS software toolchain [106]. The Beaglebone is a versatile embedded linux development board,

---

<sup>1</sup><https://github.com/krisklau/ardupilot/tree/copterdev/3.3>

offering plenty of IO<sup>2</sup>, accompanied by a 1 GHz ARM Cortex-A8 CPU. The LSTS toolchain is the core software suite used for implementation and realization of the described recovery maneuver. The toolchain consists of a minimal Linux distribution (Glued), a software framework layer (DUNE), and a ground control segment (Neptus). It is all open-source, and available online [43]. The toolchain is highly expandable, and makes it relatively easy to add new controllers to the underlying framework. It is designed to handle distributed heterogeneous vehicles operating in the same environment, which when accompanied with a dedicated GCS makes it a better alternative for complex operations than other available frameworks.<sup>3</sup> All the controllers discussed in this paper are implemented in C++ in DUNE.

The communication to the Pixhawk from the Beaglebone is done using the *MAVLink* protocol [82] over a serial interface (UART) running at 912000 kb/s baud rate. Attitude and other telemetry data is transmitted from the pixhawk at 25 Hz, with a measured maximum latency of 30 ms. As depicted in Figure 8.9, DUNE transmits desired acceleration (labeled as  $\mathbf{F}_i$ , see also (8.2)) in the inertial coordinate frame to the pixhawk, which calculates the resulting desired roll- and pitch-angle and total thrust.

As discussed, the accuracy of RTK GPS is required for the recovery maneuver. We leverage the open-source library RTKlib [129], which enables us to build a low-cost cm-level accurate positioning system. Accompanied by the u-blox M8T raw GNSS receiver and a slightly modified version of RTKlib<sup>4</sup>, we are able to measure position and velocity of all the UAVs at 10 Hz. RTKlib runs alongside DUNE on Beaglebone, and the RTKlib configurations are also available online.<sup>5</sup>

The choice of antennas are critical for successful operations with RTK GPS. Even though patch-antennas with a suitable ground plane are most common for UAVs, we have opted for a helical active L1 antenna from Maxtena which has a very low weight and does not require a ground plane. In our tests they have shown superior performance on the tilting UAV platforms.

Although our controllers receive precise position data from the RTK GPS at 10Hz, due to the calculation times of RTKlib and propagation delay from the GPS receiver the data is available with a total delay of approx. 200 ms. This puts a restriction on the maximum achievable bandwidth of the controllers. By fusing the data with inertial- or other sensors, this delay can be compensated for, but this is not done in this work. However, the achievable bandwidth proved sufficient for the proposed control architecture and the experiments conducted for this thesis.

Communication between all vehicles (fixed-wing UAV, multirotors) and the ground control station is through a wireless network solution developed by Ubiquity Networks<sup>6</sup>. Specifically, the *Ubiquity Rocket M5* provides a 5.8 GHz wireless network

---

<sup>2</sup>Inputs and Outputs, such as serial lines (UART), SPI and General Purpose Input Output (GPIO).

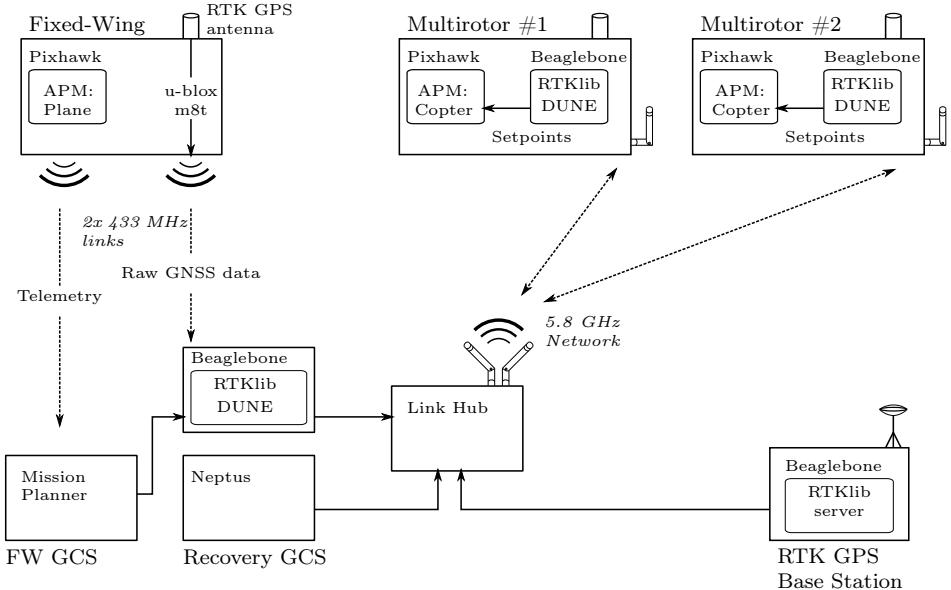
<sup>3</sup>For instance the also excellent *Robot Operating System*(ROS), which has a broad user base in the scientific community.

<sup>4</sup><https://github.com/krisklau/RTKLIB/tree/fix/iss99>

<sup>5</sup><https://github.com/LSTS/glued/blob/master/systems/ntnu-b2xx/fs/etc/rtklib/conf/rtkrcv.conf>

<sup>6</sup><http://ubnt.com>

with the AIRMAX transmission protocol. This is a *Time Division Multiple Access* (TDMA) network, providing constant throughput with minimal latency variance, which is important for the coordinated control strategies proposed in this chapter. The radio is interfaced through a standard RJ45 ether net plug, providing a bridged interface to the network.



**Figure 8.10:** Detailed communication diagram for the experimental setup. In addition to the on-board autopilots, the setup consists of four Beaglebone Black's for onboard and ground processing, ground control stations and a wireless network from Ubiquity Networks.

### 8.4.3 Net, Support Structures and Sensors

The net under the multirotors is attached to a flexible aluminum rod both at the bottom and top of the assembly, as seen in Figure 8.1. The rod helps to stretch out the net, and also works as a shock-absorber during the impact with the fixed-wing UAV. Lastly, it will make sure that most of the forces from lifting the fixed-wing UAV after recovery is directed downwards, rather than inwards, which would cause the multirotors to spend unnecessary energy to stay apart from each other.

To choose the net for the setup, various alternatives were explored. It is important to keep the weight low, and it additionally needs to be strong enough to last for several recovery maneuvers. In the end, we chose a very light-weight ( $<40$  g) net made from nylon threads, each at the thickness of a regular fishing-line, originally intended to keep birds away from berry-bushes. The net is 5 by 3 m.

After the fixed-wing UAV has made contact with the net, some mechanism is needed for it to stay attached. As can be seen in Figure 8.11, we rigidly attached two metallic rods to the nose of the UAV, and bent them once to act as a simple

hook. This proved to be very effective, with little intrusion on the fuselage and flight performance. Similar hooks could also be applied to the wings of the UAV, to enable even more robust linkage to the net.

To attach the net to the multirotors, a multi-purpose attachment device was created, see Figure 8.11. It consists of (1) a quick-release mechanism, and (2) a tension sensor. The quick-release mechanism is built around a cylinder with a circular release pin. This design, combined with a smooth nylon-coated wire attached to the top of the net, allows for a low-friction operation which can be driven by a micro-sized (4.7 g) servomotor, even with several kg's of load. The tension-sensor is a low-weight load-cell from Futek, LSB200 [40]. The combined assembly is installed on a gimbal-like structure to facilitate free range of motion, so that the load cell is only subjected to external forces along it's sensor line as illustrated in Figure 8.11.



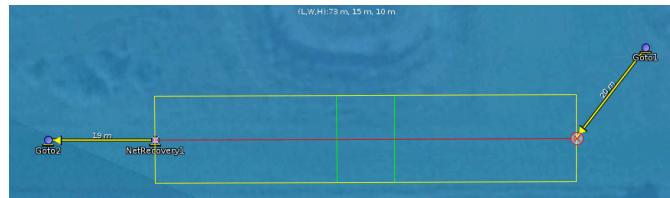
**Figure 8.11:** Left: Metallic hooks attached to the nose of the fixed-wing UAV, used to latch onto the suspended net. The hooks had little to no effect on flight performance. Right: Assembled quick-release mechanism and tension sensor on a gimbal-like structure for free range of motion.

#### 8.4.4 Ground Control Station

To control the recovery maneuver, the operator is presented with an overview of the location of the recovery maneuver, as seen in Figure 8.12. Here the operator can set the height and width of the virtual runway (works as a fence for the location of the multirotors) prior to the maneuver, as well as monitor the progress with the location of all vehicles while the recovery is taking place. In addition to the sky-view, the operator can also see the location of the vehicles in the cross-track plane, which is very useful for the initial tuning of the maneuver parameters. These views are extensions to the GCS in the LSTS toolchain as discussed in Section 8.4.2

#### 8.4.5 Controller Tuning

To tune the controllers, several fixed-wing recovery maneuvers were conducted without the actual net between the multirotors, and an additional 5 m of offset distance was put in place. This allowed safe operations while tuning and assessing the performance of the system. The whole system has three tuning profiles: *Cruise*, *Normal*, and *High Gain*, which is in use during transport, approach and final



**Figure 8.12:** Zoomed-in view of the operator interface for the net-recovery maneuver. The yellow rectangle represents the safety area (width of the virtual runway), while the green lines represent the location of the planned recovery.

recovery phase, respectively. Additionally, as described in Section 8.3.3, a faster 2nd order trajectory generator is in use in the *High Gain* phase.

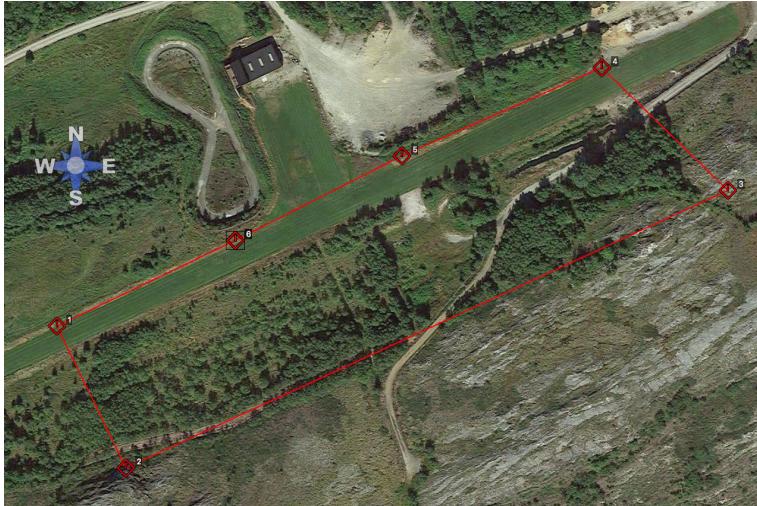
Overall, the formation controller was to a higher bandwidth than the cross-track coordination controller, and a high relative damping factor was used to ensure smooth operations and to better react to the abrupt force from the incoming fixed-wing UAV.

#### 8.4.6 Operational Aspects

As this is a rather complex experiment, involving three flying vehicles, it is vital to have a well-grounded operational environment. Three UAV pilots were involved in the operation, each responsible for a separate vehicle and able to take control. The operation was conducted as follows:

Pilot A conducted a manual takeoff of the fixed-wing UAV using an RC transmitter. After reaching cruising altitude, a rectangular flight-plan (see Figure 8.13) which included the recovery location was uploaded and initiated on the fixed-wing autopilot. On-board battery fail-safe systems monitor the fixed-wing UAV battery capacity. The two multirotor UAVs were positioned on the ground, with the net between them. A pilot-assisted control mode is then initiated, which allows a single pilot, Pilot B, to take-off and maneuver the two multirotor UAVs in a synchronous fashion. In reference to Figure 8.6, the pilot sets the desired *mission velocity* using an RC-remote controller. After takeoff, Pilot B issues a command for the autonomous recovery maneuver to start, in which the multirotor UAVs move towards the start of the virtual runway, awaiting the approaching fixed-wing UAV. At all times, Pilot B and Pilot C are able to take manual control of multirotors 1 and 2, respectively. The procedures for this, along with a number of other safety systems, are discussed in the next section.

After the autonomous recovery is completed, the multirotors return with the fixed-wing UAV to the start of the virtual runway. There, they await for the pilot to give command to enter the pilot-assisted control mode again. Then, the single pilot guides the multirotor toward the landing area. When the now suspended fixed-wing UAV is lowered to the ground, the pilot releases the net with a switch on his RC transmitter. Pilot B and Pilot C then lands the multirotors individually.



**Figure 8.13:** Nominal flight-plan for the fixed-wing UAV during the experiments.

The designated recovery location is in the middle of the grass runway, on the upper long leg of the path.

## Safety

In accordance with the local civil aviation authorities, these tests were categorized as conducted within *Visual Line Of Sight* (VLOS), and were coordinated with the nearby air traffic control tower. At all times, a pilot has the capability to resort to manual control, using an RC transmitter to the autopilot, of each aircraft should it be needed.

A number of automatic failsafe systems were put in place. To simplify the operational aspects during these trials, any of the following conditions triggered an *abort*. Should such an abort be triggered, the two multirotors would (1) simultaneously drop the suspended net, and (2) move 3 m directly away from each other, using control-logic on the low-level autopilot. There, the multirotors would wait until manual control is issued by the pilot. The following conditions triggers an *abort*:

- *Pilot abort*: Each pilot has a RC transmitter with a dedicated button to manually trigger an abort, should the need arise due to unforeseen events.
- *Battery capacity*: Battery voltage and remaining capacity is monitored by the autopilot. If the level reaches a certain lower bound, a battery-failsafe is triggered.
- *Weight of suspended load*: During recovery, should the weight on a single multirotor from the suspended load be measured to be over 1.5 kg for more than 0.5 s, a weight-failsafe is triggered.
- *Communication timeout*: Should the communication link between any of the vehicles or ground control station drop out for more than 0.5 s, the position and velocity information between the vehicles can no longer be updated, and a timeout-failsafe is triggered.

- *RTK GPS lock loss*: The system relies on the accurate positioning solution of the RTK GPS for precision recovery of the incoming fixed-wing UAV, and maintaining relative formation between the multirotors. Should the navigation solution lose lock, either due to communication failure with the base station, satellite obstruction, numerical calculation errors or other events, an abort is triggered. Note that the system permits a short (1 s) loss of lock, in which a navigation algorithm augments the location solution from a standard secondary code-based GPS with the last valid RTK GPS solution before issuing the abort.

The ground control station would inform the operator of the source of the abort signal. Although the failsafe-procedures uses straight-forward condition-based logic that all led to the same abort event, it provided a robust platform for experimental tests. Naturally, more logic could be added in the future. As an example, two levels of battery failsafes levels could be added, where only the lower critical one would trigger a drop of the suspended load, useful to prevent unwanted release of the fixed-wing UAV just after the recovery, but before being lowered to the ground.

In the experimental trials conducted, the length of the timeouts discussed above was chosen rather conservatively. In a operational scenario, longer periods of communication dropout and loss of RTK GPS lock can be tolerable, depending on the specific situation. For instance, in the case of communication loss between the fixed-wing UAV and the multirotor UAVs, a wait-and-hold pattern can be initiated until communication is re-established. Secondary short-range links can be used for redundancy and transmission of critical data between the multirotors to ensure reliability, and deeper integration with inertial sensors can lead to better precision during loss of RTK GPS lock using a standard code-based GPS as a fallback solution.

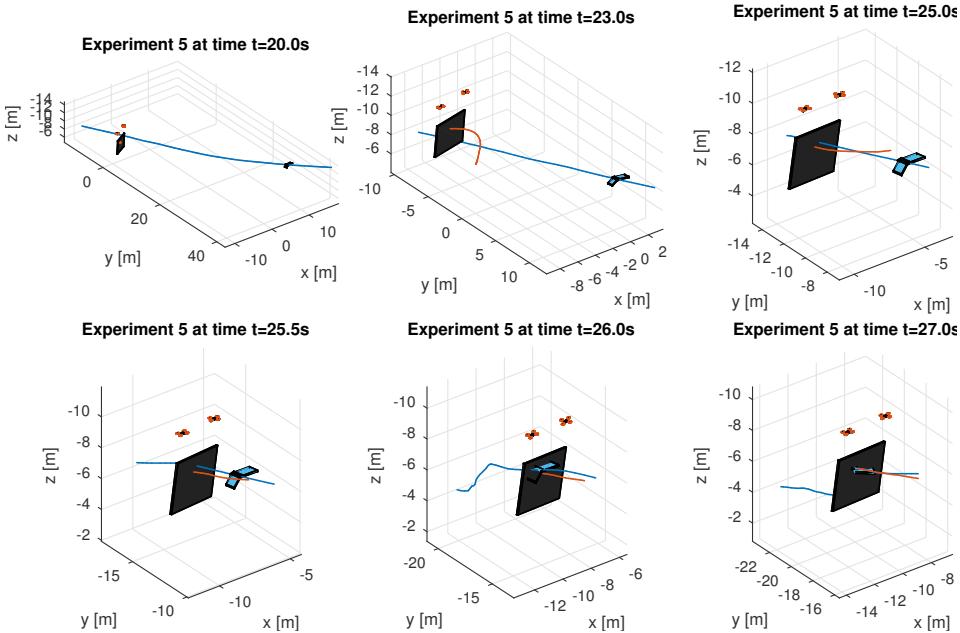
Further, the three pilots used was needed to ensure safe trials during the test-phase. As the system is designed to be fully autonomous, the number of pilots can readily be reduced. A single pilot can be used, assisted by the on-board controller software, to guide the multirotor UAVs during takeoff and landing. As the fixed-wing UAV is on respectively on autopilot or recovered in the net, the same operator crew used in the fixed-wing operation can be used for the recovery maneuver.

## 8.5 Results

Experiments were conducted to test the proposed maneuver in December of 2016 in Trondheim, Norway. The temperature was a brisk 2 °C, with a fairly calm wind of approx. 3-4 m/s, and some stronger gusts. In preparations for the actual recovery maneuver, several trial runs were conducted using all assembled equipment except for the net. That way, we could test the setup and tuning of the controllers prior to performing the recovery maneuver as described in Section 8.4.6. Snapshots of one of the experiments can be seen in Figure 8.14.

The fixed-wing UAV was set to continuously fly a rectangular pattern with an altitude of 40 m to stay clear of the surrounding trees and hills. One of the long legs corresponded to the desired landing area, in which the UAV descended to an altitude of approx. 10 m. The cruise-speed of the UAV was 16 m/s, while at

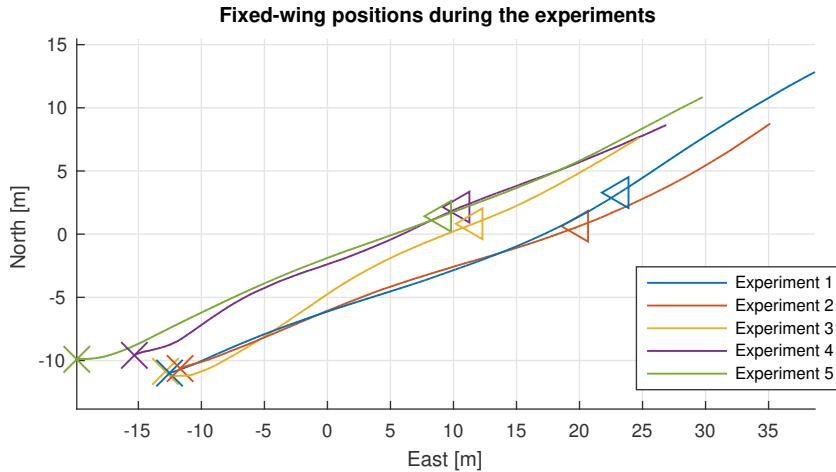
the landing zone it had a designated speed of 12 m/s. The resulting flight-path of the fixed-wing UAV has some natural variation due to the use of single-precision code-based GPS for navigation, which can be observed in Figure 8.15.



**Figure 8.14:** Snapshots of the recovery maneuver from Experiment 5, which shows the movements of the vehicles before and during the recovery. The illustrations are renderings from the data logged during the experiment. The blue line represents the trajectory of the fixed-wing UAV as recorded by the on-board RTK GPS, while the red line is the trajectory of the calculated net-position up until the snapshot time.

A total of five attempts at autonomous recovery of the fixed-wing UAV using two coordinating multirotors were conducted. Of the five tests conducted, four resulted in a successful recovery of the fixed-wing UAV. For the remaining experiment, excessive lateral movements and a structural weak point in the net caused only the wing to make contact, thus resulting in the fixed-wing UAV not attaching to the net. However, it continued its nominal flightpath unharmed. The lateral movements of the fixed-wing UAV took place the moment before recovery, most likely caused by a strong wind gust. This can be seen at the very end of Figure 8.17. By applying hooks to the wings as well, this attempt would also have been successful. Also, a more aggressively tuned or larger fixed-wing UAV would be less affected by the such wind gusts, should they appear just before the recovery. The multirotor UAVs were successful in tracking the variations in the fixed-wing flight path leading up the unsuccessful recovery, as well as on the other successful ones.

A video illustrating the successful recovery of the fixed-wing UAV is available



**Figure 8.15:** Flight-path of the fixed-wing UAV just prior to the recovery in the net. The figure illustrates the accuracy and behavior of the path-controller in the low-level fixed-wing UAV autopilot. Note that the fixed-wing UAV does not alter its path based on the location of the net; it is up to the multirotors to use position and velocity data to intercept and recover the incoming fixed-wing UAV. The cross and triangle marks the time of recovery and start of multirotor *high gain* phase, respectively.

online.<sup>7</sup>

Figure 8.14 shows an overview of one of the experiments conducted. We can see the two multirotors waiting with the suspended net for the fixed-wing UAV to approach. When it is close, the multirotors starts the crosstrack tracking as to intercept the path of the incoming fixed-wing UAV. We see that in both cases the multirotors are able to intercept and successfully recover it.

To show the performance of the control algorithms, the results from all five experiments are assembled in Figure 8.16. Here, the *relative position* of the net and fixed-wing UAV is shown at the top, while the bottom part shows the distance from the fixed-wing UAV to the *net edge* in the cross-track plane.

Figure 8.17 shows how the net position is tracking the position of the fixed-wing UAV in the y-component of the crosstrack plane in the virtual runway. The marked triangle represents the point where we enter a *high gain* profile, where 3rd order trajectory generator is replaced with a faster 2nd order one.

In Figure 8.18, the cable tensions for two of the experiments are shown. The tension was measured using on-board strain gauges, as described in Section 8.4.3. It can be seen that at the moment the fixed-wing UAV makes contact with the suspended net, a peak tension of approximately 2.5 kg is measured. However, it relatively quickly settles to the resting mass of the now suspended fixed-wing UAV. At the end of the time-series, it is seen how the weight drops to zero when the plane is put to rest on the ground, and the net is released.

<sup>7</sup><https://youtu.be/aLNwuTjmW4>

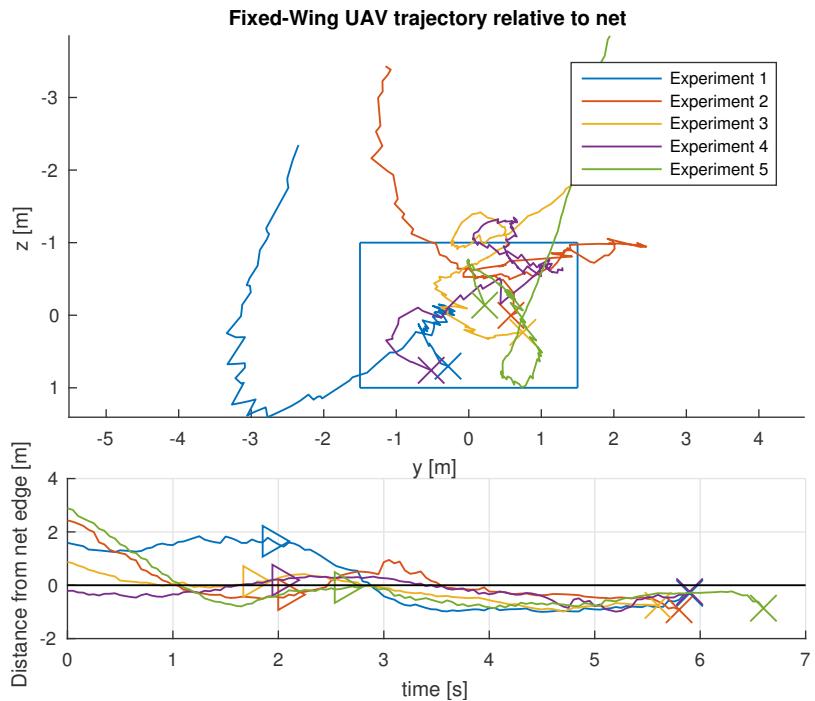
## 8.6 Chapter Summary and Conclusions

In this chapter, we have presented a novel method for autonomous recovery of fixed-wing UAVs using a net suspended below two multirotor UAVs. The proposed method enables fixed-wing UAV operations in the absence of runways or sufficient space for a stationary net. It is especially suited for ship-based marine launch and recovery operations, as the wave-induced heave motion of the ship, and other environmental factors does not affect the recovery, as it is conducted off the ship deck. In addition, the recovery maneuver can be readily adapted to changing wind conditions and direction.

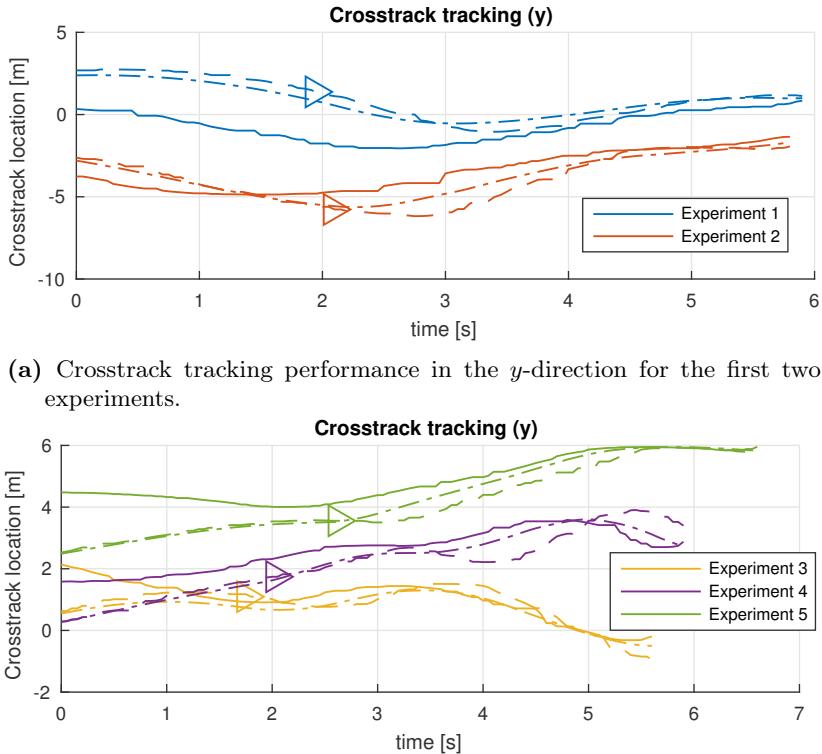
A complete control system was proposed, along with the definition of a *virtual runway* for which the recovery maneuver is performed along. The proposed control system includes cooperative formation control algorithms of the two multirotors, and a trajectory generator to track and intercept the incoming fixed-wing UAV.

Multiple experimental trials were conducted on a smaller-scale fixed-wing UAV to validate the proposed control system. The recovery maneuver was conducted fully autonomously, for which the control algorithms were implemented on-board each UAV in a distributed fashion. The experimental platform included the use of Real-time kinematic GPS for precise navigation solutions.

Although the experimental platform used consisted of a small scale fixed-wing UAV ( $\sim 1.5$  kg), the proposed control system is directly applicable to larger systems.

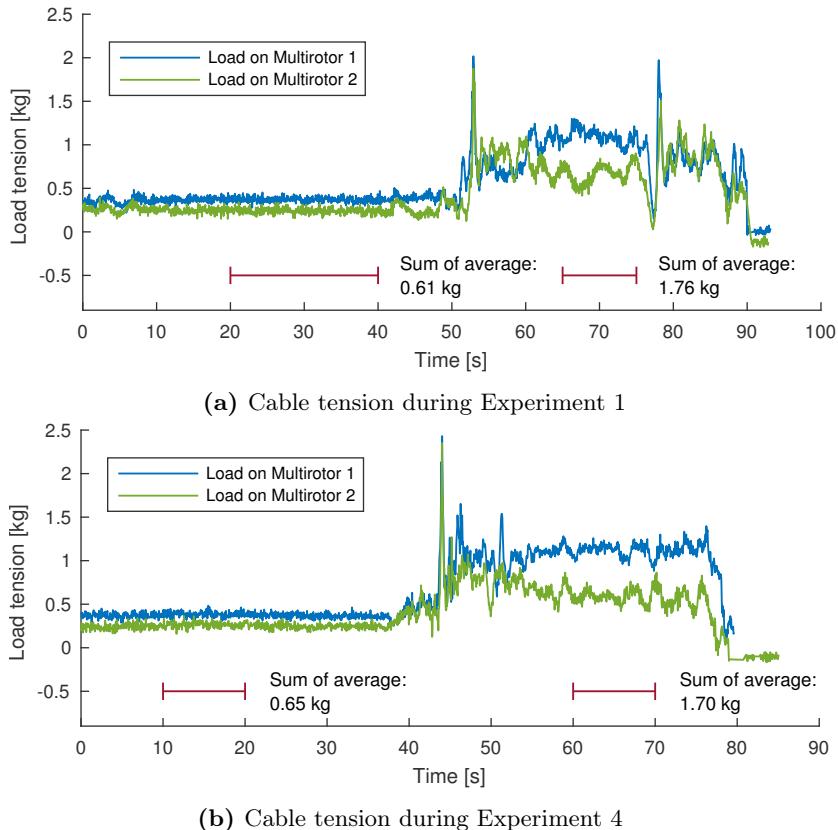


**Figure 8.16:** Tracking results from all five recovery attempts. Even though experiment three was unsuccessful in firmly attaching the fixed-wing UAV to the net, the data still shows adequate tracking performance. Note that the upper part illustrates the *relative* position of the fixed-wing UAV and the net (blue square) in the crosstrack plane.



**(b)** Crosstrack tracking performance in the  $y$ -direction for the last three experiments.

**Figure 8.17:** Crosstrack tracking performance for (a) the first two experiments, and (b) the last three experiments. Relative to experiments 1& 2, the damping parameter in the tracking generator was adjusted slightly in the last three, resulting in better performance. The solid line is the fixed-wing UAV position, while the dashed and dash-dotted line represents the net location and trajectory generator output, respectively. The triangle position mark the moment in time when the *high gain* control profile is activated.



**Figure 8.18:** Cable tension on both multirotors for experiments 1 (a) and 4 (b). In both cases, the initial wait time shows a constant weight of roughly 0.6 kg, which represents the total weight of the net + structure rod assembly. A peak force of about 2.5 kg on each multirotor is measured just as the fixed-wing UAV enters the net. After a brief settling time, the total weight settles to approximately 1.7 kg, which matches well to the weight of the fixed-wing UAV. The average weight-calculations is done on the data as marked in the figure.



# Chapter 9

## Conclusions

This chapter presents the main conclusions from the work presented in this monograph. It also provides some suggestions and remarks about future work.

### Conclusions

This monograph gave an overview of the challenges introduced by transporting a suspended payload with a single or multiple multirotor UAVs, and presented several control system designs to solve the accompanied control problems. For each case, the system dynamics was modeled, and the suggested control design verified through simulations. Further, experimental data from a number of different flight scenarios was used to justify the validity of the various proposed controllers.

In addition, a novel recovery method for fixed-wing UAVs was presented, which were tested experimentally on a small-scale fixed-wing UAV platform. The rest of this section concludes each chapter individually.

Chapter 2 gave an overview of the related hardware and software used in the work for which this monograph is based. It intended to give the reader an insight into the level of complexity one faces when designing systems for coordinated control applications, and hopefully serves as basis for future research into similar areas.

Chapter 3 provided an extensive model of the nonlinear dynamics that occurs when connecting a multirotor to a suspended load. We considered the case when the suspended load is attached to the centre of gravity of the multirotor, and designed a robust, nonlinear tracking controller based on the backstepping technique. The controller ensured asymptotic stability of the origin of the tracking error, even in the presence of unknown disturbances from wind. To suppress unwanted oscillations and swings in the suspended payload, a combination of an open- and closed-loop approach to trajectory generation was suggested. Furthermore, both numerical simulations and experimental trials verified the suggested controller.

Chapter 4 introduced multibody operations, and provided an insight into the complex dynamics of multibody systems. Different modeling techniques were discussed, and the Udwadia-Kalaba equation was introduced as an efficient alternative to more established methods. Equations of motion was derived using these equations, which was used for simulation purposes for the remainder of the thesis.

## 9. Conclusions

---

To counteract the disturbance from the suspended load, one could utilize a sensor to measure the tension of the suspension wire, which subsequently can be directly cancelled in a feed-forward fashion by the controller. This is utilized in Chapter 5, which also considers the case when external positioning systems are not of sufficient precision for close formation flight. Instead, it proposes measuring the relative angles of the suspension wires, and designs a kinematic controller to achieve the desired formation. Along with Lyapunov stability analysis, numerical simulations verify the results.

Chapter 6 formulated the control issue as a consensus problem, and designs a cooperative path-following control structure for each multirotor, which ensures convergence to the desired formation along a pre-defined parameterized path. This controller also relies on measurements of the tension from the suspended load to cancel the effects on the multirotors, and is verified by numerical simulations.

Chapter 7 presented a controller in which the suspended load is considered having an unknown, albeit slowly varying, effect on the multirotors. Through a passivity-based framework, a position-based formation controller with bias estimation is presented. In addition to numerical simulation, the controller is verified experimentally, using three multirotor UAVs cooperatively carrying a payload.

As the controller presented in Chapter 7 is based on synchronizing the relative position of each UAV, it does not require the generation of a parameterized path as the controller presented in Chapter 6, which is a non-trivial task. Further, it includes an adaptive term to compensate for the unknown tension from the suspended load, instead of relying on a force measurement.

Chapter 8 presented a novel method for autonomous recovery of fixed-wing UAVs using a net suspended below two multirotor UAVs. The proposed method enables fixed-wing UAV operations in the absence of runways or sufficient space for a stationary net, and a patent for the concept was filed on April 12, 2017. It is especially suited for ship-based marine launch and recovery operations, as the wave-induced heave motion of the ship, and other environmental factors does not affect the recovery, as it is conducted off the ship deck. In addition, the recovery maneuver can be readily adapted to changing wind conditions and direction.

A complete control system was proposed, along with the definition of a *virtual runway* for which the recovery maneuver is performed along. The proposed control system includes cooperative formation control algorithms of the two multirotors, and a trajectory generator to track and intercept the incoming fixed-wing UAV.

Multiple experimental trials were conducted on a smaller-scale fixed-wing UAV to validate the proposed control system. The recovery maneuver was conducted fully autonomously, for which the control algorithms were implemented on-board each UAV in a distributed fashion. The experimental platform included the use of Real-time kinematic GPS for precise navigation solutions.

## Future Work

This section gives some suggestions for future work, divided into the relevant topics.

---

## **Robust navigation for time-delayed signals**

Even though the platform used for the experimental verification of the enclosed work used high-precision RTK GPS for positioning, there is still room for improvement. Specifically, navigation filters which takes the delay from the RTK processing software (typically 100-200 ms) into account should be implemented. Further, by fusing with data from other sources, higher sampling rates can be achieved.

To achieve this, the natural choice is fusion of data from inertial sensors. As the low-level autopilot is already equipped with an IMU, the data from this could be used. However, the telemetry data from the autopilot over the serial connection might also be delayed.

Another approach is to use the navigation solution already present in the autopilot for fusion. The real reason RTK is needed, is the high absolute accuracy. However, the position offset from the autopilot solution is very slowly varying, which means delayed data from the autopilot can be used to estimate this offset, and use the data from the autopilot augmented with this offset as navigation data to the controllers.

## **Verification of active load tension compensation**

The two controllers presented in Chapters 5 and 6 relied on actively measuring the tension force from the suspended load to counteract this. Further tests is needed to verify that the low-cost sensors used in this work are accurate enough, and that the transient load response can be properly filtered out.

## **Explicit usage of the interconnected model in the control design**

As discussed in Chapter 4, the multirotors lifting a suspended load are subjected to *dynamic coupling*, which makes it difficult to utilize model-based control to cancel the load disturbance, as it would require feedback from the *instant acceleration* of both the load and the other multirotors to achieve explicitly. In addition, the calculations are somewhat numerically expensive to be carried out in real-time on an embedded processor. However, by exploiting a known configuration when the multirotors have achieved the desired formation, and using an estimate of the body accelerations, there is potential to utilize the Udwadia-Kalaba equations for feedback.

## **Efficient computation of formation specification**

In Chapter 6, an algorithm for designing trajectories that ensured equal tension in the wires to each multirotors by appropriately tilting and rolling the desired relative formation during acceleration was discussed. However, the calculation of the second derivative of these trajectories exactly are very numerically expensive. Approximations using small-angle assumptions, or other representations of the relative formations should be investigated to enable efficient calculations of these trajectories on an embedded platform. In addition, a similar method can be applied to the controller presented in Chapter 7 to ensure smoother operations during acceleration.



## Appendices



## Appendix A

# Experimental Setup

The following is a supplementary image of the experimental setup for the field verifications conducted in Chapter 7.



**Figure A.1:** Setup of the multirotors for the experimental validation of the passivity-based formation control.



## Appendix B

# Stability Analyses

The following details the stability analysis from Chapter 7, Section 7.3.

For each UAV  $i = 1, \dots, N$ , described by (7.11), the storage function

$$S_i(v_i, \bar{v}_i, \delta_i, \bar{\delta}_i) = \frac{m_i}{2} \|v_i - \bar{v}_i\|^2 + \frac{1}{2} \|\delta_i - \bar{\delta}_i\|^2 \quad (\text{B.1})$$

certifies equilibrium independent output strict passivity since

$$\begin{aligned} \dot{S}_i(v_i, \bar{v}_i, \delta_i, \bar{\delta}_i) &= (v_i - \bar{v}_i)^\top (-v_i + v^d - \delta_i + u_i) + (\delta_i - \bar{\delta}_i)^\top (v_i - v^d) \\ &= -\|v_i - \bar{v}_i\|^2 + (v_i - \bar{v}_i)^\top (u_i - \bar{u}_i) \end{aligned}$$

where we have used  $\bar{v}_i = v^d$  and  $\bar{u}_i = \bar{\delta}_i$  in the second equality.

For the suspended load, described by (7.7), the storage function

$$S_L(v_{N+1}, \bar{v}_{N+1}) = \frac{m_L}{2} \|v_{N+1} - \bar{v}_{N+1}\|^2 \quad (\text{B.2})$$

can be used to show that it is equilibrium independent passive since

$$\begin{aligned} \dot{S}_L(v_{N+1}, \bar{v}_{N+1}) &= (v_{N+1} - \bar{v}_{N+1})^\top (-m_L g + u_{N+1}) \\ &= (v_{N+1} - \bar{v}_{N+1})^\top (u_{N+1} - \bar{u}_{N+1}) \end{aligned}$$

where  $\bar{u}_{N+1} = m_L g$ .

For each edge  $\ell = 1, \dots, T$ , described by (7.12) with the control strategy in (7.13), the storage function is

$$\begin{aligned} R_\ell(r_\ell, \bar{r}_\ell) &= \int_{\bar{r}_\ell}^{r_\ell} (h_\ell(\zeta) - h_\ell(\bar{r}_\ell)) d\zeta \\ &= \sum_{i \in \{x, y, z\}} \int_{\bar{r}_\ell^i}^{r_\ell^i} (\sigma_\ell^i(\zeta^i) - \sigma_\ell^i(\bar{r}_\ell^i)) d\zeta^i \end{aligned} \quad (\text{B.3})$$

where the second equality holds since the curl of  $h_\ell$  in (7.13) is zero implying path independence of the integral. The storage function  $R_\ell$  is zero for  $r_\ell = \bar{r}_\ell$  and

## B. Stability Analyses

---

strictly positive for all  $r_\ell \neq \bar{r}_\ell$  since  $\sigma_\ell^x$ ,  $\sigma_\ell^y$ , and  $\sigma_\ell^z$  are strictly increasing and onto for  $\ell = 1, \dots, T$ . This storage function certifies each subsystem is equilibrium independent passive since

$$\begin{aligned}\dot{R}_\ell(r_\ell, \bar{r}_\ell) &= \begin{bmatrix} \sigma_\ell^x(r_\ell^x) - \sigma_\ell^x(\bar{r}_\ell^x) \\ \sigma_\ell^y(r_\ell^y) - \sigma_\ell^y(\bar{r}_\ell^y) \\ \sigma_\ell^z(r_\ell^z) - \sigma_\ell^z(\bar{r}_\ell^z) \end{bmatrix}^\top w_\ell \\ &= (h_\ell(r_\ell) - h_\ell(\bar{r}_\ell))^\top (w_\ell - \bar{w}_\ell)\end{aligned}$$

where  $\bar{w}_\ell = 0$ .

For each edge  $\ell = T + 1, \dots, E$ , described by (7.12) and (7.8), the storage function is

$$R_\ell(r_\ell, \bar{r}_\ell) = \int_{\|\bar{r}_\ell\|}^{\|r_\ell\|} \sigma_\ell(\zeta) d\zeta - \sigma_\ell(\|\bar{r}_\ell\|) \frac{\bar{r}_\ell^\top (r_\ell - \bar{r}_\ell)}{\|\bar{r}_\ell\|}. \quad (\text{B.4})$$

Clearly,  $R_\ell(\bar{r}_\ell, \bar{r}_\ell) = 0$  and by calculating the Hessian of  $R_\ell$  we can show that it is positive definite in a neighborhood of  $\bar{r}_\ell$ .

The gradient of the storage function is

$$\begin{aligned}\nabla_{r_\ell} R_\ell(r_\ell, \bar{r}_\ell) &= \sigma_\ell(\|r_\ell\|) \frac{r_\ell}{\|r_\ell\|} - \sigma_\ell(\|\bar{r}_\ell\|) \frac{\bar{r}_\ell}{\|\bar{r}_\ell\|} \\ &= h_\ell(r_\ell) - h_\ell(\bar{r}_\ell)\end{aligned}$$

and the Hessian is

$$\begin{aligned}H_{r_\ell} R_\ell(r_\ell, \bar{r}_\ell) &= \nabla_{r_\ell} \left( \sigma_\ell(\|r_\ell\|) \frac{r_\ell^\top}{\|r_\ell\|} \right) \\ &= \frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|} I_3 + \nabla_{r_\ell} \left( \frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|} \right) r_\ell^\top \\ &= \frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|} I_3 + \left( \sigma'_\ell(\|r_\ell\|) - \frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|^2} \right) \frac{1}{r_\ell^\top r_\ell} r_\ell r_\ell^\top\end{aligned}$$

where  $I_n \in \mathbb{R}^{n \times n}$  is the identity matrix. Note that the Hessian is the sum of a scaled identity matrix and a rank one matrix. Therefore, the eigenvalues of the Hessian are

$$\frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|}$$

with multiplicity 2 and

$$\frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|} + \left( \sigma'_\ell(\|r_\ell\|) - \frac{\sigma_\ell(\|r_\ell\|)}{\|r_\ell\|^2} \right) = \sigma'_\ell(\|r_\ell\|)$$

with multiplicity 1. Thus, if

$$\sigma_\ell(\|\bar{r}_\ell\|) > 0 \quad \text{and} \quad \sigma'_\ell(\|\bar{r}_\ell\|) > 0 \quad (\text{B.5})$$

---

then the Hessian is positive definite at  $\bar{r}_\ell$  which implies that  $R_\ell(r_\ell, \bar{r}_\ell) > 0$  for all  $r_\ell \neq \bar{r}_\ell$  in a neighborhood of  $\bar{r}_\ell$ . The first condition,  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$ , holds whenever the spring is in tension and the second condition,  $\sigma'_\ell(\|\bar{r}_\ell\|) > 0$ , always holds since  $\sigma_\ell$  is increasing.

The Lie derivative of this storage function is

$$\dot{R}_\ell(r_\ell, \bar{r}_\ell) = (h_\ell(r_\ell) - h_\ell(\bar{r}_\ell))^\top (w_\ell - \bar{w}_\ell)$$

where  $\bar{w}_\ell = 0$ . Therefore, this storage function certifies the edge subsystems are equilibrium independent passive in a neighborhood of  $\bar{r}_\ell$  for any  $\bar{r}_\ell$  satisfying  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$ .

In order to characterize the equilibrium points of the system we must consider configurations with specific numbers of UAVs. For example with  $N = 2$ , the set of equilibria of the interconnected system in Figure 7.2 with the UAV, edge, and load subsystems described by (7.11), (7.12), and (7.7) respectively, is given by

$$\mathcal{E} = \left\{ (\bar{v}, \bar{\delta}, \bar{r}) \left| \begin{array}{l} \bar{v}_i = v^d \text{ for } i = 1, \dots, 3 \\ m_L g = h_2(\bar{r}_2) + h_3(\bar{r}_3) \\ \bar{\delta} = \begin{bmatrix} -h_1(\bar{r}_2 - \bar{r}_3) - h_2(\bar{r}_2) \\ h_1(\bar{r}_2 - \bar{r}_3) - h_3(\bar{r}_3) \end{bmatrix} \end{array} \right. \right\}.$$

where the geometric relation  $\bar{r}_1 = \bar{r}_2 - \bar{r}_3$  is used in the last equation. Since the functions  $h_1, \dots, h_3$  are increasing and onto in each coordinate there exists a unique  $\bar{\delta}$  for all values of  $\bar{r}_2$  and  $\bar{r}_3$ . Therefore, there is an equilibrium point for any  $r_2$  and  $r_3$  satisfying  $m_L g = h_2(r_2) + h_3(r_3)$ . For configurations with  $N > 2$  UAVs the set of equilibria  $\mathcal{E}$  is of a similar form and is a continuum of points.

**Theorem B.1** (restated from Theorem 7.1). *Any equilibrium point  $(\bar{v}, \bar{\delta}, \bar{r}) \in \mathcal{E}$  of the interconnected system in Figure 7.2 that satisfies  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$  for  $\ell = T+1, \dots, E$  is stable.*

*Proof.* By combining the subsystem storage functions (B.1)-(B.4) we get the candidate Lyapunov function

$$\begin{aligned} V(v, \delta, r) &= \sum_{i=1}^N S_i(v_i, \bar{v}_i, \delta_i, \bar{\delta}_i) \\ &\quad + S_L(v_{N+1}, \bar{v}_{N+1}) + \sum_{\ell=1}^E R_\ell(r_\ell, \bar{r}_\ell). \end{aligned} \tag{B.6}$$

for any equilibrium point  $(\bar{v}, \bar{\delta}, \bar{r}) \in \mathcal{E}$ . By the definitions of the subsystem storage functions  $V(\bar{v}, \bar{\delta}, \bar{r}) = 0$  and  $V$  is positive definite in  $v$ ,  $\delta$ , and  $r_\ell$  for  $\ell = 1, \dots, T$ . Furthermore, since  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$  for  $\ell = T+1, \dots, E$  then there will exist an open set  $\mathcal{R}$  containing  $\bar{r}_\ell$  such that  $V(\bar{v}, \bar{\delta}, \bar{r}) = 0$  and  $V(\bar{v}, \bar{\delta}, r) > 0$  for any  $\{r_\ell \in \mathcal{R} \mid r_\ell \neq \bar{r}_\ell \text{ for } \ell = T+1, \dots, E\}$ .

## B. Stability Analyses

---

The Lie derivative of  $V$  is

$$\begin{aligned}
& \dot{V}(v, \delta, r) \\
&= \sum_{i=1}^N \left( -\|v_i - \bar{v}_i\|^2 + (v_i - \bar{v}_i)^\top (u_i - \bar{u}_i) \right) \\
&\quad + (v_{N+1} - \bar{v}_{N+1})^\top (u_{N+1} - \bar{u}_{N+1}) \\
&\quad + \sum_{\ell=1}^E (h_\ell(r_\ell) - h_\ell(\bar{r}_\ell))^\top (w_\ell - \bar{w}_\ell) \\
&= -\sum_{i=1}^N \|v_i - \bar{v}_i\|^2 + (v - \bar{v})^\top (u - \bar{u}) \\
&\quad + (h(r) - h(\bar{r}))^\top (w - \bar{w}) \\
&= -\sum_{i=1}^N \|v_i - \bar{v}_i\|^2 - (v - \bar{v})^\top D(h(r) - h(\bar{r})) \\
&\quad + (v - \bar{v})^\top D(h(r) - h(\bar{r})) \\
&= -\sum_{i=1}^N \|v_i - \bar{v}_i\|^2 \leq 0
\end{aligned}$$

where the third inequality follows from  $u = -Dh(r)$  and  $w = D^\top v$ . Hence, any equilibrium point  $(\bar{v}, \bar{\delta}, \bar{r})$  satisfying  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$  for  $\ell = T+1, \dots, E$  is stable.  $\square$

**Remark B.1.** The assumption that  $\sigma_\ell(\|\bar{r}_\ell\|) > 0$  for  $\ell = T+1, \dots, E$  is not restrictive because it is only true when the cables between the UAVs and the load are in tension. This condition is expected in normal operation and desired so that there is no slack in the cables.

# References

- [1] E. Abdel-Rahman, A. Nayfeh, and Z. Masoud. Dynamics and control of cranes: A review. *Journal of Vibration and Control*, 9(7):863–908, 2003. doi: 10.1177/107754603031852.
- [2] A. P. Aguiar and R. Ghabcheloo. Coordinated Path-Following Control of Multiple Autonomous Underwater Vehicles. In *Proceedings of the International Offshore and Polar Engineering Conference*, pages 4345 – 4350, Lisbon, Portugal, 2007. International Society of Offshore and Polar Engineers. ISBN 9781880653685. doi: ISOPE-I-07-006.
- [3] A. P. Aguiar and A. M. Pascoal. Coordinated path-following control for nonlinear systems with logic-based communication. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1473–1479, 2007. ISBN 1424414989. doi: 10.1109/CDC.2007.4434835.
- [4] Amazon. Amazon.com, 2016. URL <http://www.amazon.com/b?node=8037720011>.
- [5] F. Amrouche. *Fundamentals of Multibody Dynamics: Theory and Applications*. Birkhäuser Boston, 2006. ISBN 978-0-8176-4406-2. doi: 10.1007/b137682.
- [6] R. Andrade, G. V. Raffo, and J. E. Normey-Rico. Model Predictive Control of a Tilt-Rotor UAV for Load Transportation. In *Proceedings of the European Control Conference (ECC)*, pages 2165–2170, 2016. ISBN 9781509025916. doi: 10.1109/ECC.2016.7810612.
- [7] D. Angeli and E. D. Sontag. Forward completeness, unboundedness observability, and their Lyapunov characterizations. *Systems & Control Letters*, 38(4):209–217, 1999. ISSN 01676911. doi: 10.1016/S0167-6911(99)00055-9.
- [8] M. Arcak. Passivity as a Design Tool for Group Coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, 2007.
- [9] Ardupilot.com. Ardupilot - Open source autopilot, 2016. URL <http://ardupilot.com>.

- [10] H. Bai, M. Arcak, and J. T. Wen. *Cooperative control design: A systematic, passivity-based approach*. Springer-Verlag New York, 2011. ISBN 9781461400134. doi: 10.1007/978-1-4614-0014-1.
- [11] Beagleboard. Homepage of the BeagleBone project: <http://beagleboard.org>, 2013. URL <http://beagleboard.org/>.
- [12] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. ISBN 9780691149219.
- [13] M. Bernard and K. Kondak. Generic slung load transportation system using small size helicopters. *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 3258–3264, 2009. doi: 10.1109/ROBOT.2009.5152382.
- [14] M. Bernard, K. Kondak, I. Maza, and A. Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914–931, 2011. doi: 10.1002/rob.
- [15] M. Bisgaard. *Modeling, Estimation, and Control of Helicopter Slung Load System*. PhD thesis, Aalborg University, 2008.
- [16] M. Bisgaard, A. la Cour-Harbo, and J. Bendtsen. Input Shaping for Helicopter Slung Load Swing Reduction. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 2008.
- [17] M. Bisgaard, J. D. Bendtsen, and A. L. Cour-Harbo. Modeling of Generic Slung Load System. *Journal of Guidance, Control, and Dynamics*, 32(2): 573–585, mar 2009. ISSN 0731-5090. doi: 10.2514/1.36539.
- [18] M. Bisgaard, A. Cour-harbo, and J. D. Bendtsen. Swing Damping for Helicopter Slung Load Systems using Delayed Feedback. *Proceedings of AIAA Conference on Guidance, Navigation, and Control*, pages 1–11, 2009.
- [19] M. Bisgaard, A. la Cour-Harbo, and J. Dimon Bendtsen. Adaptive control system for autonomous helicopter slung load operations. *Control Engineering Practice*, 18(7):800–811, jul 2010. ISSN 09670661. doi: 10.1016/j.conengprac.2010.01.017.
- [20] K. E. Brenan, S. L. V. Campbell, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1996. ISBN 0898713536.
- [21] M. Bürger, D. Zelazo, and F. Allgöwer. Duality and network theory in passivity-based cooperative control. *Automatica*, 50(8):2051–2061, 2014. ISSN 00051098. doi: 10.1016/j.automatica.2014.06.002.
- [22] L. Cicolani and G. Kanning. Equations of motion of slung-load systems, including multilift systems. *NASA, Office of Management Scientific and Technical Information Program*, Technical(3280), 1992.

- 
- [23] M. Cline and D. Pai. Post-Stabilization for Rigid Body Simulation with Contact and Constraints. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003., 3(1):3744–3751, 2003.
  - [24] Cnbc.com. Dominos Pizza Drone Delivery, 2016.
  - [25] Delftdynamics.nl. DroneCatcher catches drone, 2015. URL <http://www.delftdynamics.nl/index.php/en/news-en/117-dronecatcher-ca...>.
  - [26] D. V. Dimarogonas and K. H. Johansson. Stability analysis for multi-agent systems using the incidence matrix: Quantized communication and formation control. *Automatica*, 46(4):695–700, 2010. ISSN 00051098. doi: 10.1016/j.automatica.2010.01.012.
  - [27] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson. Distributed Event-Triggered Control for Multi-Agent Systems. *IEEE Transactions on Automatic Control*, 57(5), 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2174666.
  - [28] O. Egeland and J. T. Gravdahl. *Modeling and Simulation for automatic control*. Marine Cybernetics, 2002. ISBN 9788292356012.
  - [29] J. Estévez, J. M. Lopez-Gude, and M. Graña. Quasi-stationary state transportation of a hose with quadrotors. *Robotics and Autonomous Systems*, 63:187–194, 2015. ISSN 09218890. doi: 10.1016/j.robot.2014.09.004.
  - [30] D. D. Falco, E. Pennestri, and L. Vita. The udwadia-kalaba formulation: A report on its numerical efficiency in multibody dynamics simulations and on its teaching effectiveness. In J. G. O. J.M. Goicolea, J. Cuadrado, editor, *Proceedings of the 2005 ECCOMAS Thematic Conference on Multibody Dynamics*, pages 21–24, Madrid, Spain, 2005.
  - [31] D. D. Falco, E. Pennestri, and L. Vita. Investigation of the influence of pseudoinverse matrix calculations on multibody dynamics simulations by means of the Udwadia-Kalaba formulation. *Journal of aerospace engineering*, 22(October):365–372, 2009.
  - [32] R. T. Farouki. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*. Springer-Verlag Berlin Heidelberg, 1 edition, jan 2008. ISBN 978-3-540-73398-0. doi: 10.1007/978-3-540-73398-0.
  - [33] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia. Learning swing-free trajectories for UAVs with a suspended load. *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4902–4909, 2013. doi: 10.1109/ICRA.2013.6631277.
  - [34] J. Fax and R. Murray. Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, sep 2004. ISSN 0018-9286. doi: 10.1109/TAC.2004.834433.

- [35] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *The International Journal of Robotics Research*, 30(3), sep 2010. ISSN 0278-3649. doi: 10.1177/0278364910382803.
- [36] T. Fossen, M. Breivik, and R. Skjetne. Line-of-Sight Path Following of Underactuated Marine Craft. In *Proceedings of the IFAC MCMC*, Girona, Spain, 2003. IFAC.
- [37] T. I. Fossen. *Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.
- [38] T. I. Fossen. Mathematical Models for Control of Aircraft and Satellites, 2013.
- [39] T. I. Fossen, A. Loría, and A. Teel. Theorem for UGAS and ULES of (passive) nonautonomous systems: robust control of mechanical systems and ships. *International Journal of Robust and Nonlinear Control*, 11(2):95–108, 2001. ISSN 10498923. doi: 10.1002/rnc.551.
- [40] Futek.com. LSB200 S-Beam, 2016. URL <http://www.futek.com/lsb200/overview.aspx>.
- [41] R. Ghacheloo, a. P. Aguiar, a. Pascoal, C. Silvestre, I. Kaminer, and J. Hespanha. Coordinated Path-Following in the Presence of Communication Losses and Time Delays. *SIAM Journal on Control and Optimization*, 48: 234–265, 2009. ISSN 0363-0129. doi: 10.1137/060678993.
- [42] J. Ginsberg. *Engineering Dynamics*. Cambridge University Press, 2008. ISBN 9780521883030.
- [43] Github.com. LSTS: Underwater Systems and Technology Laboratory, 2016. URL <https://github.com/LSTS/>.
- [44] F. A. Goodarzi and T. Lee. Dynamics and Control of Quadrotor UAVs Transporting a Rigid Body Connected via Flexible Cables. In *Proceedings of the American Control Conference*, pages 4677–4682, 2015. ISBN 9781479986866. doi: 10.1109/ACC.2015.7172066.
- [45] F. A. Goodarzi and T. Lee. Stabilization of a Rigid Body Payload with Multiple Cooperative Quadrotors. *Journal of Dynamic Systems, Measurement, and Control*, 138(12), 2016. doi: 10.1115/1.4033945.
- [46] F. A. Goodarzi, D. Lee, and T. Lee. Geometric Stabilization of Quadrotor UAV with a Payload Connected by Flexible Cable. In *Proceedings of the American Control Conference (ACC)*, pages 4925–4930. IEEE, sep 2014.
- [47] F. A. Goodarzi, D. Lee, and T. Lee. Geometric Control of a Quadrotor UAV Transporting a Payload Connected via Flexible Cable. *International Journal of Control, Automation, and Systems*, 13(6):1486–1498, 2015. ISSN 1598-6446. doi: 10.1007/s12555-014-0304-0.
- [48] D. T. Greenwood. *Advanced Dynamics*. Cambridge University Press, New York, 2003. ISBN 9780521826129.

- [49] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. Garc. IDA-PBC Methodology for a Quadrotor UAV Transporting a Cable-Suspended Payload. In *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 470 – 476, Denver, CO, 2015. IEEE. ISBN 9781479960095. doi: 10.1109/ICUAS.2015.7152325.
- [50] M. E. Guerrero-Sánchez, D. A. Mercado-Ravell, R. Lozano, and C. D. García-Beltrán. Swing-attenuation for a quadrotor transporting a cable-suspended payload. *ISA Transactions*, 2017. doi: <http://dx.doi.org/10.1016/j.isatra.2017.01.027>.
- [51] J. Hauser and R. Hindman. Aggressive flight maneuvers. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 86–91, 1997. ISBN 0-7803-4187-2. doi: 10.1109/CDC.1997.649490.
- [52] G. H. Hines, M. Arcak, and A. K. Packard. Equilibrium-independent passivity: A new definition and numerical certification. *Automatica*, 47(9): 1949–1956, 2011. ISSN 00051098. doi: 10.1016/j.automatica.2011.05.011.
- [53] J. Y. Hung. Feedback control with posicast. *IEEE Transactions on Industrial Electronics*, 50(1):94–99, 2003. ISSN 02780046. doi: 10.1109/TIE.2002.804979.
- [54] I. Ihle, J. Jouffroy, and T. Fossen. Robust formation control of marine craft using lagrange multipliers. In *Group Coordination and Cooperative Control*, pages 113–129. Springer, 2006. URL <http://www.springerlink.com/index/D255221718482627.pdf>.
- [55] I.-A. F. Ihle, M. Arcak, and T. I. Fossen. Passivity-based designs for synchronized path-following. *Automatica*, 43(9):1508–1518, sep 2007. ISSN 00051098. doi: 10.1016/j.automatica.2007.02.018.
- [56] Insitu.com. Insitu - ScanEagle, 2016. URL <http://www.insitu.com/information-delivery/unmanned-systems/scaneagle>.
- [57] P. A. Ioannou and J. Sun. *Robust Adaptive Control*. Dover Publications, 2012. ISBN 9780486498171.
- [58] Q. Jiang and V. Kumar. The inverse kinematics of cooperative transport with multiple aerial robots. *IEEE Transactions on Robotics*, 29(1):136–145, 2013. ISSN 15523098. doi: 10.1109/TRO.2012.2218991.
- [59] T. Kane and D. Levinson. *Dynamics: Theory and Applications*. McGraw-Hill, 1985. ISBN 9780070378469.
- [60] A. Khalid, J. R. Huey, W. E. Singhose, J. Lawrence, and D. Frakes. Human Operator Performance Testing Using an Input-Shaped Bridge Crane. *Journal of Dynamic Systems, Measurement, and Control*, 128(4):835–841, 2006. ISSN 00220434. doi: 10.1115/1.2361321.

- [61] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002. ISBN 0131227408.
- [62] H. J. Kim, M. Kim, H. Lim, C. Park, S. Yoon, D. Lee, H. Choi, G. Oh, J. Park, and Y. Kim. Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV. *IEEE/ASME Transactions on Mechatronics*, 18(4):1320–1333, 2013. ISSN 10834435. doi: 10.1109/TMECH.2013.2247411.
- [63] K. Klausen, T. I. Fossen, and T. A. Johansen. Suspended Load Motion Control using Multicopters. In *Proceedings of the Mediterranean Conference of Control and Automation (MED)*, pages 1371–1376, Palermo, 2014. IEEE. ISBN 9781479958993. doi: 10.1109/MED.2014.6961567.
- [64] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control of a Multirotor UAV with Suspended Load. In *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 176 – 184, Denver, CO, 2015. IEEE. doi: 10.1109/ICUAS.2015.7152289.
- [65] K. Klausen, T. I. Fossen, T. A. Johansen, and A. P. Aguiar. Cooperative path-following for multirotor UAVs with a suspended payload. *Proceedings of the 2015 IEEE Conference on Control Applications (CCA)*, pages 1354–1360, 2015. doi: 10.1109/CCA.2015.7320800.
- [66] K. Klausen, J. B. Moe, J. C. Van Den Hoorn, A. Gomola, T. I. Fossen, and T. A. Johansen. Coordinated Control Concept for Recovery of a Fixed-Wing UAV on a Ship using a Net Carried by Multirotor UAVs. In *Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 964–973, 2016. ISBN 9781467393331. doi: 10.1109/ICUAS.2016.7502640.
- [67] K. Klausen, T. I. Fossen, and T. A. Johansen. Nonlinear Control with Swing Damping of a Multirotor UAV with Suspended Load. *Journal of Intelligent & Robotic Systems*, 2017. ISSN 1573-0409. doi: 10.1007/s10846-017-0509-6.
- [68] K. Klausen, T. I. Fossen, and T. A. Johansen. Autonomous Recovery of a Fixed-Wing UAV Using a Net Suspended by Two Multirotor UAVs. *Journal of Field Robotics (submitted)*, 2017.
- [69] K. Klausen, C. Meissen, T. I. Fossen, M. Arcak, and T. A. Johansen. Cooperative Control for Multirotors Transporting an Unknown Suspended Load with Environmental Disturbances. *IEEE Transactions on Control Systems Technology (submitted)*, 2017.
- [70] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and adaptive control design*. John Wiley, New York, 1995.
- [71] M. Laiacker, K. Kondak, M. Schwarzbach, and T. Muskardin. Vision aided automatic landing system for fixed wing UAV. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pages 2971–2976, 2013. ISSN 21530858. doi: 10.1109/IROS.2013.6696777.

- [72] B. Y. Lee, H. I. Lee, D. W. Yoo, G. H. Moon, D. Y. Lee, Y. Y. Kim, and M. J. Tahk. Study on payload stabilization method with the slung-load transportation system using a quad-rotor. In *Proceedings of the 2015 European Control Conference (ECC)*, pages 2097–2102, 2015. ISBN 9783952426937. doi: 10.1109/ECC.2015.7330849.
- [73] T. Lee. Geometric Control of Multiple Quadrotor UAVs Transporting a Cable-Suspended Rigid Body. In *Proceedings of the 53rd Conference on Decision and Control (CDC)*, 2014. ISBN 9781467360883.
- [74] T. Lee. Collision avoidance for quadrotor UAVs transporting a payload via Voronoi tessellation. In *Proceedings of the American Control Conference (ACC)*, volume 2015-July, pages 1842–1848, 2015. ISBN 9781479986842. doi: 10.1109/ACC.2015.7171001.
- [75] T. Lee. Geometric Control of Quadrotor UAVs Transporting a Cable-Suspended Rigid Body. *IEEE Transactions on Control Systems Technology*, 2017. doi: 10.1109/TCST.2017.2656060.
- [76] T. Lee, K. Sreenath, and V. Kumar. Geometric Control of Cooperating Multiple Quadrotor UAVs with a Suspended Payload. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 5510–5515, 2013. ISBN 9781467357166.
- [77] Z. Li, J. F. Horn, and J. W. Langelaan. Coordinated Transport of a Slung Load by a Team of Autonomous Rotorcraft. *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2014. doi: 10.2514/6.2014-0968.
- [78] Lovdata.no. Forskrift om luftfartøy som ikke har fører om bord mv. (Norwegian regulations regarding RPAS), 2017. URL <https://lovdata.no/forskrift/2015-11-30-1404>.
- [79] Lsts.fe.up.pt. GLUED: GNU/Linux Uniform Environment Distribution, 2017. URL <http://lsts.fe.up.pt/toolchain/glued>.
- [80] R. Mahony, V. Kumar, and P. Corke. Modeling, Estimation, and Control of Quadrotor. *IEEE Robotics and Automation Magazine*, 19(3):20–32, sep 2012. doi: 10.1109/MRA.2012.2206474.
- [81] R. Martins, E. R. B. Marques, J. B. Sousa, P. S. Dias, J. Pinto, and F. L. Pereira. IMC: A communication protocol for networked vehicles and sensors. In *Proceedings of the IEEE OCEANS Conference*, Bremen, 2009. IEEE. ISBN 978-1-4244-2522-8. doi: 10.1109/OCEANSE.2009.5278245.
- [82] MAVlink.org. MAVLink Micro Air Vehicle Communication Protocol, 2017. URL <http://qgroundcontrol.org/mavlink/start>.
- [83] I. Maza, K. Kondak, M. Bernard, and A. Ollero. Multi-UAV Cooperation and Control for Load Transportation and Deployment. *Journal of Intelligent and Robotic Systems*, 57(1-4):417–449, aug 2009. ISSN 0921-0296. doi: 10.1007/s10846-009-9352-8.

- [84] C. Meissen, K. Klausen, M. Arcak, T. I. Fossen, and A. Packard. Passivity-based Formation Control for UAVs with a Suspended Load. In *Proceedings of the IFAC World Congress*, 2017.
- [85] D. Mellinger, M. Shomin, N. Michael, and V. Kumar. Cooperative grasping and transport using multiple quadrotors. *Distributed autonomous robotic systems*, 83:545–558, 2013.
- [86] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, sep 2010. ISSN 0929-5593. doi: 10.1007/s10514-010-9205-0.
- [87] MotionGenesis. <http://motiongenesis.com>, 2015. URL <http://www.motiongenesis.com/>.
- [88] Mtu.edu. Michigan Tech Robotic Falconry, 2016. URL <http://me.sites.mtu.edu/rastgaar/home/news/>.
- [89] T. Muskardin, G. Balmer, L. Persson, S. Wlach, M. Laiacker, A. Ollero, and K. Kondak. A Novel Landing System to Increase Payload Capacity and Operational Availability of High Altitude Long Endurance UAVs. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2017. ISSN 0921-0296. doi: 10.1007/s10846-017-0475-z.
- [90] Nextgenerationvision.fr. Drone interception, 2014. URL <https://youtu.be/APWG3VEGbJw>.
- [91] Nimbus.unl.edu. Co-Aerial-Ecologist: Robotic Water Sampling and Sensing in the Wild, 2017. URL <https://nimbus.unl.edu/projects/co-aerial-ecologist-robotic-water-sampling-and-sensing-in-the-wild/>.
- [92] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, 95(1):215–233, jan 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.887293.
- [93] H. Omar. Control optimization method for helicopters carrying suspended loads, 2012. URL <http://www.google.com/patents/US8190307>.
- [94] H. M. Omar. Anti-swing controller based on time-delayed feedback for helicopter slung load system near hover. *Journal of Aerospace Technology and Management*, 4(3):297–305, 2012. ISSN 19849648. doi: 10.5028/jatm.2012.04031512.
- [95] H. M. Omar and A. H. Nayfeh. Anti-swing control of gantry and tower cranes using fuzzy and time delayed feedback with friction compensation. *Journal of Shock and Vibration*, 12(2):73–89, 2005. ISSN 1070-9622. doi: 10.1155/2005/890127.
- [96] R. Ortega, A. Van der Schaft, B. Maschke, and G. Escobar. Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. *Automatica*, 38(4):585–596, 2002. ISSN 00051098. doi: 10.1016/S0005-1098(01)00278-3.

- [97] I. Palunko, R. Fierro, and P. Cruz. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pages 2691–2697, may 2012. doi: 10.1109/ICRA.2012.6225213.
- [98] Paparazziuav.org. Paparazzi UAV, 2017. URL [http://wiki.paparazziuav.org/wiki/Main{\\_}Page](http://wiki.paparazziuav.org/wiki/Main{_}Page).
- [99] D. W. Peng, T. Singh, and M. Milano. Zero-phase velocity tracking of vibratory systems. *Control Engineering Practice*, 40:93–101, 2015. ISSN 09670661. doi: 10.1016/j.conengprac.2015.03.008.
- [100] P. O. Pereira, M. Herzog, and D. V. Dimarogonas. Slung load transportation with a single aerial vehicle and disturbance removal. *Proceedings of the 24th Mediterranean Conference on Control and Automation*, pages 671–676, 2016. doi: 10.1109/MED.2016.7536040.
- [101] K. Y. Pettersen, J. T. Gravdahl, and H. Nijmeijer. *Group Coordination and Cooperative Control*. Springer Berlin Heidelberg, 2006. doi: 10.1007/11505532.
- [102] E. Peymani and T. Fossen. Leader-follower formation of marine craft using constraint forces and lagrange multipliers. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2447–2452, Hawaii, dec 2012. ISBN 978-1-4673-2066-5. doi: 10.1109/CDC.2012.6426592.
- [103] Piher.net. MTS-360, 2016. URL <http://www.piher.net/>.
- [104] J. Pinto, P. S. Dias, R. Gonçalves, E. Marques, G. Gonçalves, J. B. Sousa, and F. L. Pereira. Neptus – A Framework to Support a Mission Life Cycle. *Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC)*, 2006.
- [105] J. Pinto, P. Calado, J. Braga, P. Dias, R. Martins, E. Marques, and J. B. Sousa. Implementation of a control architecture for networked vehicle systems. *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, 270180:100–105, 2012. ISSN 14746670. doi: 10.3182/20120410-3-PT-4028.00018.
- [106] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa. The LSTS toolchain for networked vehicle systems. In *Proceedings of the MTS/IEEE OCEANS*, Bergen, 2013. MTS/IEEE. ISBN 9781479900015. doi: 10.1109/OCEANS-Bergen.2013.6608148.
- [107] Pixhawk.ethz.ch. PX4 Pixhawk, 2016. URL <https://pixhawk.ethz.ch/>.
- [108] Px4.io. PX4 Pro Autopilot, 2017. URL <http://px4.io/>.

- [109] F. C. Rego, A. P. Aguiar, and A. M. Pascoal. A packet loss compliant logic-based communication algorithm for cooperative path-following control. In *Proceedings of the 9th IFAC Conference on Control Applications in Marine Systems (CAMS)*, pages 262–267, Osaka, Japan, 2013. doi: 10.3182/20130918-4-JP-3022.00055.
- [110] W. R. W. Ren, R. Beard, and E. Atkins. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005 American Control Conference (CDC)*, pages 1859–1864, 2005. ISBN 0-7803-9098-9. doi: 10.1109/ACC.2005.1470239.
- [111] T. S. Richardson, C. G. Jones, A. Likhoded, E. Sparks, A. Jordan, I. Cowling, and S. Willcox. Automated vision-based recovery of a rotary wing unmanned aerial vehicle onto a moving platform. *Journal of Field Robotics*, 30(5):667–684, 2013. ISSN 15564959. doi: 10.1002/rob.21467.
- [112] J. Rife, S. Khanafseh, S. Pullen, D. De Lorenzo, U. S. Kim, M. Koenig, T. Y. Chiou, B. Kempny, and B. Pervan. Navigation, Interference Suppression, and Fault Monitoring in the Sea-Based Joint Precision Approach and Landing System. *Proceedings of the IEEE*, 96(12):1958–1975, 2008. ISSN 0018-9219. doi: Doi10.1109/Jproc.2008.2006107.
- [113] J.-H. B. Røli. *Cooperative Control and RTK Navigation System for Multirotors*. Master’s thesis, Norwegian University of Science and Technology, 2015. URL <https://brage.bibsys.no/xmlui/handle/11250/2352408>.
- [114] Ros.org. Robot Operating System (ROS), 2017. URL <http://www.ros.org/>.
- [115] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre. Landing of a Quadrotor on a Moving Target Using Dynamic Image-Based Visual Servo Control. *IEEE Transactions on Robotics*, 32(6):1524–1535, 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2604495.
- [116] N. Singer, W. Singhose, and E. Kriikku. An Input Shaping Controller Enabling Cranes to Move without Sway. In *Proceedings of the 7th Topical meeting on Robotics and Remote Systems*, volume 1, pages 225–231. ANS, 1997. doi: 10.2172/491559.
- [117] N. C. Singer and W. P. Seering. Preshaping Command Inputs to Reduce System Vibration. *Journal of Dynamic Systems, Measurement, and Control*, 112(1):76–82, 1990. ISSN 00220434. doi: 10.1115/1.2894142.
- [118] T. Singh and W. Singhose. Input Shaping / Time Delay Control of Maneuvering Flexible Structures. In *Proceedings of the 2002 American Control Conference*, pages 1717 – 1731 (3), NY, USA, 2002. IEEE.
- [119] W. Singhose. Command shaping for flexible systems: A review of the first 50 years. *International Journal of Precision Engineering and Manufacturing*, 10 (4):153–168, 2009. ISSN 12298557. doi: 10.1007/s12541-009-0084-2.

- [120] R. Skjetne, I. Ihle, and T. I. Fossen. Formation Control by Synchronizing multiple maneuvering systems. In *Proceedings of the 2003 IFAC MCMC*, 2003.
- [121] R. Skjetne, T. I. Fossen, and P. V. Kokotović. Robust output maneuvering for a class of nonlinear systems. *Automatica*, 40(3):373–383, mar 2004. ISSN 00051098. doi: 10.1016/j.automatica.2003.10.010.
- [122] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen. Autonomous Net Recovery of Fixed-Wing UAV with Single-Frequency Carrier-Phase Differential GNSS. *Aerospace and Electronic Systems Magazine, IEEE*, 30(5):18 – 27, 2015.
- [123] O. J. M. Smith. Posicast Control of Damped Oscillatory Systems. *Proceedings of the IRE*, 45(9):1249–1255, 1957. ISSN 0096-8390. doi: 10.1109/JRPROC.1957.278530.
- [124] K. Sreenath, T. Lee, and V. Kumar. Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2269–2274, 2013. ISBN 9781467357173. doi: 10.1109/CDC.2013.6760219.
- [125] K. Sreenath, N. Michael, and V. Kumar. Trajectory generation and control of a quadrotor with a cable-suspended load - A differentially-flat hybrid system. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, pages 4888–4895. Ieee, 2013. ISBN 978-1-4673-5643-5. doi: 10.1109/ICRA.2013.6631275.
- [126] M. Stokkeland, K. Klausen, and T. A. Johansen. Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection. *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 998–1007, 2015. doi: 10.1109/ICUAS.2015.7152389.
- [127] R. A. Stuckey. Mathematical Modelling of Helicopter Slung-Load Systems. Technical report, Department of Defence, DSTO, Australia, 2001. URL <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA401423>.
- [128] Suasnews.com. Maersk Drone Delivey, 2016. URL <https://www.suasnews.com/2016/03/maersk-makes-drone-delivery/>.
- [129] T. Takasu and A. Yasuda. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In *Proceedings of the International Symposium on GPS/GNSS*, Jeju, Korea, 2009.
- [130] S. Tang and V. Kumar. Mixed Integer Quadratic Program trajectory generation for a quadrotor with a cable-suspended payload. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015-June (June):2216–2222, 2015. ISSN 10504729. doi: 10.1109/ICRA.2015.7139492.

- [131] G. Tartaglione, E. D'Amato, M. Ariola, P. S. Rossi, and T. A. Johansen. Model predictive control for a multi-body slung-load system. *Robotics and Autonomous Systems*, 92:1–11, 2017. ISSN 09218890. doi: 10.1016/j.robot.2017.02.007.
- [132] Theverge.com. Tokyo police unveil net-wielding interceptor drone, 2015. URL <http://www.theverge.com/2015/12/11/9891128/tokyo-interceptor-net-drone>.
- [133] Theverge.com. This drone can detect and detonate land mines, 2016. URL <http://www.theverge.com/2016/7/19/12222104/landmine-detecting-drone-mine-kafon-drone>.
- [134] S. Thuroowgood, R. J. D. Moore, D. Soccol, M. Knight, and M. V. Srinivasan. A biologically inspired, vision-based guidance system for automatic landing of a fixed-wing aircraft. *Journal of Field Robotics*, 31(4): 699–727, 2014. ISSN 15564967. doi: 10.1002/rob.21527.
- [135] J. Trachte, F. Gonzalez, and A. McFadyen. Nonlinear model predictive control for a multi-rotor with heavy slung load. In *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1105–1110, 2014. ISBN 9781479923762. doi: 10.1109/ICUAS.2014.6842363.
- [136] U-blox.com. NEO/LEA-M8T, 2016. URL [u-blox.com](http://u-blox.com).
- [137] F. Udwadia and R. Kalaba. A new perspective on constrained motion. *Proceedings: Mathematical and Physical Sciences*, 439(2):407–410, 1992.
- [138] F. E. Udwadia. Equations of motion for mechanical systems: A unified approach. *Journal of Non-linear Mechanics*, 31(6):951–958, 1996.
- [139] F. E. Udwadia and P. Phohomsiri. Explicit Poincaré equations of motion for general constrained systems. Part I. Analytical results. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463 (2082):1421–1434, jun 2007. ISSN 1364-5021. doi: 10.1098/rspa.2007.1825.
- [140] T. K. Venugopalan, T. Taher, and G. Barbastathis. Autonomous landing of an Unmanned Aerial Vehicle on an autonomous marine vehicle. *Proceedings of the 2012 MTS/IEEE OCEANS*, 2012. doi: 10.1109/OCEANS.2012.6404893.
- [141] A. Zolich, T. A. Johansen, K. Cisek, and K. Klausen. Unmanned Aerial System Architecture for Maritime Missions. Design & Hardware Description. In *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015.