UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Vision-Based Control of an Unknown Suspended Payload with a Multirotor Unmanned Aerial Vehicle

Author: J.F. Slabber
Supervisor: Dr H.W. Jordaan

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering in the Faculty of Engineering at Stellenbosch University.

March 2021

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Abstract

In this thesis, a flight control system for a quadrotor transporting an unknown suspended payload is proposed. The suspended payload parameters, which include its mass and cable length, are unknown and direct measurements of its states are not available. The swinging motion of the suspended payload negatively affects the flight dynamics of the quadrotor. A vision-based control strategy is designed and verified in simulation to damp the swinging motion of the payload and ensure stable flight.

A feedforward-feedback adaptive control strategy is proposed that consists of a notch filter and Linear Quadratic Gaussian (LQG) controller with integral action. The control strategy adapts to the specific payload by estimating the unknown payload parameters. The notch filter serves as an input shaper to generate trajectories that suppress the generation of oscillations of the suspended payload while the LQG controller simultaneously control the quadrotor and suspended payload to actively damp the payload oscillations.

A vision-based state estimator is designed to provide payload state estimates for the optimal full-state feedback controller. A downward facing camera is attached to the bottom of the quadrotor to provide an Extended Kalman Filter (EKF) with swing angle measurements. A motion-based object detection algorithm is used to detect the suspended payload. This approach is taken to design a vision system that is independent of visual markers and can detect any arbitrary suspended payload attached to the quadrotor.

The proposed control strategy was implemented in the PX4 flight control stack and Robot Operating System (ROS) was used for the vision system. Software-in-the-loop (SITL) and hardware-in-the-loop (HITL) simulations were performed in the PX4 and Gazebo simulation environment to verify the proposed control strategy. The simulation results proved that the proposed vision-based control strategy is able to estimate the unknown payload parameters with good accuracy and use these estimates to adapt to different suspended payloads. The vision-based state estimator is able to provide accurate state estimates for full-state feedback control and the feedforward-feedback controller is able to damp the unwanted payload oscillations and ensure stable flight. The proposed control strategy is proved to be practically feasible and can be verified in practical flight tests.

Uittreksel

Hierdie tesis stel ’n vlugbeheerstelsel voor vir ’n vierrotor onbemande vliegtuig met ’n onbekende swaaiende loonvrag. Die loonvragparameters, wat die massa en kabellengte insluit, is onbekend en direkte metings van die swaaihoeke is nie moontlik nie. Die swaaibeweging van die loonvrag het ’n negatiewe effek op die vlugdinamika van die vierrotor vliegtuig. ’n Visie-gebaseerde beheerstrategie is ontwerp om die swaaibeweging van die loonvrag te demp en stabiele vlug te verseker. Die beheerstrategie is geverifieer in simulasie.

’n Aanpasbare vorentoevoer-terugvoer beheerstrategie wat bestaan uit ’n kerffilter (bandblokkeringsfilter) en ’n Lineêre Kwadratiese Gaussiese (LKG) beheerder word voorgestel. Die beheerstrategie skat die onbekende loonvragparameters af en gebruik die afskattings om aan te pas by die spesifieke loonvrag. Die kerffilter dien as ’n trajekbeplanner wat trajekte genereer om die ossilasies van die loonvrag te onderdruk, terwyl die LKG-beheerder die vierrotor voertuig en loonvrag gelyktydig beheer om die swaaibewegings van die loonvrag te demp.

’n Visie-gebaseerde toestandafskatter is ontwerp om die swaaihoek toestande van die loonvrag af te skat vir die optimale terugvoerbeheerder. ’n Kamera wat aan die onderkant van die vierrotor voertuig geplaas is word gebruik om swaaihoekmates te neem wat gebruik word in die toetstandafskatter. Die visiestelsel maak gebruik van ’n bewegingsgebaseerde algoritme om die loonvrag op te tel. Sodoende is die visiestelsel onafhanklik van die gebruik van visuele merkers vir die optel van die loonvrag en kan die visiestelsel toegepas word vir enige arbitrêre loonvrag wat gekoppel word aan die vierrotor voertuig.

Die voorgestelde beheerstrategie is geïmplementeer in die PX4 beheerargitektuur en die ROS sagteware is gebruik vir die implementasie van die visiestelsel. Sagteware-in-die-lus en hardeware-in-die-lus simulasies is uitgevoer in die PX4 en Gazebo simulasie omgewing om die voorgestelde beheerstrategie te verifieer. Die simulasies het bevestig dat die voorgestelde visie-gebaseerde beheerstrategie die onbekende loonvragparameters met goeie akkuraatheid kan afskat en hierdie afskattings gebruik om aan te pas by die verskillende loonvragte. Die simulasies het ook bevestig dat die visie-gebaseerde toestandafskatter die swaaihoek toestande van die loonvrag akkuraat kan afskat en dat die vorentoevoer-terugvoer beheerstrategie die ongewenste loonvrag ossilasies effektief kan demp en stabiele vlug verseker. Dit is bewys dat die voorgestelde beheerstrategie prakties uitvoerbaar is en kan dus geverifieer word in praktiese vlugtoetse.

Acknowledgements

I would like to thank the following people:

- My supervisor, Dr Willem Jordaan, for his guidance and support throughout the project.
- My family, for all their love and support.
- My friends, for their support and motivation througout my studies.
- All my colleagues in the ESL, for their support and willingness to help.

Contents

List of Figures

List of Tables

Abbreviations

$\mathscr{H}_\infty$	H-Infinity
2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
AVI	Approximate Value Iteration
BRIEF	Binary Robust Independent Elementary Features
CNN	Convolutional Neural Network
CoM	Center of Mass
CPU	Central Processing Unit
CSRT	Discriminative Correlation Filter with Channel and Spatial Reliability
DCF	Discriminative Correlation Filter
DOB	Disturbance Observer
DOF	Degrees of Freedom
EI	Extra Insensitive
EKF	Extended Kalman Filter
ESL	Electronic Systems Laboratory
FAST	Features from Accelerated Segment Test
FCU	Flight Controlling Unit
FFT	Fast Fourier Transform
FGD	Fading Gaussian Deterministic
FOV	Field of View
FPS	Frames per Second
GPS	Global Positioning System
HITL	Hardware-in-the-Loop
HoG	Histogram of Oriented Gradients
IDA-PBC	Interconnected and Damping Assignment-Passivity Based Controller
IMU	Inertial Measurement Unit
LPF	Low Pass Filter
LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
MRAC	Model Reference Adaptive Control
NED	North-East-Down
ORB	Oriented FAST and Rotated BRIEF

PD	Proportional Derivative
PI	Proportional Integral
PID	Proportional Integral Derivative
POSIX	Portable Operating System Interface
PRLQG	Parameter Robust Linear Quadratic Gaussian
PSO	Particle Swarm Optimization
QGC	QGroundControl
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RCAC	Retrospective Cost Adaptive Control
RLS	Recursive Least Squares
ROI	Region of Interest
ROS	Robotic Operating System
RTOS	Real Time Operating System
SIFT	Scale-Invarient Feature Transform
SITL	Software-in-the-Loop
SMC	Sliding Mode Control
SURF	Speeded Up Robust Features
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
ZVD	Zero Vibration and Derivative

Chapter 1

Introduction

1.1. Background

Over the past few decades, a rise in the popularity of unmanned aerial vehicle (UAV) applications have been seen, with multirotor applications being the most popular. Multirotors have desirable flight characteristics and capabilities making them ideal for certain applications. Multirotors have the ability to hover and vertically take-off and land which makes them ideal to move through dense areas to reach hard to reach places. Another desirable aspect of multirotors is their payload carrying capabilities. Multirotors can carry relatively large weights due to their multiple rotors making them suitable for aerial payload transportation applications.

These desirable capabilities led to the use of multirotors for applications which include aerial photography, package delivery, agriculture, mining, emergency and disaster relief services and security. Multirotors are used for package delivery to quickly transport packages in cities. Agricultural applications include crop monitoring and inspection as well as irrigation. Multirotors are used in mining applications, power line inspections and mapping. Emergency applications include disaster monitoring, the delivery of emergency supplies to hard to reach areas as well as the transportation of water buckets to aid in firefighting. Lastly, multirotors are used in security applications for surveillance, which more recently was used for anti-poaching systems.

Throughout these applications the multirotors usually carry some sort of payload, either sensors or packages. These payloads affect the flight dynamics of the multirotor. The properties of sensor payloads are usually known and constant. Therefore, control systems can be designed for these specific payloads. However, package delivery services transport a variety of payloads which are unknown before flight and can have a significant effect on the flight dynamics of the multirotor.

To summarize, multirotors offer an economically viable solution for a variety of applications which include the transportation of payloads. However, the addition of a payload greatly affects the flight dynamics of the multirotor which may result in dangerous and unstable flight. This identifies the need for adaptive and robust flight control systems for multirotors carrying unknown payloads. This project aims to develop a solution that solves this problem.

1.2. Project Definition and Objectives

The aim of this project is to design, implement and verify a flight control system for a multirotor transporting an unknown suspended payload. The payload has unknown parameters which include its mass and the cable length and a direct measurement of the suspended payload swing angle states is not available.

The swinging motion of the suspended payload affects the flight dynamics of the multirotor and may result in unstable flight. This project aims to design a flight control system that ensures stable flight by damping the oscillations of the suspended payload. To this end, the research objectives are:

1. Investigate the current solutions found in literature of multirotor payload transportation systems.
2. Derive a mathematical model of a multirotor with a suspended payload.
3. Identify the hardware and software components required for simulation and testing.
4. Design and verify in simulation a control system for a multirotor without a suspended payload. This will serve as the base control system for the multirotor and suspended payload system.
5. Identify and implement a system identification strategy to estimate the unknown payload parameters.
6. Design and implement a vision-based state estimator to estimate the suspended payload swing angle states for full-state feedback control.
7. Design, simulate and verify an adaptive control strategy to ensure stable flight by suppressing the swinging motion of an unknown suspended payload.

1.3. Thesis Outline

Chapter 1 provided the background and motivation for the research, the project objectives were stated and an outline of the thesis is given.

Chapter 2 contains a literature study on the control strategies developed for the aerial payload transportation problem focussing on multirotor unmanned aerial vehicles.

Chapter 3 contains the derivation of the mathematical model of a multirotor with a suspended payload which is used for controller design and simulation.

Chapter 4 presents the system overview. An overview of the hardware and software components that form part of the project is given.

Chapter 5 presents an overview of the adopted control system architecture for a multirotor, followed by the design and simulation of the flight control system for a multirotor without a suspended payload.

Chapter 6 presents a system identification procedure to estimate the parameters of an unknown suspended payload. An overview of a vision-based state estimation strategy to estimate the states of a suspended payload is given. These estimation algorithms are then verified in simulation.

Chapter 7 describes the design of a control strategy to damp the oscillations of a suspended payload. Simulation results of the proposed control strategy are presented.

Chapter 8 gives a brief overview of the implementation and simulation of the proposed solution. Hardware-in-the-loop simulations are performed to demonstrate the performance of the proposed control strategy and determine its practical feasibility. The simulation results are presented and discussed.

Chapter 9 presents the conclusions drawn and recommendations for future work.

Chapter 2

Literature Study

This chapter discusses the literature regarding multirotor unmanned aerial vehicle (UAV) payload transportation systems. An overview of the different payload configurations and the respective control solutions found in literature are given. This is followed by a discussion on the different control techniques used for the transportation of unknown payloads. The different methods used for the estimation of payload states are also discussed.

2.1. Multirotors with Payloads

In the earliest literature of aerial payload transportation systems, helicopters were used for payload transportation. However, more recently, the use of multirotors for payload transportation became more popular as multirotors can carry greater relative payload masses as they produce more thrust. There are two basic payload configurations considered for transportation in literature which include grasped payloads [1] and suspended payloads [2]. These configurations have certain advantages and drawbacks which need to be considered for transportation applications. Several control techniques can be found in literature for the safe transportation of these types of payloads.

2.1.1. Grasped Payloads

Grasped payloads are payloads that are rigidly attached to a multirotor using grippers or any type of fixed joint. The use of grippers are ideal for autonomous payload pick up and drop applications [1]. However, grippers limit the sizes and shapes of the payloads that can be transported. Figure 2.1 shows a multirotor with a grasped payload. Grasped payloads changes the dynamics of multirotors by shifting the center of mass (CoM) of the system and increasing its mass moment of inertia. This results in a slower attitude response which may reduce the control system robustness in the presence of disturbances. The increased weight of the system due to the payload also affects the vertical dynamics of the multirotor.

Melinger et al. [1] proposed the use of least-squares methods to estimate the mass of an unknown grasped payload and the resulting shift in the center of mass during pick up and drop applications using quadrotors. An adaptive controller was designed to take these estimates into account to improve the vertical and horizontal tracking capability for different payloads. Experimental results indicated an improvement in tracking performance.

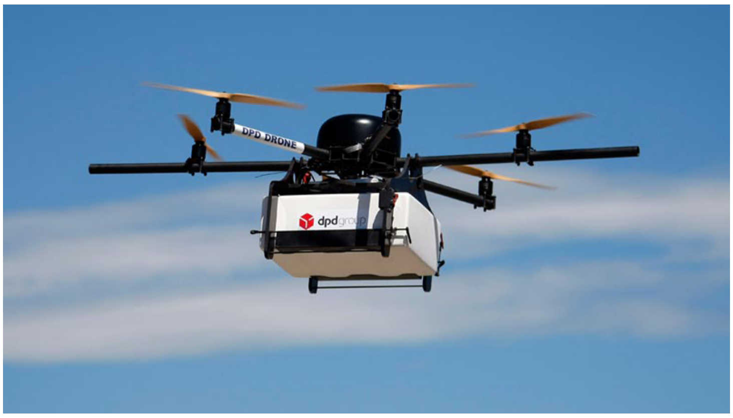


Figure 2.1: Multirotor with a grasped payload [3].

A fractional order sliding mode controller (SMC) based on the back-stepping method with an adaptive correcting coefficient to compensate for an unknown payload was implemented by Vahdanipour and Khodabandeh [4]. The aim was to reduce the effect of the unknown payload on the position tracking performance of a quadrotor. Simulation results indicated the robustness of the controller in the presence of disturbances.

Emran et al. [5] implemented a model reference adaptive control (MRAC) controller to assist the onboard proportional integral (PI) altitude controller of a quadrotor during payload pick up and drop applications. The MRAC controller can deal online with system changes and force the system to perform as desired. Simulation results indicated that the adaptive controller has superior tracking performance compared to a benchmark proportional integral derivative (PID) controller.

2.1.2. Suspended Payloads

To avoid the drawbacks of grasped payloads such as the slower system response and the limited shapes and sizes of payloads that can be transported, payloads can be attached to multirotors using cables or ropes as shown in Figure 2.2. Assuming that the cable attachment point is near the multirotor CoM, the suspended payload does not slow the attitude response of the multirotor. The shape and size of the payload are also not limited as any arbitrary payload can be attached to a cable.

However, suspended payloads significantly alters the flight dynamics of a multirotor as these payloads are free to swing. This introduces new challenges to stabilize the multirotor and payload system during flight. To this end, two methods have been investigated to stabilize and minimize suspended payload oscillations. This includes trajectory generation methods for swing free motion and active swing damping control methods. In some cases, a combination of both methods are used.

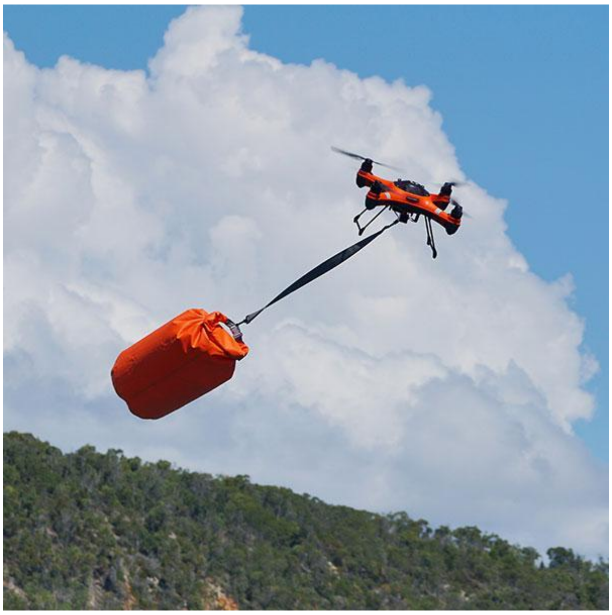


Figure 2.2: Multirotor with a suspended payload [6].

Trajectory Generation for Swing Free Motion

In order to achieve swing free motion, trajectories can be generated that specifically reduce the oscillations of a suspended payload. This reduces the effect of the suspended payload on the multirotor flight dynamics.

Some of the earliest work to generate swing free trajectories were focussed on overhead gantry cranes [7]. An open-loop control technique called input shaping was used to reduce suspended payload oscillations. Due to the similar pendulus motion of the suspended loads attached to gantry cranes and multirotors with suspended payloads, input shaping techniques can be utilized for multirotor and suspended payload systems. Input shaping involves the convolution of a reference signal with a sequence of impulses to prevent the excitation of a vibration mode.

Ichikawa et al. [8] implemented several input shaping techniques to the velocity control of a quadrotor and suspended payload system in simulation in order to compare the payload oscillation suppression capabilities of the different input shapers. Simulation results indicated that the considered input shapers were able to suppress payload oscillations when model uncertainties were introduced. It was also observed that there is a trade-off between robustness and rise time as the more robust input shapers, which include the Extra Insensitive (EI) and 2-hump EI shapers, increased the system rise time.

Potter et al. [9] and Bisgaard et al. [10] used input shaping to generate swing free trajectories for helicopters with suspended payloads. Simulation and practical results indicated that the input shaping greatly reduced residual swing of the suspended payload. Similar results were obtained in simulation for multirotors with suspended payloads [11,12]. Input shaping techniques improved the trajectory tracking performance.

A different approach was proposed by Palunko et al. [13]. Dynamic programming was used to generate optimal trajectories for swing free maneuvers of a quadrotor with a suspended payload. The approach also enabled aggressive maneuvers while maintaining swing free motion. Practical results indicated that the proposed approach was able to attenuate the suspended payload swing angles.

Both input shaping and dynamic programming are open-loop techniques and therefore sensitive to model uncertainties and unable to reject external disturbances. Xian et al. [14], therefore, proposed an online trajectory planning method which uses feedback of the suspended payload swing angles to generate swing free trajectories. The trajectory planning method consists of a positioning component and an anti-swing component for simultaneous position tracking and payload swing suppression. The suspended payload swing angles are measured using an external motion capture system. Practical results are presented and validates the performance of the proposed approach.

A reinforcement learning approach for trajectory generation with minimal payload oscillations is proposed by Faust et al. [15]. An approximate value iteration (AVI) reinforcement learning algorithm is used to train the system for a particular suspended payload. Practical results indicated that the approach was able to produce trajectories with minimal payload oscillations and is robust against noise and unmodeled system dynamics.

Active Swing Damping Control

Another approach taken to ensure stable flight with minimal payload swing oscillations is the use of active swing damping controllers. Swing damping controllers are closed-loop controllers with the objective to add damping to the multirotor and suspended payload system usually by directly controlling the suspended payload states. Several closed-loop control strategies have been investigated and implemented for active swing damping control.

Kusznir and Smoczek [16] proposed a sliding mode controller for the position tracking and oscillation damping of a quadrotor with a suspended payload. An adaptive pole placement method was used to tune the sliding surface parameters. Only motion in the XZ-plane was considered in simulation and results indicated that the controller was able to damp the suspended payload oscillations even when parameter uncertainties were introduced.

A linear quadratic regulator (LQR) was implemented for the full-state feedback control of a quadrotor and suspended payload [17]. The optimal controller was designed around hover for quadrotor position tracking and payload swing damping. Simulation results indicated that the proposed controller was able to follow relatively aggressive maneuvers while suppressing the suspended payload oscillations.

Nicotra et al. [18] proposed a nested saturation control law for simultaneous multirotor reference tracking and payload swing angle damping which was verified in simulation.

De Almeida and Raffo [19] proposed a three level cascaded input-output feedback linearization control strategy for a tilt-rotor UAV with a suspended payload where simulation results indicated a reduction in payload oscillations while a deterioration in the path tracking performance was observed. This was followed by an improved control strategy [20] where the structure was changed to a two level cascaded input-output feedback linearization control strategy using linear mixed $\mathscr{H}_2/\mathscr{H}_\infty$ controllers with pole placement constraints. An additional control term is introduced to reduce payload swing. The improved strategy was verified in simulation and showed improved trajectory tracking and swing damping performance in the presence of sensor noise, external disturbances and model uncertainties.

A fuzzy-based anti-swing controller that mimics the performance of a time-delayed feedback controller was proposed by Omar [21] for helicopters with suspended payloads. Particle swarm optimization (PSO) was used to optimally tune the distribution of the membership functions and simulation results showed that the controller is able to damp payload oscillations.

An interconnected and damping assignment-passivity based controller (IDA-PBC) for a quadrotor with a suspended payload was proposed by Guerrero et al. [22]. The control technique differs from the abovementioned control techniques as it is designed to be independent of the payload swing angles. Practical results showed that the controller was able to significantly damp the suspended payload oscillations.

2.2. Multirotors with Unknown Suspended Payloads

The literature presented in the previous section mainly focussed on the design of control strategies for aerial payload transportation, and not the practical application thereof. The system parameters and states are assumed to be known. However, considering the general suspended payload transportation problem, multirotors should be able to transport a variety of payloads in the presence of sensor noise and external disturbances. This introduces the need for adaptive or robust control of multirotors with unknown suspended payloads. Different approaches can be taken to find solutions.

2.2.1. Adaptive Control of Multirotors with Unknown Suspended Payloads

Adaptive control techniques can be implemented to adapt to a specific suspended payload by using estimates of its unknown parameters. The parameters that mainly contribute to the dynamics of the suspended payload are its mass and the cable length.

Dai et al. [23] proposed a fixed-gain geometric nonlinear proportional derivative (PD) controller with a retrospective cost adaptive controller (RCAC) to compensate for changes in payload mass where the PD controller is designed to achieve the desired performance for a nominal payload mass. The RCAC controller compensates for the different payload masses by re-optimizing the nominal controller based on past data. The tracking performance of the proposed control strategy was verified in simulation and showed an improvement compared to the fixed-gain PD controller with an integral term.

Yang and Xian [24] considered the case where the payload cable length is unknown and proposed an energy-based nonlinear adaptive controller for the position tracking of a quadrotor and swing damping of a suspended payload. Good results were obtained in practical tests.

A different approach was taken by Bisgaardt et al. [25]. A combined feedforward and feedback control scheme was proposed for the transportation of a suspended payload with an unknown cable length which requires the estimation of the cable length. A downward facing camera attached to the bottom of a helicopter was used to estimate the payload swing angles which in turn was used to estimate the unknown cable length. The estimated cable length was then used for the online redesign of the swing damping control scheme. Practical results showed that the controller was able to significantly damp payload oscillations.

Feng et al. [26] focussed on the problem where both the payload mass and cable length is unknown. The authors proposed an adaptive control method for the altitude and attitude control of a quadrotor with an unknown suspended payload. The proposed controller is designed to ensure system stability and maintain good tracking performance when under the influence of the external force and torque caused by the swinging payload. Simulation results indicated that the proposed controller was able to maintain system stability for different payload masses and cable lengths.

2.2.2. Robust Control of Multirotors with Unknown Suspended Payloads

The use of robust control techniques is a popular approach as it ensures stability for a range of parameter uncertainties. Zhou et al. [27] approached the problem by treating the suspended payload as an external disturbance and designed control systems that have good disturbance rejection. The authors compared the trajectory tracking performance of PD control, sliding mode control and model predictive control (MPC) for a quadrotor with a suspended payload which has an unknown mass.

Lee et al. [28] also considered the problem of an unknown payload mass, but approached the problem differently. A parameter robust linear quadratic Gaussian (PRLQG) controller for the trajectory tracking of a suspended payload attached to a multirotor was proposed. The proposed control method proved to be robust against varying payload masses. The PRLQG controller was then implemented for cooperative payload transportation using multiple multirotors and showed similar results in simulation.

More recently, Taylor [29] proposed a modified $\mathscr{H}_\infty$ extended loop shaping controller for the velocity reference tracking and payload swing damping of a quadrotor with an unknown suspended payload. The controller utilises $\mathscr{H}_\infty$ optimization and the v-gap metric. Simulation results showed that the controller provides robust stability for a range of unknown payload masses and cable lengths.

2.3. Suspended Payload State Estimation

Most of the control strategies discussed assume that the system states are available or can be measured using sophisticated motion capture systems [14], [24]. However, this is not realizable in real-world applications. Additional onboard sensors are required to measure the swing angle states of the suspended payload for feedback control. Alternatively, payload states can be estimated using model-based state estimation techniques which does not require additional sensors.

In [11] and [30] two magnetic encoders were used to measure the swing angles of a suspended payload. In [31] an inertial measurement unit (IMU) is attached to a suspended payload with a load cell attached to the cable to measure the disturbance force caused by the swinging payload. A disturbance observer (DOB) derived disturbance force estimator is then implemented to estimate the payload swing angles.

A more popular approach to determine the payload states is the use of onboard cameras due to the relatively low cost of cameras as well as the fact that no sensors are required to be placed on the payload. In [25], [32] and [33] downward facing cameras were attached to the bottom of aerial vehicles to measure the relative position of a suspended payload. Visual markers were placed on top of the payloads and robust visual marker detection algorithms were implemented to detect the payload position. Due to the limited field of view (FOV) of cameras, aggressive maneuvers can result in the suspended payload swinging outside the camera FOV. Therefore, Tang et al. [34] used a camera with a fisheye lense.

More recently, de Angelis [35] proposed the use of a Fading Gaussian Deterministic (FGD) filter to estimate the suspended payload states using only the embedded IMU on the multirotor flight controlling unit (FCU).

2.4. Dynamic Object Detection for Vision-Based State Estimation of a Suspended Payload

The use of an onboard camera for the state estimation of a suspended payload attached to a multirotor is an attractive solution for real-world aerial transport applications as it does not require extra sensors to be attached to the payload. Therefore, a variety of payloads can be transported. However, from the literature discussed in Section 2.3, this is not entirely the case as visual markers are placed on the suspended payloads which require specific marker detection algorithms for detection.

In order to obtain a general solution for the vision-based state estimation of a suspended payload, which does not require visual markers, dynamic object detection techniques are investigated. Much literature is available on dynamic object detection techniques. However, only the techniques considered will briefly be discussed.

Frame Differencing

Frame differencing [36] is a motion-based moving object detection technique that separates moving objects from a static background by the pixel-wise subtraction of two consecutive image frames. It is assumed that the camera is stationary and that any differences between frames form part of moving objects. Frame differencing is a simple technique to use, however, it is sensitive to noise and illumination changes and can not be used in its basic form. Different techniques can be applied to make frame differencing more robust against noise.

Furthermore, to use frame differencing for moving cameras, techniques can be applied to compensate for the camera motion. This is applicable for cameras attached to multirotors. By estimating the dominant inter-frame background motion, an indication of the relative camera motion can be found. This can then be compensated for so that only the motion of moving objects are visible [37, 38].

Optical Flow

Another technique for motion-based object detection is optical flow [39]. Optical flow describes the apparent movement or flow of pixel intensity values over time. It is assumed that the intensity changes are due to motion only. Areas with similar optical flow can then be grouped together. Optical flow has a high sensitivity to noise and can not deal with occlusions. The use of optical flow in real-time applications were limited as it is computationally expensive. However, recent advances in technology enabled its use in real-time applications.

2.5. Summary

This chapter discussed the solutions found in literature on the transportion of payloads using multirotors. The different types of payload configurations were discussed and the use of suspended payloads is found to be the more popular configuration. Two approaches are implemented to ensure stable flight and reduce suspended payload swing oscillations which include trajectory generation methods for swing free motion and active swing damping control.

At the aim of real-world applications, adaptive and robust control solutions were explored which can adapt or are robust to parameter changes. Methods for the state estimation of suspended payloads for feedback control were explored. Vision-based state estimation is a popular method as it requires no extra sensors to be attached to payloads. However, vision-based state estimation methods mostly rely on visual markers. This motivated the investigation of possible object detection techniques that can be used for vision-based state estimation of a suspended payload without visual markers.

Chapter 3

Modelling

In this chapter, the mathematical and graphical models of a quadrotor with a suspended payload are derived. The derived models are used to simulate the system and for controller design and verification in subsequent chapters.

The chapter starts with a description of the different coordinate systems, after which the six degrees of freedom equations of motion of a quadrotor are derived. An overview of quaternions and Euler angles are given. The forces and moments acting on the quadrotor are discussed. Lastly the mathematical and graphical models of a quadrotor and suspended payload are derived and compared.

3.1. Coordinate Systems

Three coordinate systems are used for describing the system, as shown in Figure 3.1. The first coordinate frame is the inertial frame, $\mathcal{I}$, which is described by a North-East-Down (NED) axis system. This system assumes a flat, non-rotating earth. This assumption is valid as the quadrotor only travels short distances. The origin of the inertial frame is usually chosen to coincide with the takeoff position of the quadrotor. North is aligned with the x-axis, East with the y-axis and Down with the z-axis.

The body frame, $\mathcal{B}$, is fixed to the quadrotor where its origin is at the quadrotor's CoM. The x-axis points in the forward direction of the quadrotor, the y-axis to the right and the z-axis downwards.

Lastly, the camera frame, $\mathcal{C}$, is fixed to the onboard camera, which is located at the bottom of the quadrotor and points downwards. The origin of the camera frame coincides with the center of the camera and has an offset from the body frame. As the camera is fixed to the quadrotor, the camera frame's rotation follows that of the body frame.

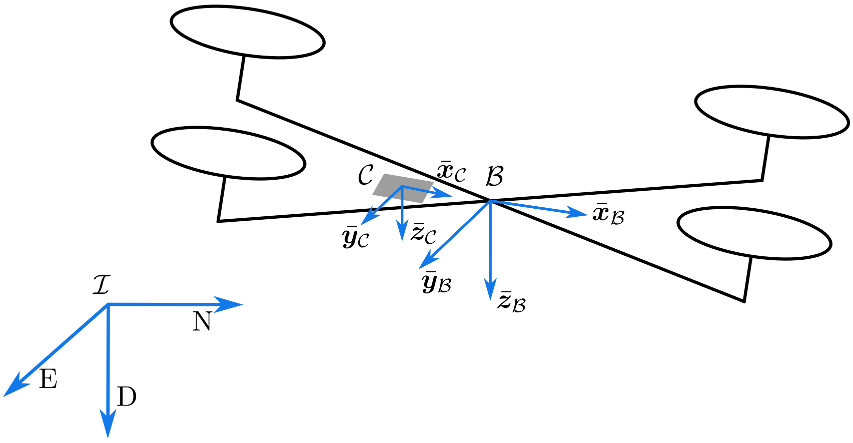

$\mathcal{C}$
$\bar{\boldsymbol{x}}_{\mathcal{C}}$
$\mathcal{B}$
$\bar{\boldsymbol{x}}_{\mathcal{B}}$
$\bar{\boldsymbol{y}}_{\mathcal{C}}$
$\bar{\boldsymbol{z}}_{\mathcal{C}}$
$\bar{\boldsymbol{y}}_{\mathcal{B}}$
$\bar{\boldsymbol{z}}_{\mathcal{B}}$
$\mathcal{I}$
N
E
D

Figure 3.1: Coordinate frames.

3.2. Quadrotor Modelling

A quadrotor can be well modelled as a six degrees of freedom (6DOF) rigid body. The six degrees of freedom include the three translational and three rotational degrees of freedom. The model is derived using quaternions instead of Euler angles as in [40].

3.2.1. Kinetics

Kinetics refers to the relationship between the forces and moments acting on a body and its resulting motion. Using Newton's second law, the equations of motion are determined as

$$\boldsymbol{F}_{\boldsymbol{\mathcal{B}}} = m_q \dot{\boldsymbol{V}}_{\boldsymbol{\mathcal{B}}} + \boldsymbol{\Omega}_{\boldsymbol{\mathcal{B}}} \times m_q \boldsymbol{V}_{\boldsymbol{\mathcal{B}}} \text{ and} \tag{3.1}$$

$$\boldsymbol{M}_{\boldsymbol{\mathcal{B}}} = \boldsymbol{I}_{\boldsymbol{q}} \dot{\boldsymbol{\Omega}}_{\boldsymbol{\mathcal{B}}} + \boldsymbol{\Omega}_{\boldsymbol{\mathcal{B}}} \times \boldsymbol{I}_{\boldsymbol{q}} \boldsymbol{V}_{\boldsymbol{\mathcal{B}}}, \tag{3.2}$$

where

$$\boldsymbol{F}_{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \text{ and} \tag{3.3}$$

$$\boldsymbol{M}_{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} L & M & N \end{bmatrix}^T \tag{3.4}$$

are the forces and moments, respectively, acting on the quadrotor, with respect to the body frame. The linear and angular velocity of the quadrotor are given by

$$\boldsymbol{V}_{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} U & V & W \end{bmatrix}^T \text{ and} \tag{3.5}$$

$$\boldsymbol{\Omega}_{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} P & Q & R \end{bmatrix}^T. \tag{3.6}$$

The quadrotor mass is denoted by m_q and its inertia tensor is given by

$$\boldsymbol{I_q} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \approx \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} . \tag{3.7}$$

The inertia tensor can be simplified as shown in Equation (3.7) due to the symmetry of the quadrotor in the XZ- and YZ-plane.

3.2.2. Kinematics

Kinematics refers to the motion of a body, and relates the linear velocity and angular velocity of the body to its position and attitude. A transformation matrix can be defined in terms of the attitude of a body to relate between coordinate frames such as the inertial frame and body frame. Attitude can be defined in terms of Euler angles or quaternions of which both are used in this project. A brief overview of both is given after which the attitude and position dynamics are discussed.

Euler Angles

The attitude of a body in a coordinate frame can be parameterized using Euler angles. The three Euler angles are applied in a predefined order to describe the rotation from the body frame to inertial frame. For this project the Euler 3-2-1 parameterization is used. The order in which the Euler angles are applied, as shown in Figure 3.2, is:

1. Yaw the body frame through the heading angle Ψ.
2. Pitch the resulting body frame through the pitch angle Θ.
3. Roll the resulting body frame through the roll angle Φ.

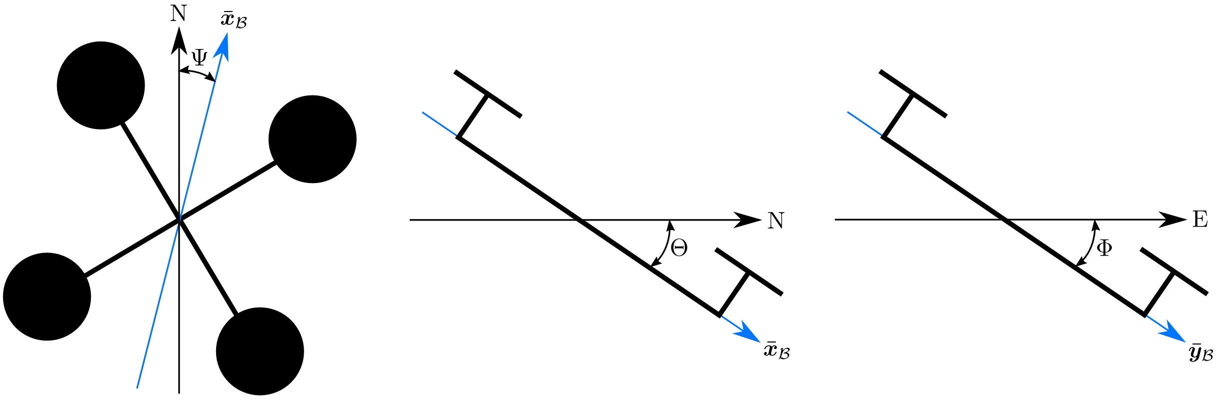


Figure 3.2: Euler attitude angles.

Quaternions

Quaternions are vectors consisting of four elements that can be used to describe rotations and orientations. The attitude of a body in some coordinate frame can be described by an axis of rotation and an angle of rotation about that axis. Consider the rotation of θ about unit vector $\bar{\boldsymbol{r}} = [\,r_x \quad x_y \quad r_z]^T$ in Figure 3.3:

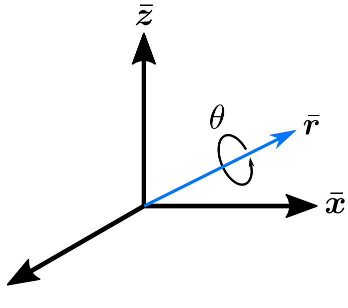


Figure 3.3: Axis angle representation.

The unit quaternion, $\bar{\boldsymbol{q}}$, is defined as

$$\bar{\boldsymbol{q}} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ r_x \sin\left(\frac{\theta}{2}\right) \\ r_y \sin\left(\frac{\theta}{2}\right) \\ r_z \sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \tag{3.8}$$

where q_0 is the magnitude and $\boldsymbol{q}_v = [\,r_x \quad x_y \quad r_z]^T$ is the vector part of the quaternion.

Attitude Dynamics

It is necessary to relate the quaternions to the other kinematic states of the quadrotor. The rate of change of the quaternions can be related to the body angular velocity of the quadrotor, defined in Equation (3.2), by

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ P \\ Q \\ R \end{bmatrix}. \tag{3.9}$$

The derivation of Equation (3.9) can be found in [32].

Position Dynamics

The attitude can now be used to transform the linear velocity of the quadrotor into the inertial frame

$$\boldsymbol{V}_{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}^{-1}} \boldsymbol{V}_{\mathcal{B}}, \tag{3.10}$$

by using the transformation matrix

$$\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_0q_2+q_1q_3) \\ 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2-q_3^2 \end{bmatrix}. \tag{3.11}$$

$\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}$ is the transformation matrix that describes the rotation of the quadrotor from the inertial frame to the body frame. The derivation of Equation (3.11) can be found in [32].

3.3. Forces and Moments

With the equations of motion formulated, the forces and moments acting on the quadrotor need to be determined. The forces and moments acting in on a quadrotor includes gravity, aerodynamics, the thrust generated by the actuators as well as the suspended payload and are given by

$$\boldsymbol{F}_{\mathcal{B}} = \boldsymbol{F}_{\mathcal{B}}^{G} + \boldsymbol{F}_{\mathcal{B}}^{A} + \boldsymbol{F}_{\mathcal{B}}^{T} + \boldsymbol{F}_{\mathcal{B}}^{P} \text{ and} \tag{3.12}$$

$$\boldsymbol{M}_{\mathcal{B}} = \boldsymbol{M}_{\mathcal{B}}^{G} + \boldsymbol{M}_{\mathcal{B}}^{A} + \boldsymbol{M}_{\mathcal{B}}^{T} + \boldsymbol{M}_{\mathcal{B}}^{P}, \tag{3.13}$$

where subscripts G, A, T and P refers to gravity, aerodynamics, thrust and the payload, respectively. The forces and moments due to the swinging payload are discussed in Section 3.4.3.

3.3.1. Gravity

The gravitational force acts in the Down direction in the inertial frame and has a magnitude that is related to the mass of the quadrotor in the Down direction. The transformation matrix, $\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}$, is required to transform the force into the body frame. The forces and moments due to gravity can be written as

$$\boldsymbol{F}_{\mathcal{B}}^{G} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \begin{bmatrix} 0 \\ 0 \\ m_q g \end{bmatrix} \text{ and} \tag{3.14}$$

$$\boldsymbol{M}_{\mathcal{B}}^{G} = \boldsymbol{0}. \tag{3.15}$$

3.3.2. Aerodynamics

The aerodynamic drag forces acting on the quadrotor are modelled as drag forces acting on an object that is moving through a fluid and is given by

$$\boldsymbol{F}_{\mathcal{B}}^{A} = \frac{1}{2}\rho \boldsymbol{V}_{\mathcal{B}_w} |\boldsymbol{V}_{\mathcal{B}_w}| \boldsymbol{C}_{D_q}, \tag{3.16}$$

where ρ is the air density, chosen to be at 15°C, $\boldsymbol{V}_{\mathcal{B}_w}$ is the relative velocity of the quadrotor and wind in the body frame and $\boldsymbol{C}_{D_q}$ is the lumped drag coefficients and reference area of the quadrotor, which are equal in each axis (refer to Section 4.1). The relative velocity, $\boldsymbol{V}_{\mathcal{B}_w}$, is calculated as

$$\boldsymbol{V}_{\mathcal{B}_w} = -\boldsymbol{V}_{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{V}_w, \tag{3.17}$$

where $\boldsymbol{V}_w$ is the wind velocity in the inertial frame.

3.3.3. Thrust

A first order lag model is used to describe the thrust forces generated by the actuators of the quadrotor. The lag dynamics for each actuator is given by

$$\dot{T}_i = \frac{-T_i + T_{iR}}{\tau}, \tag{3.18}$$

where T_i is the thrust produced by the ith actuator, T_{iR} is the reference thrust and τ is the actuator time constant.

Virtual actuators are defined to follow that of a traditional fixed-wing aircraft. This enables the use of thrust, aileron, elevator and rudder commands to produce thrust, rolling moments, pitching moments and yawing moments, respectively. For a quadrotor in the cross configuration, the virtual actuators are defined as

$$\delta_T = T_1 + T_2 + T_3 + T_4, \tag{3.19}$$

$$\delta_A = \frac{1}{\sqrt{2}}(-T_1 + T_2 + T_3 - T_4), \tag{3.20}$$

$$\delta_E = \frac{1}{\sqrt{2}}(T_1 - T_2 + T_3 - T_4) \text{ and} \tag{3.21}$$

$$\delta_R = T_1 + T_2 - T_3 - T_4. \tag{3.22}$$

A mixing matrix can be defined for the transformation between the actual and virtual actuators. The mixing matrix is given by

$$\begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}. \tag{3.23}$$

The forces and moments produced by the actuators are

$$\boldsymbol{F}_{\mathcal{B}}^{T} = \begin{bmatrix} 0 \\ 0 \\ \delta_T \end{bmatrix} \text{ and} \tag{3.24}$$

$$\boldsymbol{M}_{\mathcal{B}}^{T} = \begin{bmatrix} d\delta_A \\ d\delta_E \\ \frac{r_D}{R_{LD}}\delta_R \end{bmatrix} \tag{3.25}$$

where d is the distance from the actuator to the quadrotor CoM, r_D is the chord length where the rotor drag forces are exerted and R_{LD} is the lift to drag ratio of the rotor.

3.4. Quadrotor with Suspended Payload Modelling

The quadrotor and payload system is modelled as a system of two rigid bodies that is connected with a link. The system dynamics follow that of a moving pendulum model. The following assumptions are made regarding the suspended payload:

- The payload is treated as a point mass.
- The link is massless and rigid.
- The link is attached to the CoM of the quadrotor.

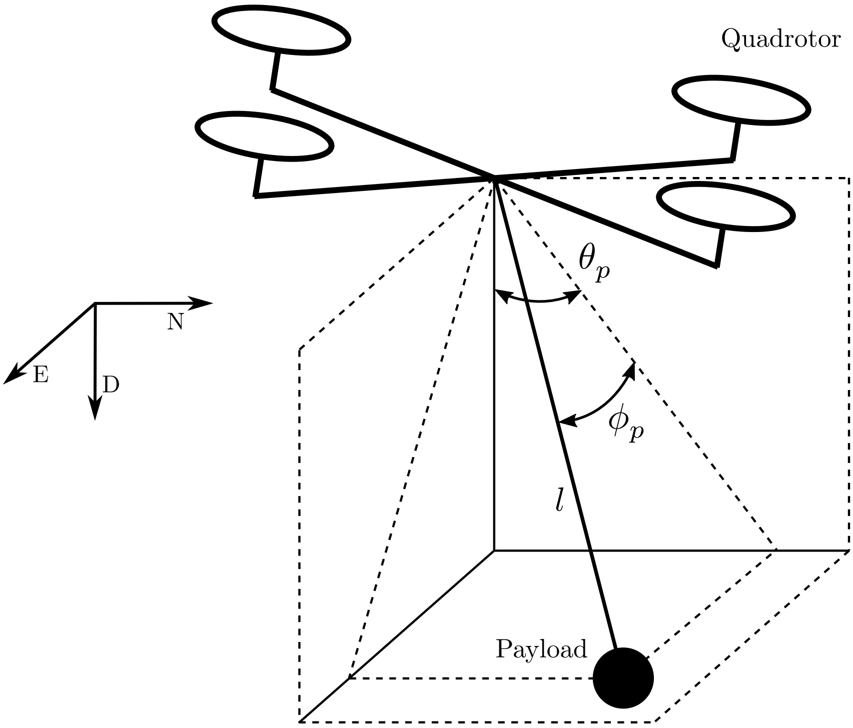


Figure 3.4: Quadrotor with suspended payload.

It can be assumed that the cable is rigid during flight as there will be tension present in the cable, except when the payload enters free-fall. However, this is unlikely as aggressive maneuvers will not be performed during flight. The mass of the cable is negligible relative to the payload masses considered (1 kg to 3 kg).

3.4.1. Modelling Approaches

Different approaches can be used to model the quadrotor and payload system. The approaches considered include:

1. Newtonian mechanics
2. Lagrangian mechanics
3. Graphical modelling

Different modelling approaches hold different advantages based on the properties of the system to be modelled. Therefore, an approach should be chosen to simplify the problem at hand. The advantages and disadvantages of the considered approaches are summarized in Table 3.1.

Modelling Approach	**Advantages**	**Disadvantages**
Newtonian mechanics	Provides a simple and intuitive description of the system.	Complexity grows as more constraints and degrees of freedom are present in systems.
Lagrangian mechanics	Do not need to explicitly account for the system constraints. Free to use any set of generalized coordinates.	Does not always work well with non-conservative forces or dissipative forces.
Graphical modelling	Defines a system in terms of links and joints. No need to mathematically derive the equations of motion.	The system becomes a black box. The equations of motion of the system are unknown.

Table 3.1: Comparison of modelling approaches.

Considering the positives and negatives of the different modelling approaches, Lagrangian mechanics is chosen to model the quadrotor and payload system. Lagrangian mechanics simplifies the problem by allowing the use of generalized coordinates. The coordinates are chosen to enforce constraints so that one does not need to explicitly account for these constraints as with Newtonian mechanics.

Graphical modelling is also a very attractive approach due to its simplicity, but lacks the derivation of the equations of motion. However, graphical modelling can be used as a simple and efficient way to validate the derived mathematical models.

3.4.2. Lagrangian Mechanics

The equations of motions of the quadrotor and payload system are now derived using Lagrangian mechanics. The quadrotor and suspended payload is shown in Figure 3.4, where m_q is the mass of the quadrotor, m_p is the mass of the payload and l is the payload cable length. The position of the quadrotor in the inertial frame is given by $\boldsymbol{r}_q = [x_q \; y_q \; z_q]^T$, where x_q, y_q and z_q are the North, East and Down coordinates of the quadrotor, repectively. The position of the suspended payload relative to the quadrotor is defined using two angles, θ_p and ϕ_p. These angles form an Euler 2-1 angle rotation around the East axis and North axis, respectively.

Lagrangian of a Moving Pendulum

The Lagrangian is defined as

$$\mathcal{L} = T - V, \tag{3.26}$$

where T is the total kinetic energy of the system and V is the total potential energy of the system. The generalized equations of motion of the system can then be described by the Euler-Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{p_j}}\right) - \frac{\partial \mathcal{L}}{\partial p_j} = Q_j, \tag{3.27}$$

where p_j is an element in the set of generalized coordinates, $\boldsymbol{p}$, and $\boldsymbol{Q}$ the respective non-conservative forces. For the quadrotor and payload system, $\boldsymbol{p}$ is chosen as

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{r}_q \\ \phi_p \\ \theta_p \end{bmatrix}. \tag{3.28}$$

With the position of the quadrotor in the inertial frame being described by, $\boldsymbol{r}_q$, the next requirement is to describe the position of the payload in the inertial frame. Considering Figure 3.4, the position of the payload relative to the quadrotor can be described in terms of the swing angles θ_p and ϕ_p. Applying the Euler 2-1 rotation, the position of the payload in the inertial frame can be written as

$$\boldsymbol{r}_p = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} x_q + l\cos(\phi_p)\sin(\theta_p) \\ y_q + \sin(\phi_p) \\ z_q + l\cos(\phi_p)\cos(\theta_p) \end{bmatrix}. \tag{3.29}$$

The quadrotor and payload positions can then be differentiated to determined their velocities in the inertial frame. The payload velocity can be written as

$$\dot{\boldsymbol{r}}_p = \begin{bmatrix} \dot{x_p} \\ \dot{y_p} \\ \dot{z_p} \end{bmatrix} = \begin{bmatrix} \dot{x_q} + \dot{\theta}_p l\cos(\phi_p)\cos(\theta_p) - \dot{\phi}_p l\sin(\phi_p)\sin(\theta_p) \\ \dot{y_q} + \dot{\phi}_p l\cos(\phi_p) \\ \dot{z_q} - \dot{\theta}_p l\cos(\phi_p)\sin(\theta_p) - \dot{\phi}_p l\cos(\theta_p)\sin(\phi_p) \end{bmatrix}. \tag{3.30}$$

With the quadrotor and payload positions and velocities defined in the inertial frame, the total kinetic energy, T, and potential energy, V of the system are determined to be

$$T = \frac{1}{2} m_q \|\dot{\boldsymbol{r}}_q\|^2 + \frac{1}{2} m_p \|\dot{\boldsymbol{r}}_p\|^2 \tag{3.31}$$

and

$$V = -m_q g z_q - m_p g z_p. \tag{3.32}$$

Non-Conservative Forces

The effects of non-conservative forces on the quadrotor and payload system should also be considered in the equations of motion. Non-conservative forces are forces that add or remove energy from the system. The non-conservative forces to be considered are shown in Figure 3.5. It includes the friction at the payload attachment point on the quadrotor, the aerodynamic drag forces present on the quadrotor, $\boldsymbol{F}_{D_q}$, and payload, $\boldsymbol{F}_{D_p}$, as well as the thrust generated by the quadrotor, $\boldsymbol{F}_{\mathcal{I}}^T$. The damping coefficient of the attachment point is denoted by c. As discussed in Section 3.3.2, the aerodynamic drag forces are modelled as drag experienced by an object moving through a fluid and is given by Equation (3.16). Aerodynamic drag is experienced by both the quadrotor and payload. These forces act in the opposite direction of movement.

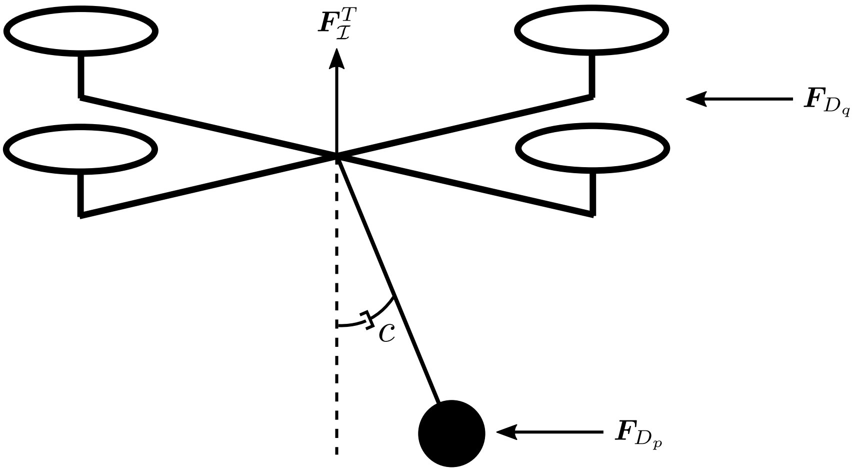


Figure 3.5: Non-conservative forces on the quadrotor and suspended payload.

.

It is assumed that the payload has a spherical shape and that the drag coefficients in all three directions are equal. The drag forces can be written as

$$\boldsymbol{F}_{D_q} = \begin{bmatrix} F^x_{D_q} \\ F^y_{D_q} \\ F^z_{D_q} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho C^x_{D_q}\dot{x_q}^2 \\ \frac{1}{2}\rho C^y_{D_q}\dot{y_q}^2 \\ \frac{1}{2}\rho C^z_{D_q}\dot{z_q}^2 \end{bmatrix} \text{ and} \tag{3.33}$$

$$\boldsymbol{F}_{D_p} = \begin{bmatrix} F^x_{D_p} \\ F^y_{D_p} \\ F^z_{D_p} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho C_{D_p}\dot{x_p}^2 \\ \frac{1}{2}\rho C_{D_p}\dot{y_p}^2 \\ \frac{1}{2}\rho C_{D_p}\dot{z_p}^2 \end{bmatrix}, \tag{3.34}$$

where C_{D_p} is the lumped aerodynamic drag coefficient and reference area of the payload.

The thrust force, $\boldsymbol{F}^T_{\mathcal{I}}$, is obtained by transforming Equation (3.24) from the body frame to the inertial frame:

$$\boldsymbol{F}^T_{\mathcal{I}} = \boldsymbol{R}^{\mathcal{I}^{-1}}_{\mathcal{B}} \boldsymbol{F}^T_{\mathcal{B}}. \tag{3.35}$$

The non-conservative forces for the set of generalized coordinates given in Equation (3.28) is given by

$$\boldsymbol{Q} = \begin{bmatrix} F_{\mathcal{I}_x} - F^x_{D_q} - F^x_{D_p} \\ F_{\mathcal{I}_y} - F^y_{D_q} - F^y_{D_p} \\ F_{\mathcal{I}_z} - F^z_{D_q} - F^z_{D_p} \\ -c\dot{\phi}_p - F^y_{D_p}\cos\phi \\ -c\dot{\theta}_p - F^x_{D_p}\cos\theta \end{bmatrix}. \tag{3.36}$$

With the non-conservative forces defined, the equations of motion of the quadrotor and payload system can now be determined.

Quadrotor and Suspended Payload Equations of Motion

The equations of motion of the quadrotor and payload system are obtained by solving the Euler-Lagrange equation using the Symbolic Toolbox of MATLAB. The resulting equations of motion are verified against a graphical model discussed in Section 3.4.4.

3.4.3. Suspended Payload Forces and Moments

By following the same approach as described in Section 3.4.2, the forces exerted on the quadrotor due to the swinging payload are determined. To determine these forces, the same set of generalised coordinates as in Equation (3.28) are used. However, only the potential energy and non-conservative forces on the payload are considered as shown below:

$$V = -m_p g z_p, \tag{3.37}$$

$$\boldsymbol{Q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -c\dot{\phi}_p - F^y_{D_p}\cos\phi \\ -c\dot{\theta}_p - F^x_{D_p}\cos\theta \end{bmatrix}. \tag{3.38}$$

Using the Symbolic Toolbox of MATLAB to solve the Euler-Lagrange equation, the acceleration of the quadrotor due to the payload is calculated. The inertial forces caused by the swinging payload are then determined by using Newton's second law of motion and are given by

$$\boldsymbol{F}^P_{\mathcal{I}} = m_q \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \\ \ddot{z}_q \end{bmatrix}. \tag{3.39}$$

The equivalent forces in the body frame are then calculated by

$$\boldsymbol{F}^P_{\mathcal{B}} = \boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}} \boldsymbol{F}^P_{\mathcal{I}}. \tag{3.40}$$

3.4.4. Graphical Modelling

Graphical modelling is a simple way to model physical systems in simulation environments. Such simulation environments use physics engines to simulate the physics of the system. Models are composed of different components, such as links and joints, for which physical properties can be specified. For example, the inertia and dimensions can be specified for links while the dynamic and kinematic properties can be specified for joints. The properties given to links and joints tell the physics engine how each component interact with each other and the environment. These components are then combined to obtain the models of systems. Figure 3.6 represents the graphical model of the quadrotor and suspended payload system. The model contains three links that is connected by two joints. The quadrotor is modelled as a 6DOF link. The connection between the quadrotor and the payload cable can be represented by a universal joint which has two rotational degrees of freedom. The two degrees of freedom represent the swing angles, θ_p and ϕ_p. The payload and cable connection is represented with a fixed joint as the payload is fixed to the end of the cable.

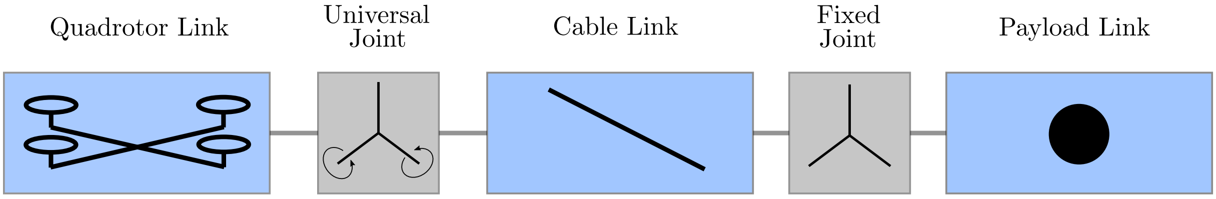


Figure 3.6: Graphical model schematic.

3.4.5. Model Verification

The equations of motion of the moving pendulum system derived in Section 3.4.2 are verified by comparing it to the graphical model described in Section 3.4.4.

The natural responses of both models are simulated and compared. The simulation is initialized with payload swing angles of $\phi_p = 30°$ and $\theta_p = 45°$ and motion in the z-direction is constrained for the quadrotor point mass. A quadrotor with a mass of 4.5 kg and a suspended payload with a mass of 2 kg and cable length of 1 m is used for the simulation.

The resulting quadrotor motion and payload swing angle responses are compared in Figures 3.7 to 3.10. The model derived using Lagrangian mechanics follows the graphical model well. Therefore, it can be concluded that the mathematical model is correct and can be used to aid in controller design, whereas the graphical model can be used for the verification of the designed controllers.

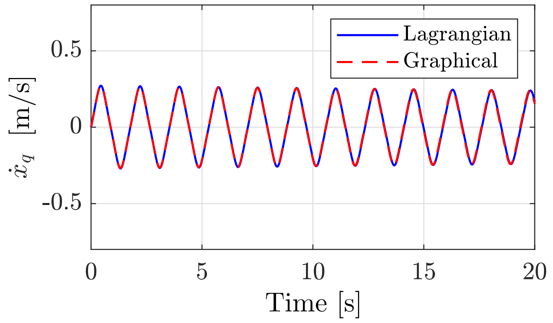


Figure 3.7: The quadrotor velocity response in the x-direction.

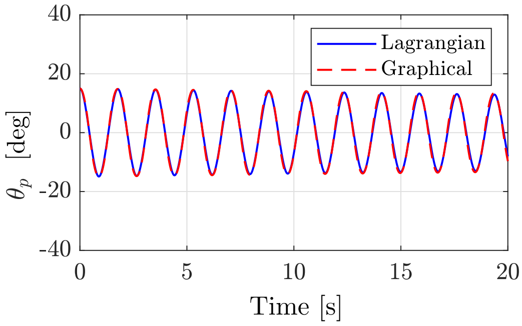


Figure 3.8: The θ_p payload swing angle response.

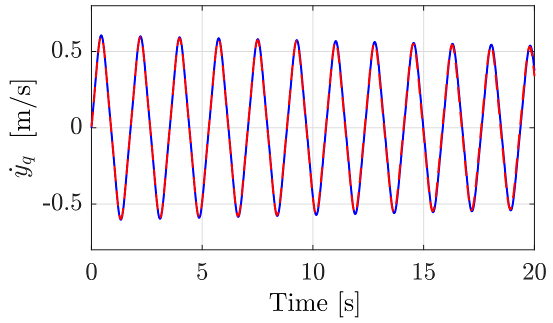


Figure 3.9: The quadrotor velocity response in the y-direction.

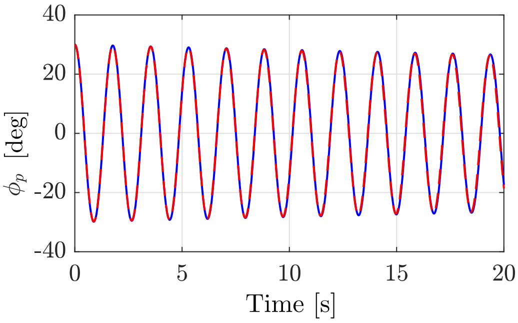


Figure 3.10: The ϕ_p payload swing angle response.

3.5. Summary

In this chapter, the mathematical model of a quadrotor with a suspended payload was derived and the forces and moments for the system were discussed. A block diagram of the model is shown in Figure 3.11. A graphical model of a quadrotor and payload system

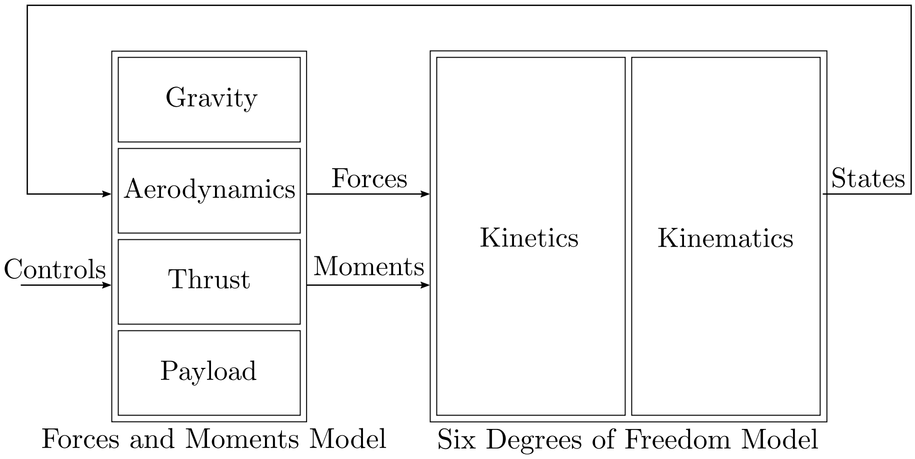


Figure 3.11: Block diagram of the quadrotor and payload model.

was introduced to verify the derived mathematical model. The derived models can be used to simulate similar systems with different parameters and aid in controller design.

Chapter 4

System Overview

This chapter gives an overview of all the components and tools used in this project. The hardware components as well as the flight control toolchain is discussed. An overview of the simulation environments is also given.

4.1. Hardware

The hardware configuration on the quadrotor is shown in Figure 4.1. The flight controller is directly connected to the sensors which provide measurements for the estimation of the quadrotor's state vector. The flight controller commands the actuator thrust output according to the vehicle state vector. A radio receiver is connected to the flight controller for wireless manual control. The camera is directly connected to the onboard computer while the onboard computer is connected to the flight controller.

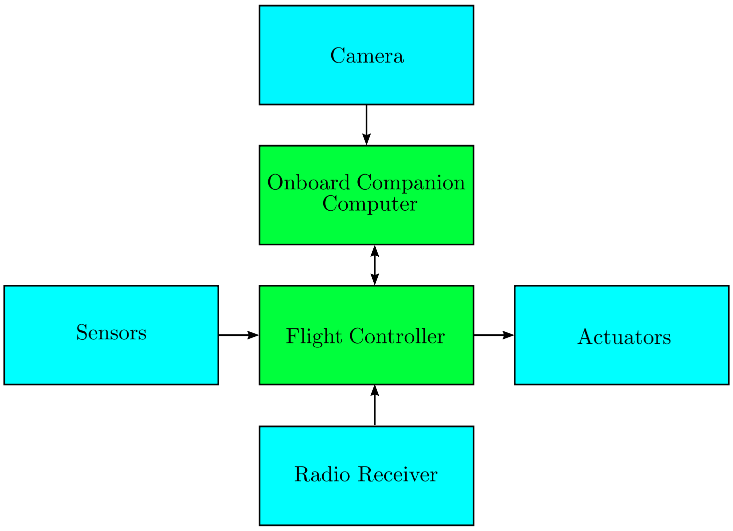


Figure 4.1: Hardware configuration on the quadrotor.

Quadrotor

The quadrotor model is based on a quadrotor named Griffin that was developed in the Electronic Systems Laboratory (ESL) in the Department of Electrical and Electronic Engineering at Stellenbosch University [41]. The quadrotor is capable of carrying payloads of up to 4.2 kg and is shown in Figure 4.2.

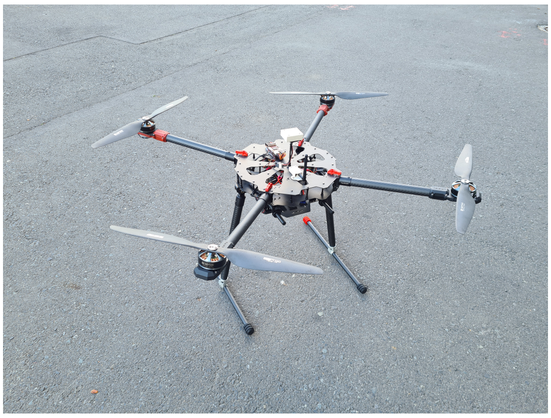


Figure 4.2: The Griffin quadrotor.

Table 4.1 contains the physical parameters of the quadrotor used to accurately model it. In work done by [41], it was found that the lumped aerodynamic drag coefficient and reference area, C_{D_q}, of the quadrotor are equal in all three coordinates of motion.

Parameter	**Value**
Mass (m_q)	4.5 kg
Moment of inertia around $\bar{\boldsymbol{x}}_{\mathcal{B}}$ (I_{xx})	0.23 kgm^2
Moment of inertia around $\bar{\boldsymbol{y}}_{\mathcal{B}}$ (I_{yy})	0.235 kgm^2
Moment of inertia around $\bar{\boldsymbol{z}}_{\mathcal{B}}$ (I_{zz})	0.328 kgm^2
Actuator distance (d)	0.49 m
Rotor chord length (r_D)	0.18 m
Rotor lift to drag ratio (R_{LD})	10
Actuator time constant (τ)	0.07 s
Aerodynamic drag coefficient (C_{D_q})	0.2 m^2

Table 4.1: Physical parameters of the Griffin quadrotor.

Flight Controller

Flight controllers typically only support specific flight control software. As the PX4 flight control stack is used, a supported flight controller should be chosen. PX4 is supported by many flight controllers, however, the Pixhawk Series is the reference hardware platform for PX4. The Pixhawk Series is open-hardware flight controllers that run PX4 on the NuttX Real Time Operating System (RTOS). These flight controllers also contain onboard sensors which include an inertial measurement unit (IMU), a magnetometer and a barometer. The Pixhawk Series has flexibility in terms of hardware peripherals that can be attached, which enables communication with onboard computers. The Pixhawk 4, shown in Figure 4.3a, is chosen for this project.

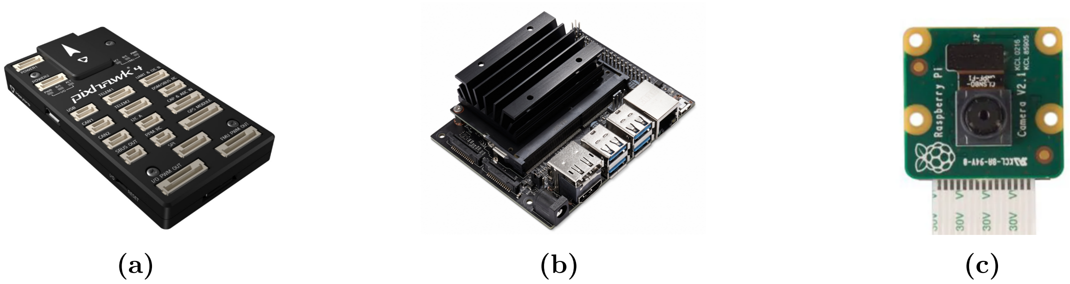


Figure 4.3: (a): Pixhawk 4 flight controller [43]. (b): Nvidia Jetson Nano onboard computer [44]. (c): Raspberry Pi camera module [45].

Onboard Computer

An onboard companion computer is required to process the image feed from the camera module for the state estimation of the suspended payload. For this purpose a Nvidia Jetson Nano, shown in Figure 4.3b is used. This is a small and powerful computer well suited for onboard image processing applications. The Nvidia Jetson Nano runs a full Linux operating system. Therefore, software developed and tested on a desktop computer is easily transfered to the Nvidia Jetson Nano.

Image Sensor

A downward facing camera, located at the bottom of the quadrotor, is used to provide state information of the suspended payload. The camera is a Raspberry Pi Camera Module v2 shown in Figure 4.3c. This is an inexpensive camera module designed for easy integration and use with onboard computers. The camera module has an eight megapixel Sony IMX219 sensor, with a maximum resolution of 3280×2464 pixels and focal length of 3.04 mm [42].

4.2. Software

The flight control toolchain includes the PX4 flight control stack, the QGroundControl (QGC) ground control software, the Robot Operating System (ROS) and the Gazebo simulation environment.

4.2.1. PX4 Flight Control Stack

PX4 is open-source flight control software for unmanned vehicles, mainly focussing on drones. PX4 has a modular structure where each module has a specific function. The communication between PX4 modules is in the form of a publish-subscribe message system which means that the system is asynchronous. This architecture makes it easy for developers to customize modules or add new modules for their needs. The basic architecture of the PX4 flight stack is shown in Figure 4.4 [46].

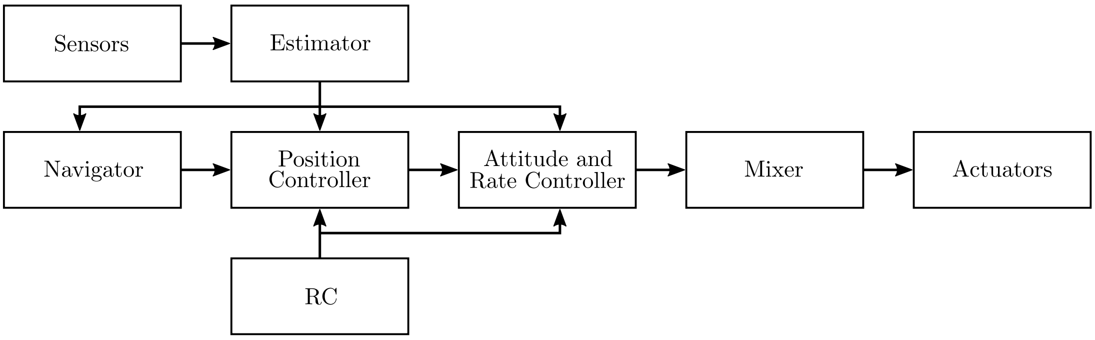


Figure 4.4: PX4 architecture.

PX4 uses a delayed time horizon Extended Kalman Filter (EKF) to provide vehicle state estimates from sensor measurements. This architecture allows the use of sensors with different time delays and data rates.

PX4 adopts a cascaded loop closure control architecture that consists of the inner attitude and outer translational controllers. PX4 uses PID controllers for the vehicle control which are discussed in detail in Chapter 5.

PX4 supports software-in-the-loop (SITL) and hardware-in-the-loop (HITL) simulations. PX4 can run on the NuttX RTOS or any other POSIX platform such as Linux. This enables developers to test their code in simulation on desktop computers, as it would be implemented on target hardware. HITL simulations enable developers to implement and test their custom firmware on the target flight controller board in the safety of a simulation environment.

4.2.2. QGroundControl

QGroundControl (QGC) served as the ground control software for this project. QGC is open-source software that provides full flight support for vehicles using the MAVLink message protocol. QGC provides a communication link between a ground station computer and a vehicle. QGC can be used for mission planning for autonomous flight as well as to monitor vehicle states during flight. QGC also aids users during vehicle setup.

4.2.3. Robot Operating System

The Robot Operating System (ROS) is a collection of open-source tools and libraries that can be used to create software for a wide variety of robotic platforms. It is not an operating system as the name suggests, but can rather be seen as a communication framework for processes in robotic applications. ROS also makes use of a publish-subscribe message system for communication between processes as shown in Figure 4.5. Processes are called nodes and can publish messages or subscribe to messages from other nodes. Messages are sent and received over topics. Nodes are required to be registered to the ROS master before it can be used. The master is responsible for setting up the communication between nodes.

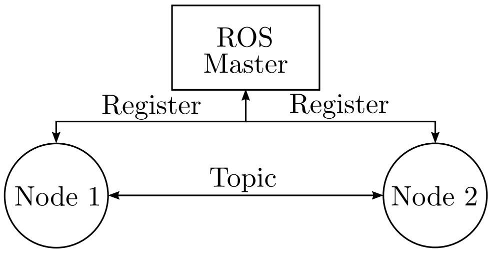


Figure 4.5: ROS communication structure.

For this project, standard and custom ROS nodes are used for communication between the onboard computer and flight controller, navigation, image processing, payload parameter estimation as well as controller adjustment.

MAVROS Node

The MAVROS node provides a communication bridge between ROS and the MAVLink message protocol used by the PX4 flight stack. This allows communication between the onboard computer and the flight controller. This node also provides a UDP MAVLink bridge for ground control stations.

Waypoint Scheduler Node

The waypoint scheduler node generates waypoints for the quadrotor to follow. The waypoints consist of a NED position and heading angle. This node is mainly used for system identification. The resulting quadrotor response is used to estimate the unknown parameters of the suspended payload.

Image Processing Node

This custom node processes the image feed received from the downward facing camera to provide state information of the swinging payload for swing damping control.

Payload Parameter Estimation Node

This node is used to process the payload state information received from the image processing node in order to estimate the cable length of the suspended payload.

Controller Adjustment Node

The controller adjustment node is used to calculate and adjust the payload swing damping controller gains. The controller gains are initialized for the nominal payload case and adjusted according to the estimated payload parameters. The controller design is discussed in Chapter 7.

4.2.4. Simulation Environment

Two simulation environments are used, namely MATLAB/Simulink and Gazebo. MATLAB/Simulink is used to design and test flight controllers and estimation algorithms. The designed controllers and algorithms are then implemented in the PX4 flight control stack and ROS and tested in SITL and HITL simulations in Gazebo. The image processing algorithms are designed and tested using the Gazebo environment.

MATLAB and Simulink

MATLAB/Simulink provides tools to aid in controller and algorithm design. The quadrotor and payload model derived in Chapter 3 and PX4 control architecture are implemented in the MATLAB/Simulink environment. Using the same controller architecture as PX4 ensures easy integration of the designed controllers.

The estimation algorithms, which are discussed in Chapter 6, are also designed and tested in the MATLAB/Simulink environments before implemented in PX4 and ROS. However, the image processing algorithm is excluded as Gazebo provides a more convenient environment.

MATLAB/Simulink allows the use of C/C++ code further simplifying integration as the PX4 flight stack is developed in C++ and ROS applications can be developed using either Python or C++.

Sensor noise and disturbances can be added in simulation to evaluate its effect on the system and the robustness of the designed controllers and estimation algorithms.

Gazebo

Gazebo is an open-source robotics simulator where robots can be designed and algorithms can be tested. Communication protocols exist for PX4 and Gazebo as well as for ROS and Gazebo. Therefore, Gazebo is the ideal simulation environment for SITL and HITL simulations.

Gazebo operates in two parts. This includes a physics engine responsible for simulating the physics of the simulation environment and a graphical renderer which visualizes the environment. Robot models are built using graphical modelling and simulation environments can be populated with multiple models. This allows for the creation of more complex models with ease. Gazebo offers a wide variety of sensors that can be added to models. Sensor noise, drift and biases can be specified for sensors to replicate real-world sensors.

Objects can also be added to the simulation environment where both its dynamic and visual properties can be specified. This allows users to replicate real-world environments and is ideal for developing and testing object avoidance and computer vision algorithms.

The quadrotor and payload system is modelled as described in Section 3.4.4. A downward facing camera, attached to the bottom of the quadrotor, is added to the model. A plugin is used to apply the forces and moments generated by the actuators to the six degree of freedom quadrotor model. A plugin is a piece of code that can be used to manipulate models or extract information, such as sensor information. The quadrotor and payload model in the Gazebo simulation environment is shown in Figure 4.6.

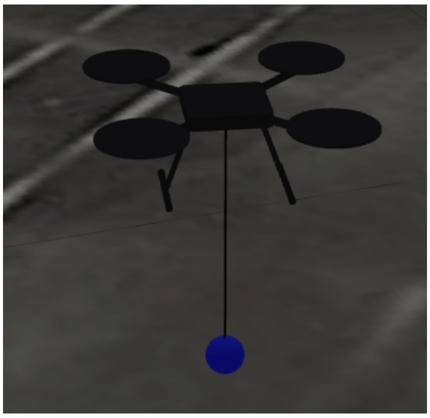


Figure 4.6: The quadrotor and payload in Gazebo.

The SITL simulation configuration is shown in Figure 4.7. The PX4 flight stack as well as ROS nodes are run on the same desktop computer as the Gazebo simulation environment. This allows testing of the designed controllers and algorithms as implemented in the PX4 flight control stack and the ROS environment.

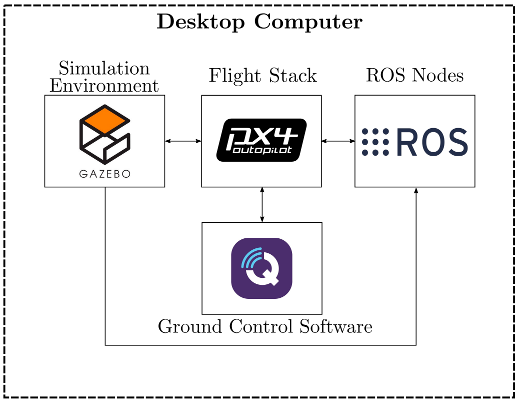


Figure 4.7: The SITL simulation configuration.

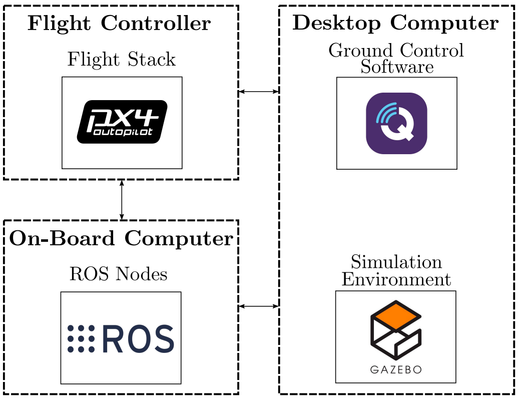


Figure 4.8: The HITL simulation configuration.

The HITL configuration is shown in Figure 4.8. In HITL simulations, the customized PX4 flight stack is uploaded to the real flight controller board. The ROS nodes are run on the onboard computer and a desktop computer is only used to run the simulation environment. This allows testing of the modified software on the target hardware.

4.3. Summary

This chapter discussed the components and tools used in this project. The Griffin quadrotor developed in the Electronic Systems Laboratory at Stellenbosch University is used with the Pixhawk 4 flight controller, Nvidia Jetson Nano onboard computer and Raspberry Pi Camera Module v2. PX4 is used for the flight control software and ROS is used for image processing, controller adjustment and parameter estimation. QGroundControl serves as the ground control software to monitor the quadrotor during flight. MATLAB/Simulink is used for controller and algorithm design. The quadrotor and payload model is implemented in Gazebo to reproduce real-world environments for SITL and HITL simulations.

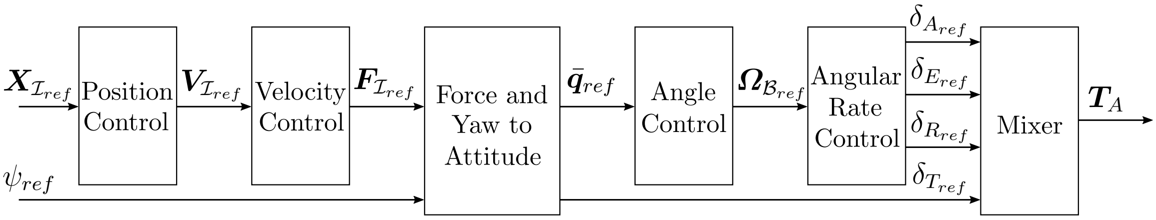
Chapter 5

Quadrotor Control System

This chapter discusses the control system design of the quadrotor without the suspended payload. The quadrotor dynamics are linearized to serve as a basis for the control design. An overview of the PX4 control architecture is given as the controllers are designed following this architecture. The controllers are verified in simulation in the MATLAB/Simulink and PX4 and Gazebo environments. The performance of the designed controllers, but with the quadrotor with a payload, is also investigated.

5.1. PX4 Architecture

PX4 adopts a cascaded loop closure control architecture. It consists of four PID controllers which include the inner angle and angular rate controllers and the outer position and linear velocity controllers. The inner attitude controllers are executed at 250 Hz while the outer translational controllers are executed at 50 Hz. The architecture is illustrated in Figure 5.1 with the control signals as described in Table 5.1.



$\boldsymbol{X}_{\mathcal{I}_{ref}}$
Position Control
$\boldsymbol{V}_{\mathcal{I}_{ref}}$
Velocity Control
$\boldsymbol{F}_{\mathcal{I}_{ref}}$
Force and Yaw to Attitude
$\bar{\boldsymbol{q}}_{ref}$
Angle Control
$\boldsymbol{\Omega}_{\mathcal{B}_{ref}}$
Angular Rate Control
$\delta_{A_{ref}}$
$\delta_{E_{ref}}$
$\delta_{R_{ref}}$
$\delta_{T_{ref}}$
Mixer
$\boldsymbol{T}_A$
ψ_{ref}

Figure 5.1: PX4 control system architecture.

The PX4 architecture ensures that the controllers can be used for various UAV platforms with little modification. This is done by using normalized forces and thrusts throughout the control loop and a mixer to convert virtual actuator commands to the actual actuator commands.

The mixer uses a mixing matrix to map the virtual actuator commands to the individual actuator commands. The mixing matrix implemented by PX4 for a quadrotor in the cross configuration is given by

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}_{PX4} = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \\ 1 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \\ 1 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -1 \\ 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -1 \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix}_{PX4} . \tag{5.1}$$

Control Signal	Description
$\boldsymbol{X}_{\mathcal{I}_{ref}}$	Position reference in inertial frame
ψ_{ref}	Yaw reference
$\boldsymbol{V}_{\mathcal{I}_{ref}}$	Linear velocity reference in inertial frame
$\boldsymbol{F}_{\mathcal{I}_{ref}}$	Force reference in inertial frame
$\bar{\boldsymbol{q}}_{ref}$	Unit quaternion reference
$\boldsymbol{\Omega}_{\mathcal{B}_{ref}}$	Angular rate reference in body frame
$\delta_{A_{ref}}$	Aileron reference
$\delta_{E_{ref}}$	Elevator reference
$\delta_{R_{ref}}$	Rudder reference
$\delta_{T_{ref}}$	Thrust reference
$\boldsymbol{T}_A$	Actuator reference

Table 5.1: Control signals of the PX4 control architecture.

It should be noted that the mixer implemented by PX4 also contains actuator saturation checks which is not represented in Equation (5.1).

As stated, PX4 makes use of normalized forces and thrusts. The virtual aileron, elevator and rudder actuator commands are normalized to $[-1,\, 1]$, while the virtual thrust and actual actuator commands are normalized to $[\,0,\, 1]$. These normalized thrusts can be multiplied with the maximum thrust produced by the actual actuators to determine the forces and moments produced. Therefore, the thrust produced by an actuator is given by

$$\boldsymbol{T}_i = T_{max}\boldsymbol{T}_{i_{PX4}}, \tag{5.2}$$

where T_{max} is the maximum thrust produced by an actuator, $\boldsymbol{T}_{i_{PX4}}$ is the normalized thrust used by PX4 and i represents the actuator number. Using Equations (3.23), (5.1) and (5.2), the relationship between the virtual actuator and normalized virtual actuator commands is found to be

$$\begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix} = T_{max} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_A \\ \delta_E \\ \delta_R \end{bmatrix}_{PX4} . \tag{5.3}$$

This relationship needs to be taken into account when designing controller gains for the PX4 control architecture.

5.2. Linearized Quadrotor Model

The non-linear quadrotor model derived in Chapter 3 is linearized to serve as the basis for the quadrotor controller design. Small perturbation theory is applied to find the linear dynamics at the hover trim condition.

Recall the non-linear quadrotor dynamics given in Equations (3.1), (3.2), (3.9) and (3.10) which can be represented in state space form by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{5.4}$$

where

$$\mathbf{x} = \begin{bmatrix} U & V & W & P & Q & R & q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T, \tag{5.5}$$

$$\mathbf{u} = \begin{bmatrix} \delta_{T_{ref}} & \delta_{A_{ref}} & \delta_{E_{ref}} & \delta_{R_{ref}} \end{bmatrix}^T. \tag{5.6}$$

The position dynamics do not form part of the fundamental quadrotor dynamics and are therefore omitted from Equation (5.5). Each state and control input can be defined as the sum of the trim condition and a small perturbation as

$$\mathbf{x} = \mathbf{x}_{trim} + \Delta\mathbf{x} \tag{5.7}$$

$$\mathbf{u} = \mathbf{u}_{trim} + \Delta\mathbf{u}. \tag{5.8}$$

where the trim conditions and small perturbations are defined as

$$\mathbf{x}_{trim} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \tag{5.9}$$

$$\mathbf{u}_{trim} = \begin{bmatrix} T_{hover} & 0 & 0 & 0 \end{bmatrix}^T, \tag{5.10}$$

$$\Delta\mathbf{x} = \begin{bmatrix} \Delta U & \Delta V & \Delta W & \Delta P & \Delta Q & \Delta R & \Delta q_0 & \Delta q_1 & \Delta q_2 & \Delta q_3 \end{bmatrix}^T \text{ and} \tag{5.11}$$

$$\Delta\mathbf{u} = \begin{bmatrix} \delta_{t_{ref}} & \delta_{a_{ref}} & \delta_{e_{ref}} & \delta_{r_{ref}} \end{bmatrix}^T, \tag{5.12}$$

where T_{hover} is the thrust required to hover. Using a Taylor series expansion, Equation (5.4) can be linearized at the hover trim condition and the quadrotor dynamics can be written as

$$\boldsymbol{F}_{\mathcal{B}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} m_q\Delta\dot{U} \\ m_q\Delta\dot{V} \\ m_q\Delta\dot{W} \end{bmatrix}, \tag{5.13}$$

$$\boldsymbol{M}_{\mathcal{B}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\Delta\dot{P} \\ I_{yy}\Delta\dot{Q} \\ I_{zz}\Delta\dot{R} \end{bmatrix} \text{ and} \tag{5.14}$$

$$\dot{\boldsymbol{q}}_v = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \Delta P \\ \Delta Q \\ \Delta R \end{bmatrix}. \tag{5.15}$$

By applying small angle approximations and ignoring the drag forces given in Equation (3.16), the forces and moments are also linearized around hover. The linearized forces and moments are given by

$$\boldsymbol{F}_{\mathcal{B}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2q_2 m_q g \\ -2q_1 m_q g \\ -\delta_T + m_q g \end{bmatrix} \text{ and} \tag{5.16}$$

$$\boldsymbol{M}_{\mathcal{B}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} d\delta_A \\ d\delta_E \\ \frac{r_D}{R_{LD}}\delta_R \end{bmatrix}. \tag{5.17}$$

Lastly, assuming a yaw angle of zero, the linear actuator dynamics given in Equation (3.18), can be applied to the virtual actuator dynamics and is given by

$$\dot{\delta}_i = \frac{-\delta_i}{\tau} + \frac{\delta_{i_{ref}}}{\tau}, \tag{5.18}$$

where $i = \{T,\, A,\, E,\, R\}$.

The linear system dynamics are found by combining Equations (5.13) to (5.18). The linear dynamics can be decoupled to provide the longitudinal, lateral, directional and heave dynamics which are written in state space form as:

Longitudinal Dynamics:

$$\dot{\mathbf{x}}_{long} = \mathbf{A}_{long}\mathbf{x}_{long} + \mathbf{B}_{long}\mathbf{u}_{long} \tag{5.19}$$

$$\begin{bmatrix} \dot{\delta}_E \\ \dot{Q} \\ \dot{q}_2 \\ \dot{U} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 & 0 & 0 \\ \frac{d}{I_{yy}} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 2g & 0 \end{bmatrix} \begin{bmatrix} \delta_E \\ Q \\ q_2 \\ U \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_{E_{ref}} \tag{5.20}$$

Lateral Dynamics:

$$\dot{\mathbf{x}}_{lat} = \mathbf{A}_{lat}\mathbf{x}_{lat} + \mathbf{B}_{lat}\mathbf{u}_{lat} \tag{5.21}$$

$$\begin{bmatrix} \dot{\delta}_A \\ \dot{P} \\ \dot{q}_1 \\ \dot{V} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 & 0 & 0 \\ \frac{d}{I_{xx}} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & -2g & 0 \end{bmatrix} \begin{bmatrix} \delta_A \\ P \\ q_1 \\ V \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta_{A_{ref}} \tag{5.22}$$

Directional Dynamics:

$$\dot{\mathbf{x}}_{dir} = \mathbf{A}_{dir}\mathbf{x}_{dir} + \mathbf{B}_{dir}\mathbf{u}_{dir} \tag{5.23}$$

$$\begin{bmatrix} \dot{\delta}_R \\ \dot{R} \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 & 0 \\ \frac{r_D}{R_{LD} I_{xx}} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} \delta_R \\ R \\ q_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \\ 0 \end{bmatrix} \delta_{R_{ref}} \tag{5.24}$$

Heave Dynamics:

$$\dot{\mathbf{x}}_h = \mathbf{A}_h\mathbf{x}_h + \mathbf{B}_h\mathbf{u}_h \tag{5.25}$$

$$\begin{bmatrix} \dot{\delta}_T \\ \dot{W} \\ V_D \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 & 0 \\ -\frac{1}{m_q} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta_T \\ W \\ D \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \\ 0 \end{bmatrix} \delta_{T_{ref}} + \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}, \tag{5.26}$$

where V_D is the Down velocity in the inertial frame.

Lastly, the horizontal inertial velocity are related to the body velocity as

$$V_N = U \tag{5.27}$$

$$V_E = V. \tag{5.28}$$

With the linear plant derived, the controllers can now be designed.

5.3. Control System Design

The controllers are designed based on the linearized quadrotor dynamics derived in Section 5.2. The controllers are designed in the continuous-time and should be discretized for discrete implementation in PX4. However, as the execution rates of the PX4 controllers are much higher than the bandwidth of the quadrotor dynamics, the continuous controller gains can be implemented without discretization.

The controllers are designed starting at the inner-most angular rate controller and ending with the outer-most position controller. Only the design of the longitudinal controllers are presented in this section. The design process of the lateral, directional and heave controllers are very similar to that of the longitudinal controllers.

5.3.1. Angular Rate Controllers

The angular rate controllers are the inner-most controllers and very important as it affects the performance of the outer controllers. These controllers are responsible for the stability of the quadrotor and follow angular rate references, $\boldsymbol{\Omega}_{\mathcal{B}_{ref}} = \{P_{ref},\ Q_{ref},\ R_{ref}\}$, to provide the virtual actuator thrust commands $\boldsymbol{\delta}_{ref} = \{\delta_{A_{ref}},\ \delta_{E_{ref}},\ \delta_{R_{ref}}\}$. The controllers receive attitude feedback directly from the gyroscope on the quadrotor. Three PID controllers are implemented for roll, pitch and yaw rate control. The controller structure is illustrated in Figure 5.2.

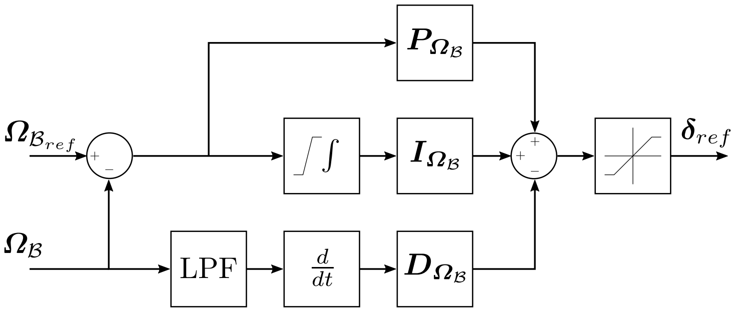

$\boldsymbol{\Omega}_{\mathcal{B}_{ref}}$
+
−
$\boldsymbol{\Omega}_{\mathcal{B}}$
$\boldsymbol{P}_{\boldsymbol{\Omega}_{\mathcal{B}}}$
$\int$
$\boldsymbol{I}_{\boldsymbol{\Omega}_{\mathcal{B}}}$
LPF
$\frac{d}{dt}$
$\boldsymbol{D}_{\boldsymbol{\Omega}_{\mathcal{B}}}$
+
+
−
$\boldsymbol{\delta}_{ref}$

Figure 5.2: PX4 Angular Rate Controller.

PX4 added modifications to the standard PID control structure to improve the controller performance. Only the modifications relevant to this project are presented.

A low pass filter (LPF) is added to the derivative path to prevent the amplification of noise from noisy signals. The derivative path acts on the state instead of the error. This prevents a phenomenon called "derivative kick" which occurs when there is a sudden jump in the error due to a change in the reference. A saturation limit is added to the integral path to prevent integrator windup which occurs when the control signal is saturated for an extended period of time. Lastly, the control signal is limited to prevent actuator saturation.

Consider the longitudinal dynamics without the pitch angle and linear velocity:

$$\dot{\mathbf{x}}^*_{long} = \mathbf{A}^*_{long}\mathbf{x}^*_{long} + \mathbf{B}^*_{long}\mathbf{u}^*_{long} \tag{5.29}$$

$$\begin{bmatrix} \dot{\delta}_E \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ \frac{d}{I_{yy}} & 0 \end{bmatrix} \begin{bmatrix} \delta_E \\ Q \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix} \delta_{E_{ref}} \tag{5.30}$$

The linear pitch rate plant is then determined to be

$$G_Q(s) = \frac{Q(s)}{\delta_{E_{ref}}(s)} = \mathbf{C}_Q \left(s\mathbf{I} - \mathbf{A}^*_{long}\right)^{-1} \mathbf{B}^*_{long} \tag{5.31}$$

$$G_Q(s) = \frac{\frac{d}{\tau I_{yy}}}{s\left(s + \frac{1}{\tau}\right)}, \tag{5.32}$$

where

$$\mathbf{C}_Q = \begin{bmatrix} 0 & 1 \end{bmatrix}. \tag{5.33}$$

The transfer function, $G_Q(s)$, needs to be adapted as PX4 uses normalized actuator commands. Therefore, applying Equation (5.3), the linear pitch rate plant used for the controller design is

$$G_Q^{PX4}(s) = \frac{Q(s)}{\delta_{E_{ref}}^{PX4}} = \frac{2T_{max}\frac{d}{\tau I_{yy}}}{s\left(s + \frac{1}{\tau}\right)}. \tag{5.34}$$

The block diagram of the system used for the pitch rate controller design is given in Figure 5.3.

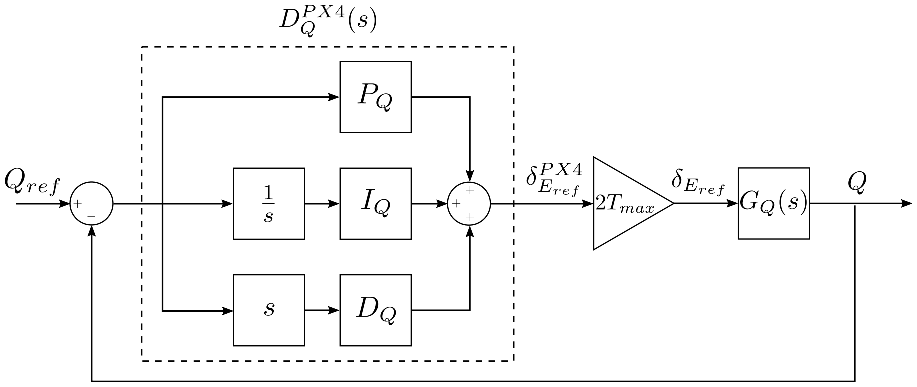


Figure 5.3: Pitch rate block diagram.

As the angular rate controllers affect the whole system, these controllers should have a quick and well damped response. The angular rate controllers should also be able to reject steady-state tracking errors and external disturbances due to wind, asymmetry in produced actuator thrusts or an offset in the quadrotor CoM.

The root locus of the linear plant is shown in Figure 5.4 and the root locus of the linear plant and designed PID controller is shown in Figure 5.5. From Figure 5.4, it can be seen that the pitch rate plant has two poles, one at the origin and one at $s = -14.29$ due to the actuator lag dynamics.

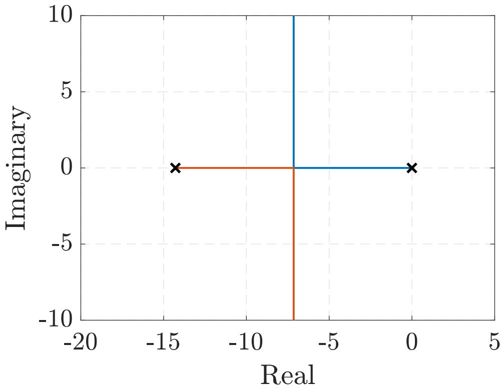


Figure 5.4: Root locus of the pitch rate plant.

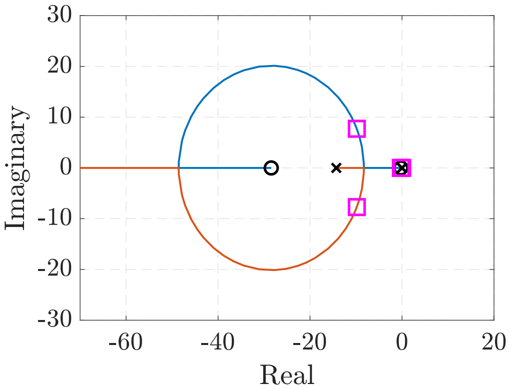


Figure 5.5: Root locus of the pitch rate controller.

The controller design process starts with the proportional gain which is chosen to obtain the desired bandwidth and decrease the system response time. The integral term is added to reject disturbance torques. With one free integrator present in the pitch rate dynamics, the integral term makes the system a type 2 system able to track step and ramp inputs with a zero steady-state error. The integral term adds a zero at $s = -0.233$ and a pole at the origin. The resulting system features three closed-loop poles of which one is near the origin and two are underdamped. Pole-zero cancellation minimizes the effects of the pole near zero. However, the two underdamped poles results in increased overshoot. Therefore, the derivative term is added to provide more damping.

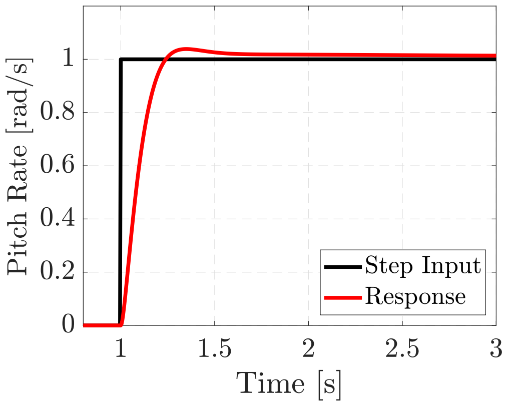


Figure 5.6: Pitch rate controller step response.

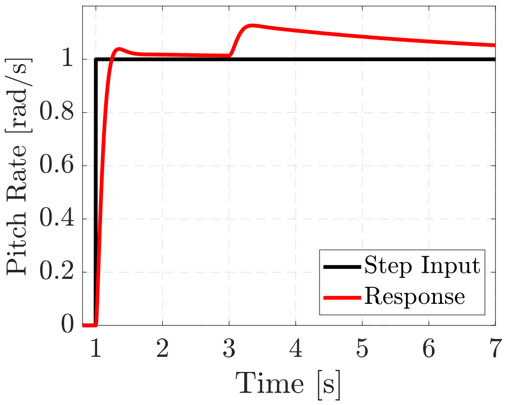


Figure 5.7: Pitch rate controller disturbance rejection.

The step response of the pitch rate controller is shown in Figure 5.6. The response has an overshoot of 4%, a 2% settling time of 0.6 s and a bandwidth of 12.53 rad/s. Figure 5.7 shows the response when a constant disturbance torque is introduced at time $t = 3\ s$. The integrator term is able to reject the disturbance and produce a zero steady-state error. By increasing the integrator gain, the disturbance torque can be negated more quickly, but at the expense of greater overshoot. However, a more damped response is preffered.

5.3.2. Angular Controllers

The angular controllers allow the quadrotor to follow roll, pitch and yaw references by providing angular rate commands in the body frame. PX4 implements quaternion based attitude control as discussed in [47]. The controller structure is shown in Figure 5.8 with the control law given by

$$\boldsymbol{\Omega}_{\mathcal{B}_{ref}} = 2\boldsymbol{P}_q \operatorname{sgn}(q_{0_e})\boldsymbol{q}_{v_e}, \qquad \operatorname{sgn}(q_{0_e}) = \begin{cases} 1, & q_{0_e} \geq 0 \\ -1, & q_{0_e} < 0 \end{cases} \tag{5.35}$$

where q_{0_e} is the magnitude and $\boldsymbol{q}_{v_e}$ is the vector part of the error quaternion, $\bar{\boldsymbol{q}}_e$, and $\boldsymbol{P}_q$ is the proportional gains of the controllers. The error quaternion is determined by

$$\bar{\boldsymbol{q}}_e = \bar{\boldsymbol{q}}^{-1} \cdot \bar{\boldsymbol{q}}_{ref}, \tag{5.36}$$

where $\bar{\boldsymbol{q}}$ and $\bar{\boldsymbol{q}}_{ref}$ are the quadrotor and reference attitude quaternions, respectively.

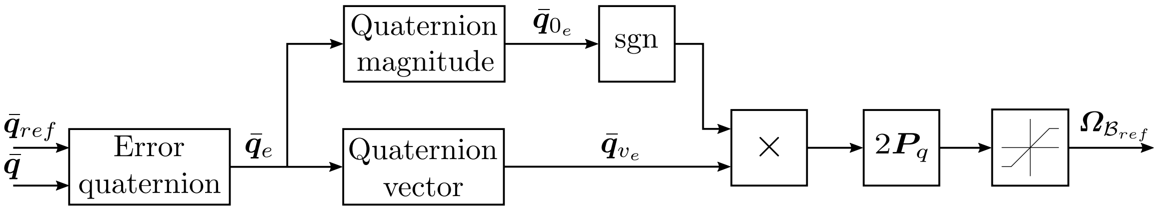


Figure 5.8: PX4 Angle Controller.

The inner PID angular rate controllers and velocity controllers, which form the next control loop and are discussed in Section 5.3.3, are capable of rejecting steady-state disturbances. Therefore, only proportional control can be implemented for the angular controllers.

From the linear longitudinal dynamics given by Equation (5.19), the linear pitch angle plant is determined to be

$$G_{q_2}(s) = \frac{q_2(s)}{Q_{ref}(s)} = G_{Q_{cl}}^{PX4}(s) \cdot \frac{1}{2} \cdot \frac{1}{s}, \tag{5.37}$$

where $G_{Q_{cl}}^{PX4}(s)$ is the closed-loop pitch rate dynamics given by

$$G_{Q_{cl}}^{PX4}(s) = \frac{D_Q^{PX4}(s) \cdot G_Q^{PX4}(s)}{1 + D_Q^{PX4}(s) \cdot G_Q^{PX4}(s)}. \tag{5.38}$$

The control law given in Equation (5.35) is linearized by applying small angle approximation and assuming that the error magnitude, q_{0_e}, is positive. The linearized control law for the pitch angle, which also applies to the roll and yaw angles, is given by

$$Q_{ref} = 2P_{q_2}\left(q_{2_{ref}} - q_2\right). \tag{5.39}$$

The block diagram of the system used for the pitch angle controller design is shown in Figure 5.9.

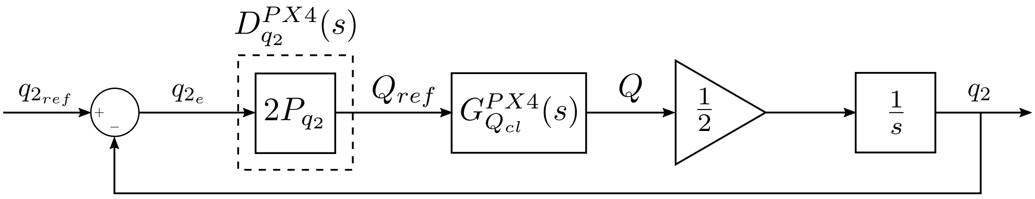


Figure 5.9: Pitch angle block diagram.

The pitch angle controller should have a quick and well damped response. A good time-scale separation between the pitch angle and pitch rate controllers is also important and can be achieved by designing the pitch angle controller to have a bandwidth that is less than half of the pitch rate controller bandwidth.

The root locus of the linear pitch angle plant is shown in Figure 5.10 and the root locus of the linear pitch angle plant and proposed controller is shown in Figure 5.11. The resulting system is overdamped due to the most dominant pole located at $s = -4.64$ on the real axis. The open-loop system has one free integrator which makes it a type 1 system capable of tracking step references with a zero steady-state error.

The step response of the pitch angle controller is shown in Figure 5.12. The response has no overshoot, a 2% settling time of 0.95 s and a bandwidth of 4.41 rad/s which is less than half the bandwidth of the pitch rate controller.

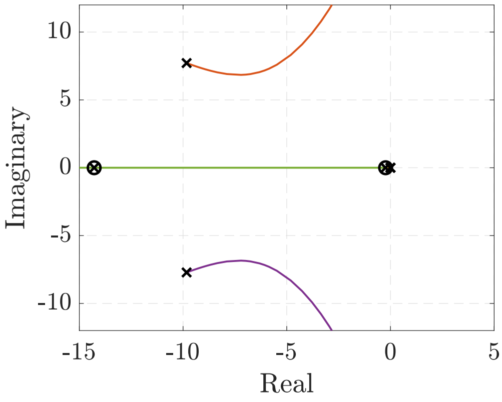


Figure 5.10: Root locus of the pitch angle plant.

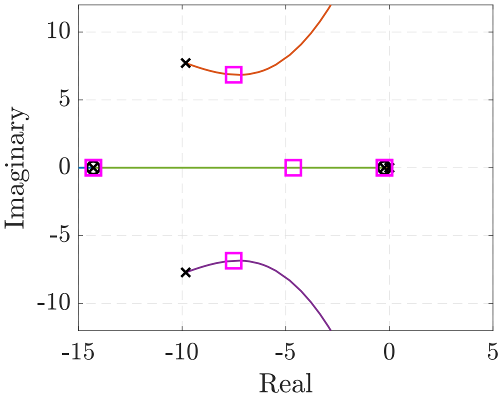


Figure 5.11: Root locus of the pitch angle controller.

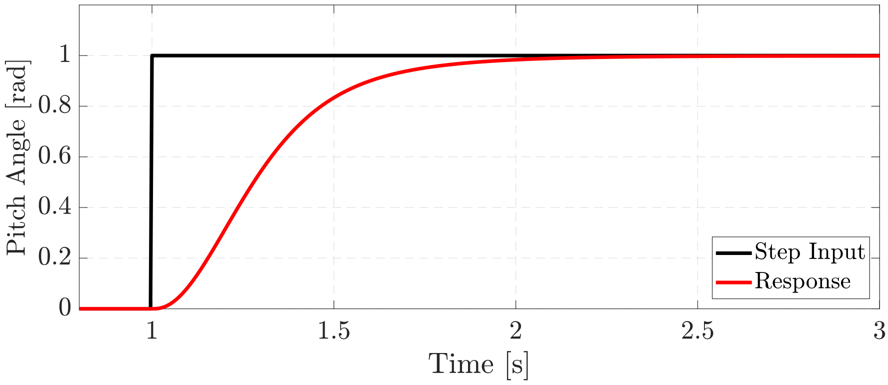


Figure 5.12: Pitch angle controller step response.

5.3.3. Velocity Controllers

The velocity controllers enable the quadrotor to follow velocity references, $\boldsymbol{V}_{\mathcal{I}_{ref}} = \{V_{N_{ref}},\ V_{E_{ref}},\ V_{D_{ref}}\}$, in the inertial frame by providing an inertial force reference, $\boldsymbol{F}_{\mathcal{I}_{ref}} = \{F_{N_{ref}},\ F_{E_{ref}},\ F_{D_{ref}}\}$. The force reference, together with a yaw reference, are converted to the desired attitude and thrust. This conversion is based on work done by [48].

Three PID controllers are implemented for the North, East and Down velocity control. The controller structure of the angular rate controllers as discussed in Section 5.3.1 is also implemented for the linear velocity controllers.

The linear longitudinal velocity plant is determined by considering the linearized dynamics from Equations (5.19) and (5.27) which show that the relationship between the acceleration in the North direction and the pitch quaternion component is

$$\dot{V}_D = \dot{U} = 2gq_2. \tag{5.40}$$

The linear plant describing the dynamics from the reference pitch quaternion component to the longitudinal velocity is then determined to be

$$G_{V_N}(s) = \frac{V_N(s)}{q_{2_{ref}}(s)} = G^{PX4}_{q_{2_{cl}}}(s) \cdot \frac{2g}{s}, \tag{5.41}$$

where the closed-loop pitch dynamics is given by

$$G^{PX4}_{q_{2_{cl}}}(s) = \frac{D^{PX4}_{q_2}(s) \cdot G_{q_2}(s)}{1 + D^{PX4}_{q_2}(s) \cdot G_{q_2}(s)}. \tag{5.42}$$

However, as the PX4 velocity controllers command a normalized inertial force, the linear plant is adapted to describe the dynamics from the commanded normalized longitudinal force to the longitudinal velocity. Considering the relationship between the pitch quaternion component and commanded longitudinal force derived from Equation (5.16),

$$F_{N_{ref}} = 2q_{2_{ref}} m_q g, \tag{5.43}$$

and the relationship between the commanded longitudinal force and normalized force,

$$F_{N_{ref}} = 4T_{max} F^{PX4}_{N_{ref}}, \tag{5.44}$$

the adapted linear longitudinal velocity plant is then expressed as

$$G_{V_N}(s) = \frac{V_N(s)}{F^{PX4}_{N_{ref}}(s)} = G^{PX4}_{q_{2_{cl}}}(s) \cdot \frac{4T_{max}}{m_q} \cdot \frac{1}{s}. \tag{5.45}$$

The block diagram of the linear plant and controller is shown in Figure 5.13.

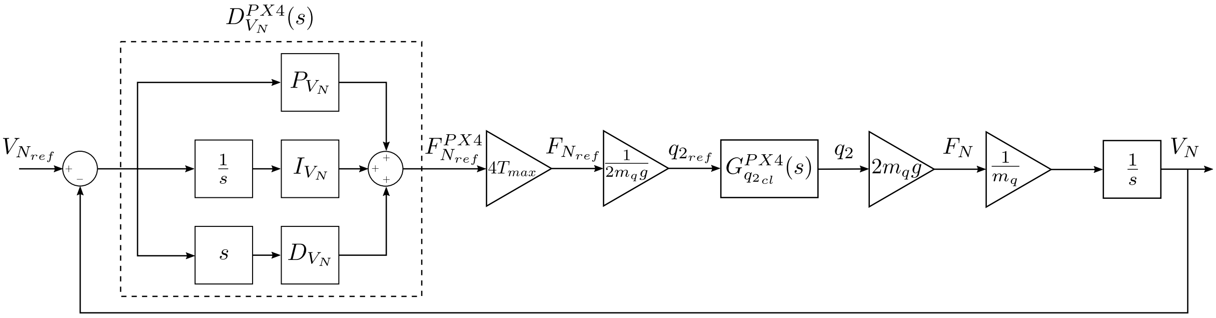


Figure 5.13: Longitudinal velocity block diagram.

The velocity controllers should have a well damped response and be able to reject steady-state tracking errors and drift caused by external disturbance forces. For the design of the velocity controllers, more emphasis is placed on the steady-state tracking capabilities than on the transient response.

The root locus of the linear longitudinal velocity plant is shown in Figure 5.14 and the root locus of the linear plant and designed PID controller is shown in Figure 5.15.

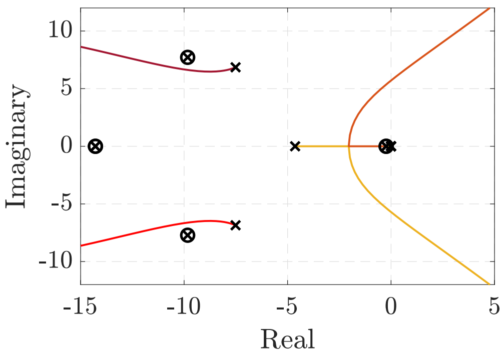


Figure 5.14: Root locus of the longitudinal velocity plant.

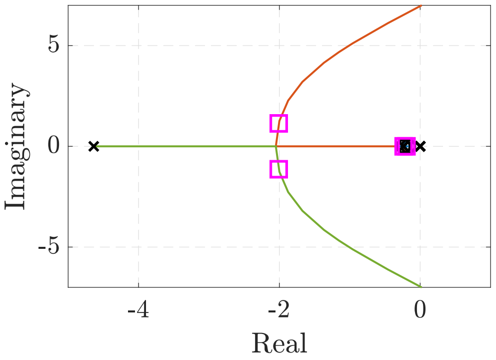


Figure 5.15: Root locus of the longitudinal velocity controller.

The longitudinal velocity controller design process is similar to that of the pitch rate controller. The proportional gain is chosen to obtain the desired bandwidth to ensure good time-scale separation between the velocity and pitch angle controllers. The integral term is added to reject disturbance forces and make the system a type 2 system. Lastly, the derivative term is added to provide more damping.

The step response of the longitudinal velocity controller is shown in Figure 5.16. The response has an overshoot of 12%, a 2% settling time of 11.5 s and a bandwidth of 2.17 rad/s which is less than half the 4.41 rad/s bandwidth of the pitch angle controller. The step response shows a long tail due to the integrator term. The slower transient response is a result of the emphasis being placed on the steady-state tracking performance of the controller, which is shown in Figure 5.17 where a constant disturbance force is introduced at time $t = 12\ s$.

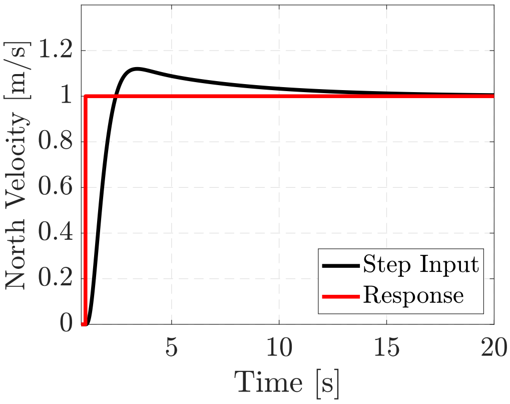


Figure 5.16: Longitudinal velocity controller step response.

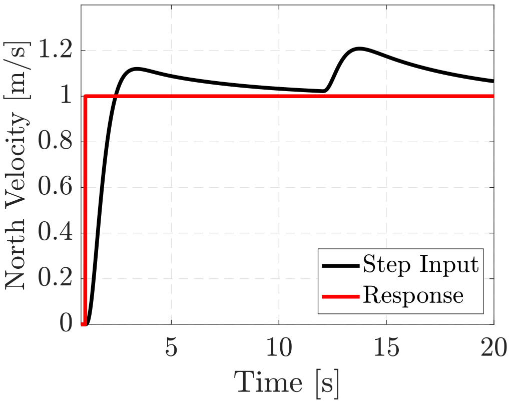


Figure 5.17: Longitudinal velocity controller disturbance rejection.

5.3.4. Position Controllers

The position controllers are the outer-most controllers and allow the quadrotor to follow a position reference, $\boldsymbol{X}_{\mathcal{I}_{ref}} = \{N_{ref},\ E_{ref},\ D_{ref}\}$, by commanding an inertial velocity reference, $\boldsymbol{V}_{\mathcal{I}_{ref}}$. Standard proportional control is implemented for the position controllers as shown in Figure 5.18. Integral control is not required for the position controllers as steady-state disturbances are rejected by the inner velocity controllers.

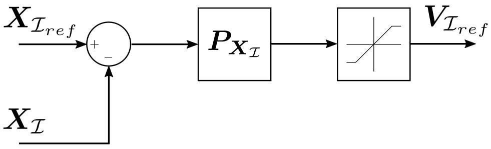

$\boldsymbol{X}_{\mathcal{I}_{ref}}$
+
−
$\boldsymbol{P}_{\boldsymbol{X}_{\mathcal{I}}}$
$\boldsymbol{V}_{\mathcal{I}_{ref}}$
$\boldsymbol{X}_{\mathcal{I}}$

Figure 5.18: PX4 position controller.

The linear North position plant is given by

$$G_N(s) = \frac{N(s)}{V_{N_{ref}}(s)} = G^{PX4}_{V_{N_{cl}}}(s) \cdot \frac{1}{s}, \tag{5.46}$$

where the closed-loop longitudinal velocity dynamics is given by

$$G^{PX4}_{V_{N_{cl}}}(s) = \frac{D^{PX4}_{V_N}(s) \cdot G_{V_N}(s)}{1 + D^{PX4}_{V_N}(s) \cdot G_{V_N}(s)}. \tag{5.47}$$

The block diagram of the linear North position plant and proportional controller is shown in Figure 5.19.

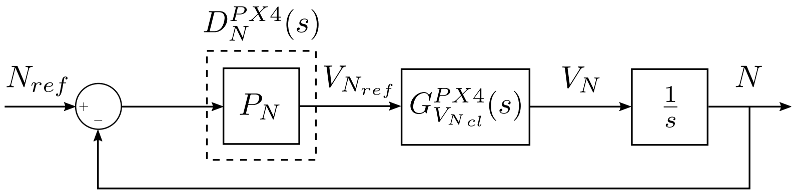


Figure 5.19: North position block diagram.

The position controller is designed to have a well damped response. The root locus of the linear North position plant is shown in Figure 5.20 and the root locus of the linear plant and controller is shown in Figure 5.21.

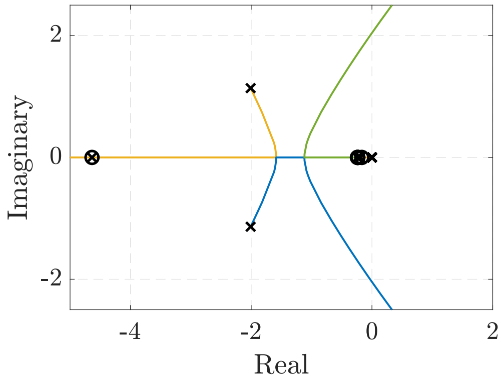


Figure 5.20: Root locus of the North position plant.

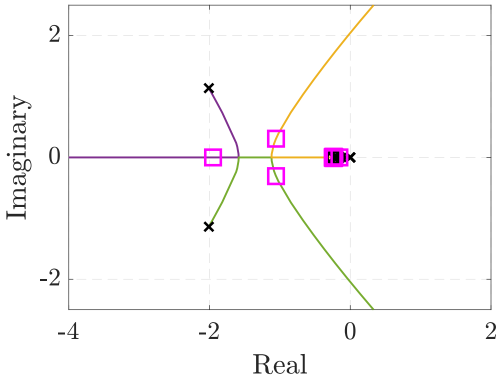


Figure 5.21: Root locus of the North position controller.

The step response of the North position controller is shown in Figure 5.22. The response has no overshoot, a 2% settling time of 11.55 s and a bandwidth of 0.59 rad/s which is less than half of the longitudinal velocity controller bandwidth of 2.17 rad/s.

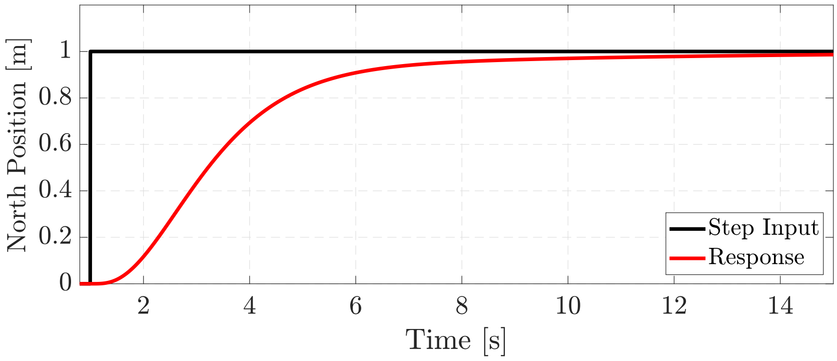


Figure 5.22: North position controller step response.

This concludes the design of the longitudinal controllers. The same process is followed to design the lateral, directional and heave controllers. The controller gains are provided in Appendix A.

5.4. Simulation Results

The designed controllers are implemented and verified using the non-linear model in the MATLAB/Simulink simulation environment. A comparison is made between the linear and non-linear responses. The controllers are also implemented in the PX4 flight control stack and verified using SITL simulations in the Gazebo simulation environment. The SITL simulations provide confirmation that the designed controllers are implemented correctly in the PX4 flight stack and perform as expected when compared to the linear and non-linear MATLAB/Simulink simulations.

The effect of sensor noise is also investigated using the SITL simulation to ensure that the controllers still perform adequately under the influence of high-frequency sensor noise, low-frequency sensor drift and sensor biases. The performance of the designed controllers when a suspended payload is attached to the quadrotor are investigated.

5.4.1. Simulation Results of the Quadrotor

Figures 5.23 to 5.26 show the pitch rate, pitch angle, longitudinal velocity and North position step responses of the linear and non-linear MATLAB/Simulink simulations and SITL simulations. Sensor noise is excluded from the simulations in order to compare the responses. It is clear that the differences between the responses are negligible and that the different simulations provide very similar results. It can therefore be concluded that the implemented controllers meet the design requirements.

Figure 5.27 shows the North position response of the quadrotor under the influence of sensor noise. High-frequency noise, low-frequency drift and sensor biases were added to

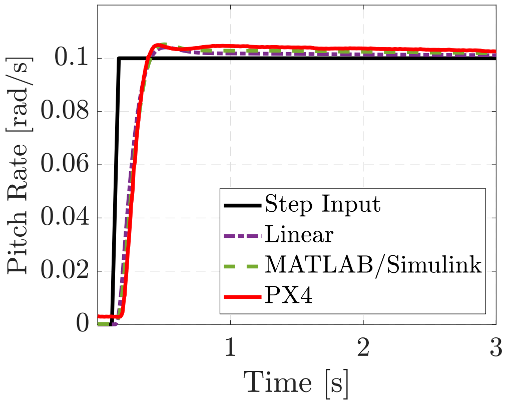


Figure 5.23: Pitch rate step response.

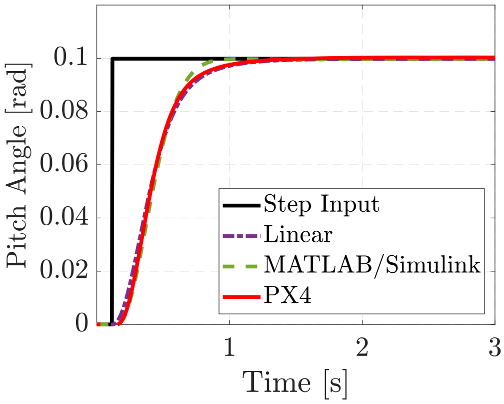


Figure 5.24: Pitch angle step response.

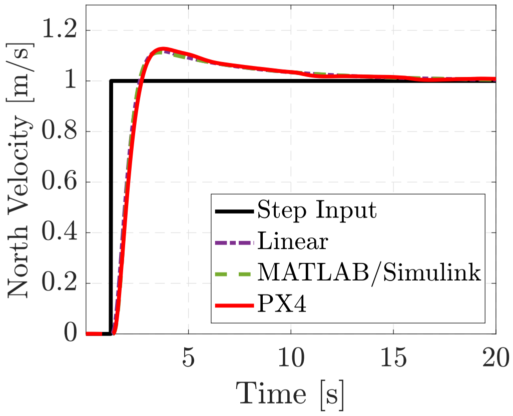


Figure 5.25: Longitudinal velocity step response.

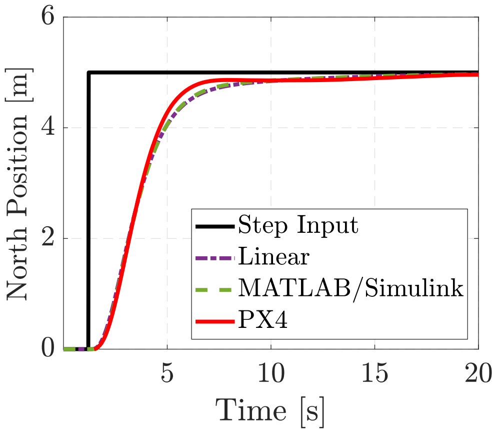


Figure 5.26: North position step response.

the quadrotor model in the SITL simulation environment. From the North position step response, it can be seen that the controllers still perform well.

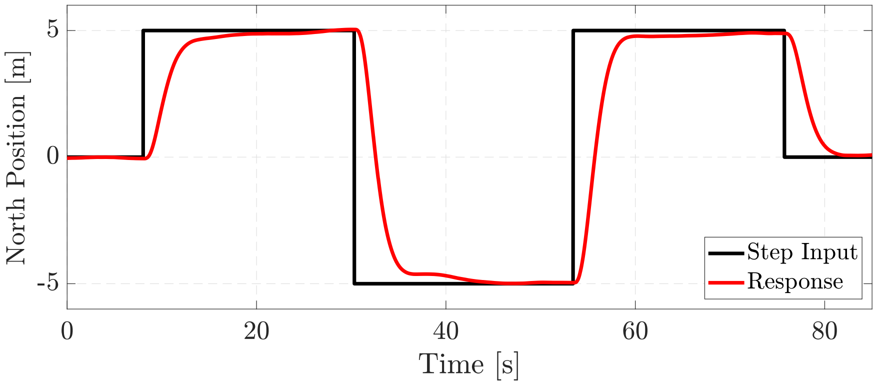


Figure 5.27: PX4 and Gazebo North position step response with sensor noise.

5.4.2. Simulation Results of the Quadrotor with a Suspended Payload

As stated, the standard quadrotor controllers serve as the basis control system for the quadrotor and payload system. As it is assumed that the payload is attached to the center of mass of the quadrotor, the suspended payload only affects the translational dynamics of the quadrotor. The performance of the longitudinal velocity controller under the influence of the nominal payload which has a mass of 2 kg and a cable length of 1 m is shown in Figure 5.28 with the response of the resulting payload swing angle, $\theta_p^{\mathcal{I}}$, shown in Figure 5.29. The lateral velocity step response is similar and not shown here.

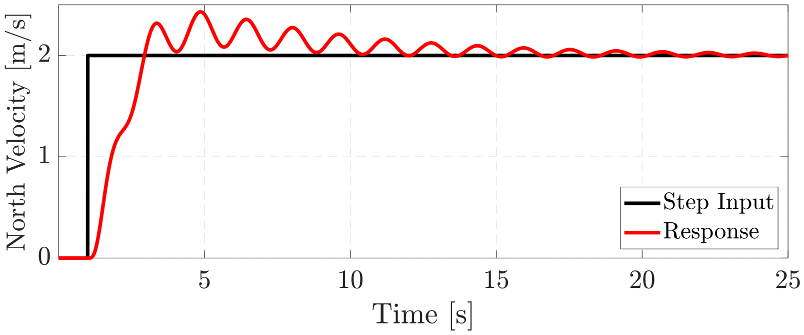


Figure 5.28: Longitudinal velocity step response of the quadrotor with a suspended payload.

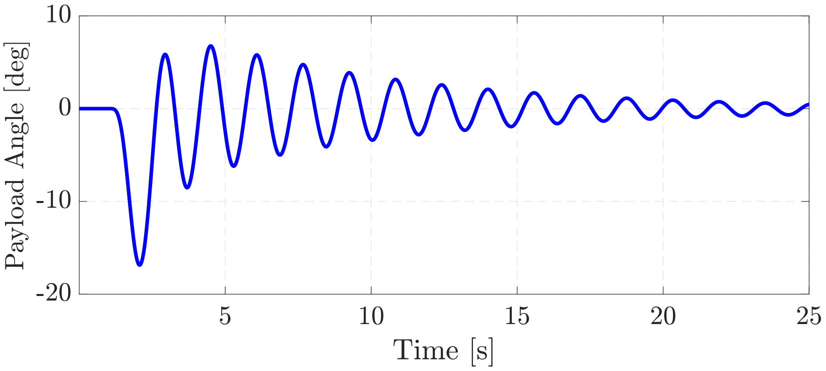


Figure 5.29: Payload angle response to a velocity reference.

The suspended payload greatly affects the longitudinal velocity reponse of the system. The standard controller is still able to damp the swing angle of the nominal payload. However, the controller shows a lightly damped response which is not ideal. Constructive interference may occur when a series of maneuvers are performed before the payload oscillations are damped which may ultimately result in instability.

5.5. Summary

This chapter provides an overview of the PX4 control architecture for which the controller gains were designed. The quadrotor model derived in Chapter 3 was linearized and the linear model served as the basis for the controller design. The designed controllers were implemented and verified using the non-linear quadrotor models in the MATLAB/Simulink and PX4 and Gazebo simulation environments. It is clear that the controllers meet the design requirements and are robust against sensor noise. The standard controllers were tested on the quadrotor and suspended payload system and it was observed that the system response is lightly damped and may result in unstable flight. Therefore, swing damping control is required to ensure stable flight when a suspended payload is attached to the quadrotor. The standard controllers will serve as the basis system against which the swing damping control system discussed in Chapter 7 will be compared.

Chapter 6

Payload Parameter and State Estimation

This chapter discusses the estimation of the unknown suspended payload parameters, the vision system that provides measurements of the suspended payload swing angles as well as the estimator used to provide payload state estimates for full-state feedback control.

Firstly, an overview of the estimation process is given. The camera model used is discussed, followed by a discussion on the payload detection and tracking algorithms implemented in the vision system. The algorithms implemented to estimate the unknown payload parameters, which include the cable length, l, and the payload mass, m_p, are discussed. Lastly, an overview of the Extended Kalman Filter (EKF) implemented to provide payload state estimates are given.

6.1. Estimation Process Overview

As there is no direct access to the suspended payload states for full-state feedback control, vision-based state estimation is implemented to provide estimates of the suspended payload states. An image feed is received from a downward facing camera, located at the bottom of the quadrotor, which is then processed to provide swing angle measurements for the state estimator.

For the vision system, the following assumptions are made about the visual properties of the suspended payload:

- The payload can have any arbitrary shape.
- The size of the payload is unknown.
- The colour of the payload is unknown.

It is therefore required that the vision system should be able to detect and track any arbitrary payload that is attached to the quadrotor. This results in a state estimation pipeline as illustrated in Figure 6.1.

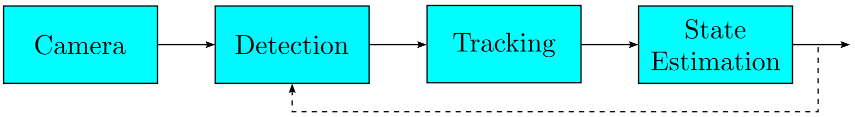


Figure 6.1: Payload state estimation pipeline.

As the visual properties of the payload is unknown, the image feed from the camera is first processed by the payload detection algorithm to identify the payload. After the payload is detected, the tracking algorithm is initialized with the visual information gained from the detection algorithm. The tracking algorithm now processes the image feed and provides the state estimator with payload swing angle measurements. Lastly, the payload state estimates are used for full-state feedback control. However, the state estimates are also used to aid the detection algorithm in the case where the vision system has lost the payload.

From the quadrotor and payload model derived in Section 3.4 it is clear that the payload swing angles, $\theta_p^{\mathcal{I}}$ and $\phi_p^{\mathcal{I}}$, are dependent on the cable length, l, the payload mass, m_p, the damping at the cable attachment related to the damping coefficient, c, and the aerodynamic drag on the payload represented by C_{D_p}. Therefore, these unknown parameters need to be estimated first as it is required for the model based state estimator.

The damping and aerodynamic drag coefficients are difficult to estimate. However, as the damping at the attachment point and the payload aerodynamic drag provide extra damping to the system, these parameters do not need to be taken into account. The payload mass, m_p, is estimated by determining the extra thrust required for the quadrotor to hover and the cable length, l, is estimated by observing the frequency of the payload oscillations.

A system identification phase is implemented to initialize the vision system and estimate the payload parameters, l and m_p, before the EKF is executed to provide the full-state feedback controller with the payload state estimates. The system identification steps are:

1. Command an altitude reference and estimate m_p.
2. Command a small position step to induce payload oscillations.
3. Detect the swinging payload using the detection algorithm.
4. Repeat steps 2 and 3 until the payload is detected.
5. Initialize the payload tracking algorithm.
6. Command a small position step to induce payload oscillations.
7. Observe the payload oscillation frequency and estimate l.
8. Begin the execution of the EKF to estimate the suspended payload states.

After the unknown parameters are estimated and estimates of the payload states are produced by the EKF, the full-state feedback controller is activated to control the horizontal velocity of the quadrotor to attenuate the suspended payload oscillations.

6.2. Vision System

The downward facing camera attached to the bottom of the quadrotor gives a top-down view of the suspended payload. The projected two-dimensional (2D) position of the payload in the image frame is described by the pixel coordinates determined by the vision system. The OpenCV computer vision library is used for the image processing. Using the pinhole camera model, the relationship between the pixel coordinates and the three-dimensional (3D) position of the payload in the camera frame can be described.

6.2.1. Camera Model

The pinhole camera model is a basic camera model used in various computer vision problems to describe the relationship between a point in a 3D space and its 2D projection onto an image frame. The pinhole camera model is very basic and does not account for effects such as geometric distortions or image blur, therefore, the validity of the model is dependent on the quality of the camera.

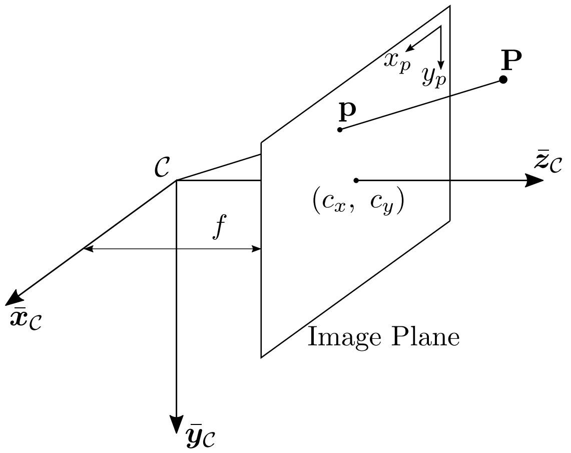

x_p
y_p
$\mathbf{P}$
$\mathbf{p}$
$\mathcal{C}$
$\bar{\boldsymbol{z}}_\mathcal{C}$
$(c_x,\ c_y)$
f
$\bar{\boldsymbol{x}}_\mathcal{C}$
Image Plane
$\bar{\boldsymbol{y}}_\mathcal{C}$

Figure 6.2: The pinhole camera model.

The pinhole camera model is illustrated in Figure 6.2, where $\mathcal{C} = \{\bar{\boldsymbol{x}}_\mathcal{C},\ \bar{\boldsymbol{y}}_\mathcal{C},\ \bar{\boldsymbol{z}}_\mathcal{C}\}$ is the camera coordinate frame with the origin located at the camera center. Point **P**, is the 3D point that is projected onto the image plane at point **p**. The position of point **p** in the image plane is described in the pixel coordinate frame, $\{x_p,\ y_p\}$. The focal length, f, is the distance between the camera center and the image plane, where the image plane is perpendicular to the principle axis, $\bar{\boldsymbol{z}}_\mathcal{C}$ and intersects at the principle point $(c_x,\ c_y)$. The principle point ideally coincides with the center of the image plane.

Consider the position of point **P** described in the camera coordinate frame as

$$\mathbf{P} = \begin{bmatrix} x_{\mathcal{C}_P} & y_{\mathcal{C}_P} & z_{\mathcal{C}_P} \end{bmatrix}, \tag{6.1}$$

then, using similar triangles, the position of the projected point **p** in the image plane is described by

$$\mathbf{p} = \begin{bmatrix} f\frac{x_{\mathcal{C}_P}}{z_{\mathcal{C}_P}} + c_x & f\frac{y_{\mathcal{C}_P}}{z_{\mathcal{C}_P}} + c_y & f \end{bmatrix}, \tag{6.2}$$

where c_x and c_y are both zero for the ideal case where the camera frame and image plane centers align.

Alternatively, the pixel coordinates of point **p** can be mapped to the 3D position of point **P** in the camera frame by converting the pixel coordinates into two angles. Considering the suspended payload, the payload pixel coordinates are converted into the angles, $\theta_p^\mathcal{C}$

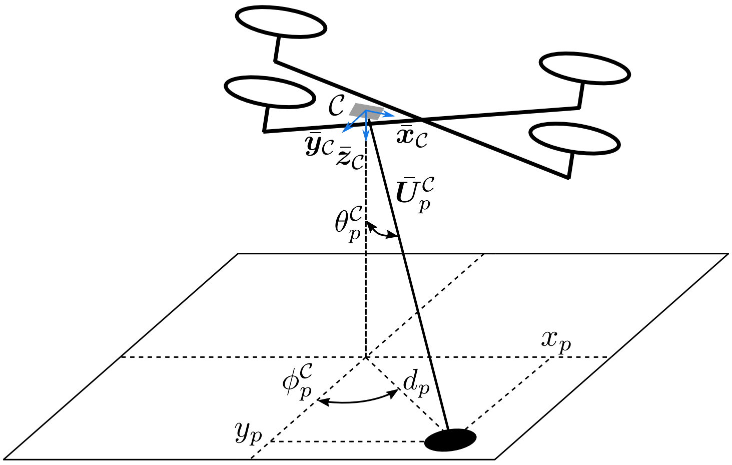

$\mathcal{C}$
$\bar{\boldsymbol{x}}_{\mathcal{C}}$
$\bar{\boldsymbol{y}}_{\mathcal{C}}$
$\bar{\boldsymbol{z}}_{\mathcal{C}}$
$\bar{\boldsymbol{U}}_p^{\mathcal{C}}$
$\theta_p^{\mathcal{C}}$
x_p
$\phi_p^{\mathcal{C}}$
d_p
y_p

Figure 6.3: Mapping of the pixel coordinates to the 3D location.

and $\phi_p^{\mathcal{C}}$, as shown in Figure 6.3. By assuming that the camera has no distortion, the pixel distance of the payload from the center of the image frame is proportional to the angle, $\theta_p^{\mathcal{C}}$, which is determined by

$$\theta_p^{\mathcal{C}} = \frac{\alpha_{HFOV}}{\eta_{HFOV}} d_p, \tag{6.3}$$

where α_{HFOV} is the horizontal field of view angle of the specific camera and η_{HFOV} is the number of pixels related to α_{HFOV}. Lastly, d_p is the distance of the payload pixel position from the image center determined by

$$d_p = \sqrt{x_p^2 + y_p^2}. \tag{6.4}$$

Angle $\phi_p^{\mathcal{C}}$ is found by the relationship between the x and y payload pixel coordinates as

$$\phi_p^{\mathcal{C}} = \arctan\left(\frac{x_p}{y_p}\right). \tag{6.5}$$

The unit vector from the camera frame to the payload is then given by

$$\bar{\boldsymbol{U}}_p^{\mathcal{C}} = \begin{bmatrix} \sin\theta_p^{\mathcal{C}} \sin\phi_p^{\mathcal{C}} \\ \sin\theta_p^{\mathcal{C}} \cos\phi_p^{\mathcal{C}} \\ \cos\theta_p^{\mathcal{C}} \end{bmatrix}. \tag{6.6}$$

6.2.2. Detection

A motion-based detection algorithm is required to detect the suspended payload. As the visual properties of the payload is unknown, feature-based detection can not be used. The motion-based detector determines the payload position in the image frame based on its motion relative to the background.

Different methods can be used for motion-based object detection. Two methods, which include frame differencing and optical flow were investigated and discussed in Section

2.4. It was decided to make use of frame differencing for the payload detection. As the camera is attached to the moving quadrotor, frame differencing with motion compensation is implemented.

The payload detection algorithm consists of the following four steps:

1. Estimate the apparent background motion and apply motion compensation using a perspective transformation.
2. Perform pixel-wise subtraction of the current image frame and transformed previous image frame and apply a threshold to obtain a binary image.
3. Apply filtering to the binary image to reduce noise.
4. Find the region of interest (ROI) containing the moving payload.

The payload detection algorithm is executed during the third step of the system identification phase. When the payload is detected, the payload ROI is used to initialize the tracking algorithm as discussed in Section 6.2.3.

Step 1: Background Motion Estimation and Compensation

The first step of the detection algorithm is to compensate for the apparent background motion so that only the motion of moving objects are visible. A homography is found that estimates the camera motion between frames which can then be used to apply a perspective transformation on the image frame to compensate for the relative motion. The homography is a 3 × 3 matrix denoted by $\boldsymbol{H}$ that maps a point with pixel position, $[x_i,\ y_i]$, in the previous frame to a pixel position, $[X_i,\ Y_i]$, so that it matches with the corresponding point in the current frame. The transformation is given by

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}. \tag{6.7}$$

In order to determine the homography, a set of at least four matching points between the current and previous image frames are required. The ORB (Oriented FAST and Rotated BRIEF) feature detector [49] is used to find features with corresponding descriptors in the current and previous image frames. Alternative feature detectors that can also be used include the SIFT [50] and SURF [51] feature detectors.

Features are significant areas in the image such as corners that are stable under transformations. Descriptors are vectors describing the area around a feature so that features can be compared. A matching algorithm is used to find the matching features in the two image frames by comparing the feature descriptors of the two frames. The matching algorithm calculates the Hamming distance between descriptors which serves as a measure of the similarity between the descriptors. The matches are then sorted according to their quality so that only good matches are used to find the homography.

Figure 6.4 shows the matched features between two image frames obtained during a practical flight test of a quadrotor and suspended payload with a downward facing camera.

It is evident that some features were incorrectly matched. It should also be noted that there are matches for the payload features. Using these matches to find the homography will result in an incorrect transformation. Therefore, a robust estimation technique, called Random Sample Consensus (RANSAC) [52], is used to accurately estimate the homography. In most cases more than half of the matches are of background features and RANSAC then identifies incorrect and payload matches as outliers which are not used for the homography estimation. In the case where the background is uniform with few features, only a small number of good matches will be identified. The risk then exists that more than half of the matches may be of the payload features resulting in an incorrect perspective transformation. Therefore, in case only a few matches are found, perspective transformation is not applied.

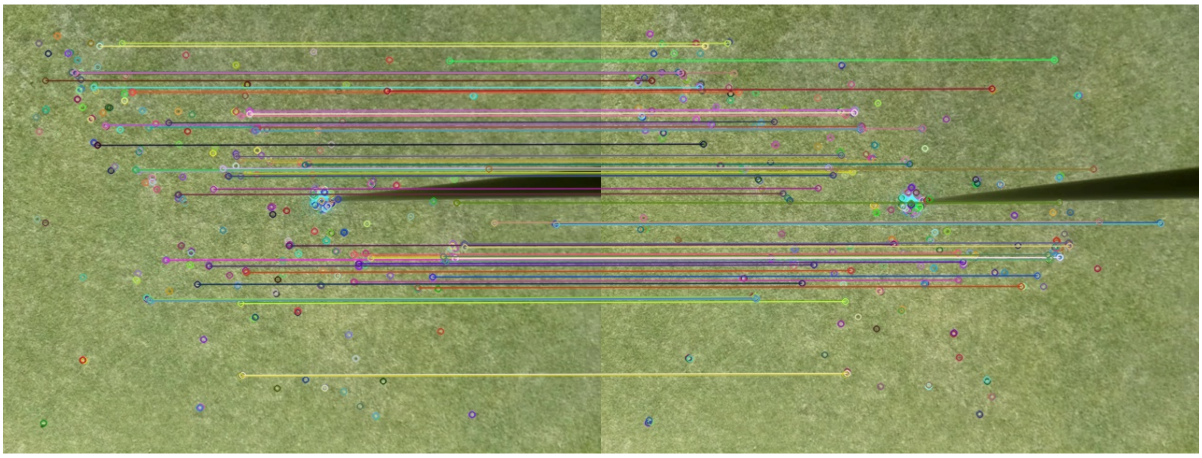


Figure 6.4: Matched features found for consecutive image frames.

Step 2: Frame Differencing

The second step is the pixel-wise subtraction of the current image frame and the transformed previous frame. The difference between the frames reveals the moving objects. Thresholding is applied to obtain a binary image where the white areas indicate moving objects and the black area form part of the static background. Examples of the thresholded inter-frame differences before and after motion compensation are shown in Figure 6.5 and Figure 6.6, respectively.

Step 3: Noise Filtering

The resulting binary image obtained from the frame differencing may still contain some noise as indicated in Figure 6.6. The noise may be seen as moving objects and are therefore filtered by applying morphological operations to the binary image.

A technique called erosion is applied to remove the noise. Erosion is a basic morphological operation where a structuring element, also known as a kernel, is superimposed on top of a subset of pixels of a binary image. The structuring element is compared to the subset of pixels for every pixel in the image frame. If every pixel in the subset for a specific pixel is part of the foreground (white), then that specific pixel stays as is, otherwise the pixel is

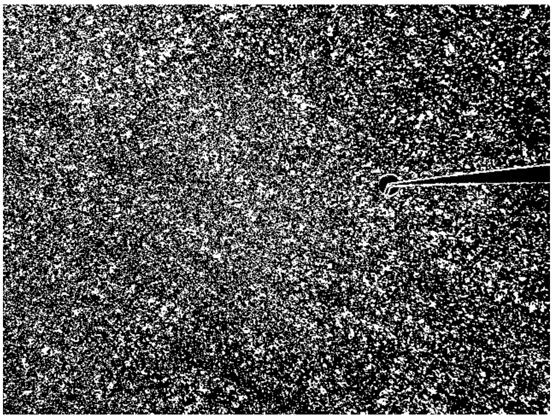


Figure 6.5: Frame Differencing before motion compensation.

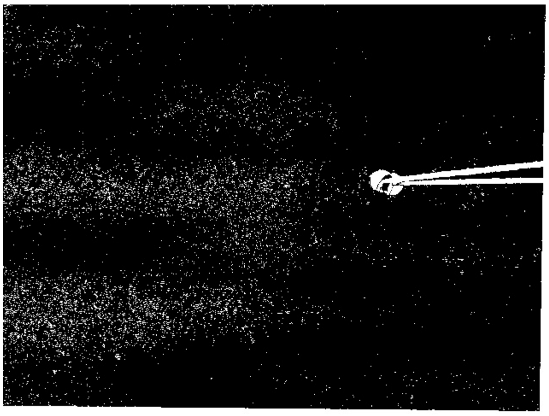


Figure 6.6: Frame Differencing after motion compensation.

changed to a background (black) pixel. Erosion is applied to the image frame in Figure 6.6 and the resulting frame is shown in Figure 6.7.

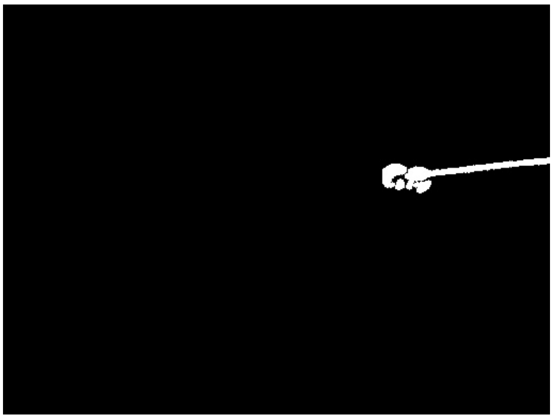


Figure 6.7: The image frame in Figure 6.6 after erosion is applied.

Step 4: Payload Region of Interest Identification

The individual white areas in the binary image of Figure 6.7 are referred to as contours and only some of the contours form part of the payload, while the rest are part of the payload cable. This step involves the initial search of a contour that most likely form part of the payload, followed by a search to find all the neigbouring payload contours which form the ROI containing the payload.

To simplify the problem, the assumption is made that the location of the payload attachment point relative to the camera is known and, therefore, the position of the payload contours relative to the cable contours are known. It is then assumed that the leftmost contour will form part of the payload. However, residual noise may still be present after step 2. Therefore, the previous location of the detected payload is also taken into account when searching for the payload contour.

A region growing algorithm is implemented to identify neigbouring contours that also form part of the payload as one contour may not always represent the whole payload. The algorithm determines if a contour forms part of the payload based on its position relative to the intially found contour as well as considering the historic ROI sizes. The output of this step is a ROI containing the suspended payload as illustrated in Figure 6.8.

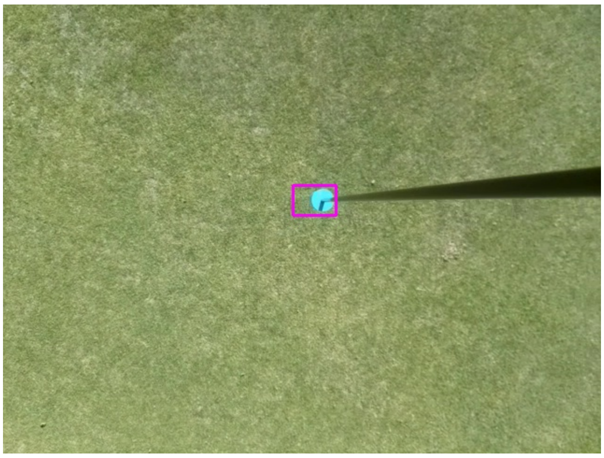


Figure 6.8: The output of the detection algorithm indicating the detected payload.

If the relative motion of the payload and background is too small to be detected, or no valid contour is found, the current frame is neglected and frame differencing is applied to the previous and next image frames.

Robustness

The detection algorithm is relatively robust and does well in detecting the payload during the system identification phase. However, as the algorithm is based on frame differencing, concerns arise when used in highly dynamic scenes or when the relative motion between the payload and background is small. The algorithm fails to detect the payload when it does not move. Flying in highly dynamic scenes will result in contours belonging to several moving objects to be visible. This may result in false detections when other moving objects are identified as the payload. Lastly, the differences between perspectives of 3D objects in consecutive image frames may be seen as moving objects when applying a perspective transformation. However, this obstacle can be overcome by increasing the camera frame rate or flying at higher altitudes to lower the perspective changes.

After the payload is detected, the vision system switches to the more robust tracking algorithm discussed in Section 6.2.3. The system switches back to the detection algorithm if the payload is lost.

6.2.3. Tracking

The Discriminative Correlation Filter with Channel and Spatial Reliability (CSRT) tracker [53] is used to track the suspended payload after it has been detected. The OpenCV computer vision library features an object tracking application programming interface (API) which includes an implementation of the CSRT tracker. The CSRT tracker is a discriminative correlation filter (DCF) tracker and uses discriminative methods to train filters to track a pre-defined object. These trackers are usually used together with object detectors.

A correlation filter based tracker tracks an object by correlating a filter over a ROI that is centered around the position of the object in the previous frame [54]. The maximum value in the filter output indicates the new location of the object. The object localization is then followed by an online update of the filter based on the new appearance of the object. The correlation is calculated using the Fast Fourier Transform (FFT) for faster tracking. The 2D Fourier transforms of the input image, F, and filter, H, are determined and used to calculate the correlation, G, as

$$G = F \odot H^*, \tag{6.8}$$

where $\odot$ represents the element-wise multiplication and $*$ indicates the complex conjugate.

Equation (6.8) represents the correlation for a basic single channel correlation filter. However, the CSRT tracker uses multiple channels for its correlation filters. The channels include histogram of oriented gradients (HoG) and Colournames. Equation (6.8) then becomes

$$G = \sum_{d=1}^{N_c} F_d \odot H_d^*, \tag{6.9}$$

where F_d is the set of N_c channels, H_d is the corresponding filter templates and G is the sum of the correlations of the individual channels. Considering the j images of the object, the correlation filters are optimized by minimizing the cost funtion:

$$J(H_d^*) = \sum_{i=1}^{j} \left| F_{d_i} \odot H_d^* - G_{d_i} \right|^2 . \tag{6.10}$$

Furthermore, to increase robustness, the CSRT tracker introduces channel reliability weights, ω_d, which are determined based on the discriminative power of each channel, to reduce noise in the filter responses during object localization.

A spatial reliability map, m, is introduced to constrain the correlation filter training. The spatial reliability map is estimated by identifying the pixels in the object bounding box which likely belongs to the object itself. These pixels are the most reliable and are used to solve the optimization problem in Equation (6.10). This improves the tracker performance for nonrectangular shaped objects.

The CSRT tracker is a state of the art tracker and is implemented in the vision system. The tracker is initialized with the ROI containing the payload which is received from the detection algrorithm. The tracker is able to accurately track the suspended payload. When the tracking algorithm loses the payload as a result of aggressive maneuvers or the payload swinging outside the camera FOV, the vision system switches back to the detection algorithm and the tracker is reinitialized with a new payload ROI.

6.3. Payload Mass Estimation

The payload mass is estimated by applying the recursive least squares (RLS) algorithm to the vertical dynamics of the quadrotor and payload system and is based on work done by [55]. By observing the extra thrust required for the quadrotor to hover with the attached payload, the payload mass can be estimated. It is assumed that the quadrotor mass is known.

The quadrotor and payload vertical dynamics, as derived in Section 3.4.2, is linearized by applying small angle approximations and ignoring aerodynamic drag in the vertical direction. The linearized vertical dynamics is then given by

$$\dot{V}_D = g + F_D \left(\frac{1}{m_q + m_p} \right). \tag{6.11}$$

To obtain better payload mass estimates, the Down velocity, V_D, is used instead of the Down acceleration, $\dot{V}_D$, as the Down velocity are estimates filtered by the PX4 EKF instead of raw Down acceleration measurements received from the onboard IMU. Equation (6.11) can then be written in the form

$$y = \boldsymbol{x}^T \boldsymbol{\beta}, \tag{6.12}$$

where

$$y = h(s) s V_D, \tag{6.13}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & \frac{1}{m_q + m_p} \end{bmatrix}^T \text{ and} \tag{6.14}$$

$$\boldsymbol{\beta} = h(s) \begin{bmatrix} g & F_D \end{bmatrix}^T. \tag{6.15}$$

The system output is represented by y, $\boldsymbol{x}$ is the parameter vector to be estimated and $\boldsymbol{\beta}$ is the measurement. The filter, $h(s)$, is implemented to prevent the amplification of noise due to the derivative term in y and is chosen to be faster than the actuator dynamics. The filter is then given by

$$h(s) = \frac{30}{s + 30}. \tag{6.16}$$

Equation (6.12) is discretized for the discrete implementation of the RLS algorithm. The RLS algorithm tries to estimate the parameters, $\boldsymbol{x}$, recursively in time to minimize the cost function

$$J(k) = \sum_{i=1}^{k} \lambda^{k-i} \epsilon^2(i) \tag{6.17}$$

with respect to $\boldsymbol{x}$, where k is the timestep, λ is a forgetting factor and ϵ is the error term. The forgetting factor adds more weight to recent data points to help reduce the effect of payload oscillations during hover. A forgetting factor of $\lambda = 0.2$ is used.

The parameter estimates, $\hat{\boldsymbol{x}}$, are updated at each time step by

$$\hat{\boldsymbol{x}}(k) = \hat{\boldsymbol{x}}(k-1) + \boldsymbol{K}(k) \boldsymbol{\epsilon}(k), \tag{6.18}$$

where $\boldsymbol{K}(k)$ is the Kalman gain and $\boldsymbol{\epsilon}(k)$ is the error term at timestep k given by

$$\boldsymbol{K}(k) = \frac{\boldsymbol{P}(k-1)\boldsymbol{\beta}(k)}{\lambda + \boldsymbol{\beta}^T(k)\boldsymbol{P}(k-1)\boldsymbol{\beta}(k)} \quad \text{and} \tag{6.19}$$

$$\boldsymbol{\epsilon}(k) = y - \boldsymbol{x}^T\boldsymbol{\beta}. \tag{6.20}$$

The covariance matrix, $\boldsymbol{P}(k)$, is determined by

$$\boldsymbol{P}(k) = \lambda^{-1}\left[\boldsymbol{P}(k-1) - \boldsymbol{K}(k)\boldsymbol{\beta}(k)\boldsymbol{P}(k-1)\right]. \tag{6.21}$$

The estimated payload mass is then determined from the parameter estimate vector, $\hat{\boldsymbol{x}}(k)$, as

$$\hat{m}_p = \begin{cases} \dfrac{1}{\hat{x}_2(k)} - m_q, & \hat{x}_2(k) \neq 0 \\ 0, & \hat{x}_2(k) = 0 \end{cases}. \tag{6.22}$$

The RLS algorithm was implemented in simulation for 1 kg, 2 kg and 3 kg payload masses and produced the results as illustrated in Figure 6.9. It can be seen that the algorithm quickly converges to the correct payload masses. From the results, it was determined that a 3 s time window is sufficient for the payload mass estimation step during the system identification phase.

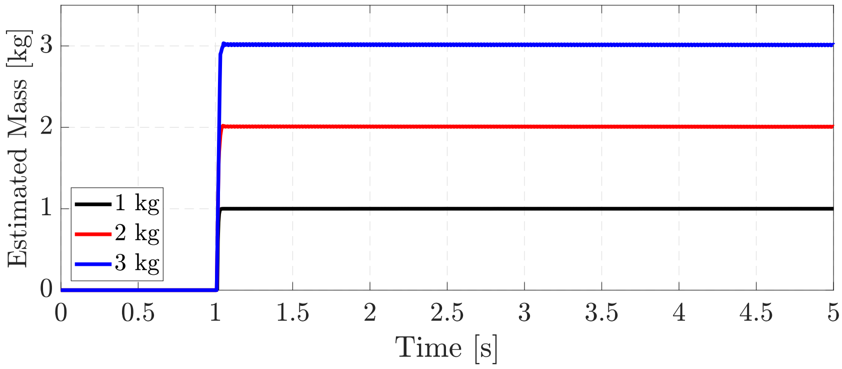


Figure 6.9: Estimates of the payload mass, m_p, using RLS.

6.4. Cable Length Estimation

The payload cable length is estimated by observing the frequency of the oscillations exhibited by the suspended payload as a result of commanding the quadrotor with a position step as stated by steps 6 and 7 of the system identification phase. The natural frequency of a quadrotor and suspended payload system is described by [25]

$$\omega_n = \sqrt{\frac{g(m_q + m_p)}{l(m_q)}}. \tag{6.23}$$

As the masses, m_q and m_p, are now known, the frequency, ω_n, needs to be determined in order to estimate the cable length, l. To determine the oscillation frequency, the vision system is utilized as it observes the payload swing angles. The Fast Fourier Transform (FFT) of the swing angle measurements can be taken to determine the frequency after which the cable length can be calculated using Equation (6.23). However, the FFT requires a relatively high number of samples which results in the cable length estimation step requiring a greater time window. This is not ideal as the system identification phase should be completed quickly. Therefore an alternative solution is implemented.

It is assumed that the payload cable length can range from 0.5 m to 5 m, therefore, the frequency range is also known. This enables the use of a dedicated sine wave estimator [25] which requires fewer samples to search for the approximately known frequency. The estimator is a steepest ascent search on the function

$$f(\omega, \alpha) = \frac{\int_0^t \theta(t)\cos(\omega t + \alpha)dt}{\int_0^t \cos^2(\omega t + \alpha)dt}, \tag{6.24}$$

where $\theta(t)$ is the input signal containing the swing angle measurements, ω is the possible frequencies of the oscillations and α is a set of phase shifts. The function peaks when a possible frequency ω coincides with the main frequency of the input signal. The estimator is discretized for implementation and executed at a rate of 1 Hz.

In Figure 6.10, the estimator is applied to a quadrotor and payload system with cable lengths of 0.5 m, 2 m and 4 m. The estimator quickly converges after only a few seconds to estimated cable lengths of 0.45 m, 1.82 m and 4.2 m, respectively. The estimates show errors of less than 10%. These errors are acceptable and may be the result of a change in the oscillation frequency due to the quadrotor controllers.

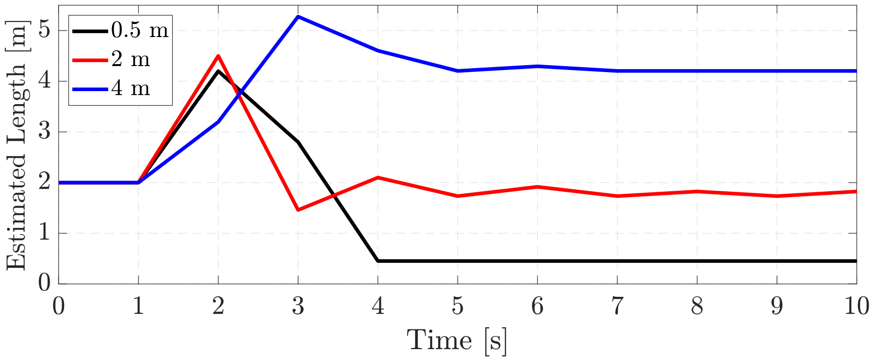


Figure 6.10: Estimates of the cable length, l, for a 2 kg payload with a 0.5 m, 2 m and 4 m cable length.

The robustness of the estimator is tested for a variety of cable lengths and payload masses and the results are summarized in Figure 6.11. It is clear that the estimator performs well for the various suspended payload systems. From the results it can be seen that an increase in the payload mass and cable length results in an increase in the estimation error. As heavier payloads and longer cable lengths result in slower systems, less swing is produced which reduces the accuracy of the estimator. A maximum error of 12.5% occurs for a 3 kg payload and 5 m cable length. However, this error is still acceptable.

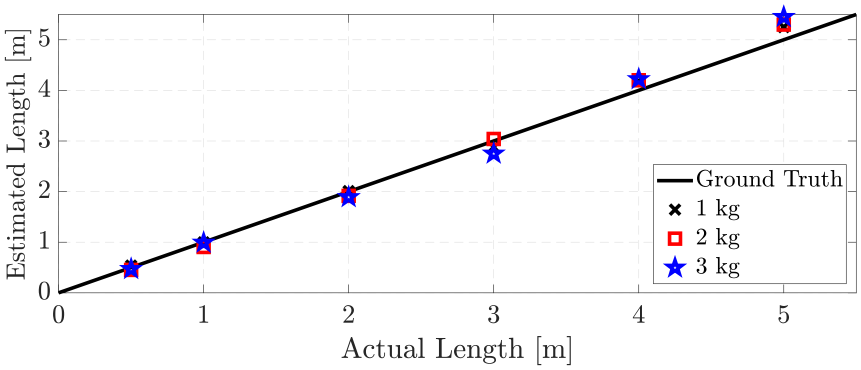


Figure 6.11: Cable length estimates for various cable lengths and payload masses.

As the estimator quickly converges, a 10 s time window is allocated for the length estimation step. With the unknown parameters of the payload estimated, the EKF can be initialized and executed to provide payload state estimates for full-state feedback control.

6.5. Payload State Estimation

An EKF is implemented to provide accurate and high rate state estimates of the suspended payload. The EKF is designed to aid in general vision-based payload transportation applications. The filter is model based and dependent on the payload parameter estimates m_p and l, therefore, the filter performance is affected by the accuracy of these estimates.

The EKF uses the quadrotor attitude estimates from the EKF of PX4, the payload swing angle measurements from the vision system as well as the commanded actuator thrusts to estimate the payload swing angles and angular rates in the inertial frame as illustrated in Figure 6.12.

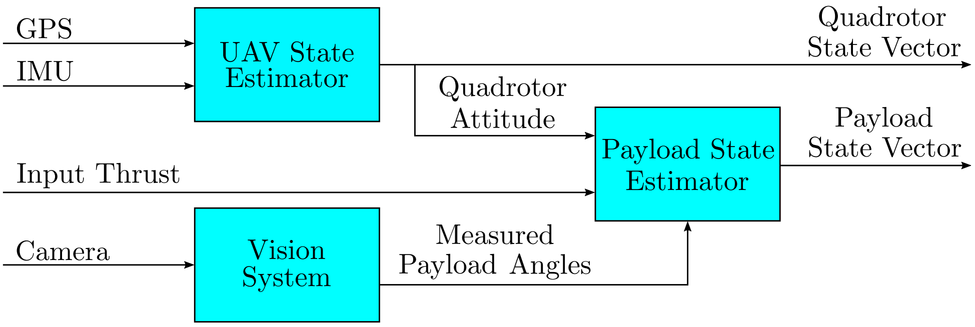


Figure 6.12: The architecture of the suspended payload state estimator.

The continuous non-linear dynamics of the suspended payload model is represented by

$$\mathbf{f}(\mathbf{X}_t,\ \mathbf{u}_t) = \begin{bmatrix} \ddot{\phi}_p^{\mathcal{I}} & \ddot{\theta}_p^{\mathcal{I}} \end{bmatrix}^T, \tag{6.25}$$

where

$$\mathbf{X}_t = \begin{bmatrix} \dot{\phi}_p^{\mathcal{I}} & \phi_p^{\mathcal{I}} & \dot{\theta}_p^{\mathcal{I}} & \theta_p^{\mathcal{I}} \end{bmatrix}^T \text{ and} \tag{6.26}$$

$$\mathbf{u}_t = \begin{bmatrix} F_N & F_E & F_D \end{bmatrix}^T . \tag{6.27}$$

The system input, $\mathbf{u}_t$, is the resulting forces in the inertial frame due to the quadrotor actuator commands. The measurement equation is given by

$$\mathbf{h}(\mathbf{X}_t) = \begin{bmatrix} \theta_p^{\mathcal{C}} & \phi_p^{\mathcal{C}} \end{bmatrix} . \tag{6.28}$$

6.5.1. Process Model

The payload swing angle dynamics derived in Section 3.4.2 is adapted and serve as the process model. As stated, the effect of aerodynamic drag on the payload and the friction at the cable attachment point is ommited as these effects will be different for different suspended payload systems and difficult to estimate accurately. The dynamic equations are then given by

$$\ddot{\phi}_p^{\mathcal{I}} = \frac{-lm_q C_\phi S_\phi \dot{\theta}_p^{\mathcal{I}^2} + F_N S_\theta S_\phi - F_E C_\phi + F_D C_\theta S_\phi}{lm_q} \quad \text{and} \tag{6.29}$$

$$\ddot{\theta}_p^{\mathcal{I}} = \frac{-F_N C_\theta + F_D S_\theta + 2\dot{\phi}_p^{\mathcal{I}} \dot{\theta}_p^{\mathcal{I}} lm_q S_\phi}{lm_q C_\phi}, \tag{6.30}$$

where $S_0 = \sin(\ )$ and $C_0 = \cos(\ )$.

6.5.2. Measurement Model

The purpose of the measurement model is to predict measurements from the predicted states. Therefore, the measurement model is found to transform the predicted swing angle states, $\begin{bmatrix} \hat{\phi}_p^{\mathcal{I}} & \hat{\theta}_p^{\mathcal{I}} \end{bmatrix}$, to predicted measurements, $\begin{bmatrix} \hat{\theta}_p^{\mathcal{C}} & \hat{\phi}_p^{\mathcal{C}} \end{bmatrix}$.

The measurements received from the vision system are in the form of two angles which are defined as illustrated in Figure 6.3 where the measured unit vector, $\bar{\boldsymbol{U}}_p^{\mathcal{C}}$, pointing from the camera to the payload is defined by Equation (6.6).

The position of the payload in the inertial coordinate frame is given by Equation (3.29) and restated here as

$$\boldsymbol{P}_p^{\mathcal{I}} = \boldsymbol{P}_q^{\mathcal{I}} + \hat{\boldsymbol{P}}_p^{A} \tag{6.31}$$

where $\hat{\boldsymbol{P}}_p^{A}$ is the position of the payload relative to the attachment point on the quadrotor, which is also the origin of the body coordinate frame, and is determined by

$$\hat{\boldsymbol{P}}_p^{A} = l \begin{bmatrix} \cos\phi_p^{\mathcal{I}} \sin\theta_p^{\mathcal{I}} \\ \sin\phi_p^{\mathcal{I}} \\ \cos\phi_p^{\mathcal{I}} \cos\theta_p^{\mathcal{I}} \end{bmatrix} . \tag{6.32}$$

The unit vector pointing from the camera to the payload is found by rotating the relative payload position to the body frame, offsetting it to the camera frame and normalizing:

$$\hat{\boldsymbol{U}}_p^{\mathcal{C}} = \frac{\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\hat{\boldsymbol{P}}_p^{A} + \boldsymbol{P}_{\mathcal{B}}^{\mathcal{C}}}{|\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\hat{\boldsymbol{P}}_p^{A} + \boldsymbol{P}_{\mathcal{B}}^{\mathcal{C}}|}, \tag{6.33}$$

where $\boldsymbol{P}_{\mathcal{B}}^{\mathcal{C}}$ is the position of the body frame in the camera frame. As the camera is fixed to the quadrotor, the camera frame has the same rotation as the body frame. The predicted angle measurements is then found by using Equation (6.6).

6.5.3. EKF Algorithm

The EKF consists of two recursive steps, namely the control update and measurement update steps. The steps are as follows:

1. Control Update:

   (a) Propagate the state vector using Euler integration:

   $$\hat{\mathbf{X}}_k^- = \hat{\mathbf{X}}_{k-1}^+ + T_s\mathbf{f}\left(\hat{\mathbf{X}}_{k-1}^+,\ \mathbf{u}_{k-1}\right), \tag{6.34}$$

   where $\hat{\mathbf{X}}_{k-1}^+$ is the states of the previous timestep and T_s is the EKF execution rate.

   (b) Update the covariance matrix:

   $$\mathbf{P}_k^- = \mathbf{F}_k\mathbf{P}_{k-1}^+\mathbf{F}_k^T + \mathbf{Q}_k, \tag{6.35}$$

   where $\mathbf{P}_{k-1}^+$ is the covariance of the previous timestep, $\mathbf{F}_k$ is the discrete Jacobian matrix and $\mathbf{Q}_k$ is the process noise covariance matrix chosen for the system. The discrete Jacobian matrix is calculated using a Taylor series expansion as

   $$\mathbf{F}_k = \mathbf{I} + \mathbf{F}_t T_s, \tag{6.36}$$

   where $\mathbf{I}$ is an identity matrix and $\mathbf{F}_t$ is the Jacobian matrix of the continuous system defined as:

   $$\mathbf{F}_t = \left.\frac{\partial\mathbf{f}\left(\mathbf{x}_t,\ \mathbf{u}_{(k-1)T_s}\right)}{\partial\mathbf{X}_t}\right|_{\mathbf{x}_t=\hat{\mathbf{X}}_{k-1}^+}. \tag{6.37}$$

2. Measurement Update:

   (a) Determine the Kalman gain:

   $$\mathbf{L}_k = \mathbf{P}_k^-\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1}, \tag{6.38}$$

   where $\mathbf{R}_k$ is the measurement noise covariance matrix, chosen according to the noise of the vision system measurements and $\mathbf{H}_k$ is the discrete Jacobian matrix calculated by

$$\mathbf{H}_k = \mathbf{H}_t = \left.\frac{\partial \mathbf{h}\left(\mathbf{x}_t\right)}{\partial \mathbf{X}_t}\right|_{\mathbf{x}_t = \hat{\mathbf{X}}_k^-} . \tag{6.39}$$

(b) Update the state vector using the angle measurements from the vision system:

$$\hat{\mathbf{X}}_k^+ = \hat{\mathbf{X}}_k^- + \mathbf{L}_k \left[\mathbf{Y}_k - \mathbf{h}\left(\hat{\mathbf{X}}_k^-\right)\right], \tag{6.40}$$

where $\mathbf{Y}_k$ is the angle measurements.

(c) Update the covariance matrix:

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{L}_k \mathbf{H}_k\right) \mathbf{P}_k^- . \tag{6.41}$$

6.5.4. Estimator Verification

The estimator design is verified in simulation where the state estimates are compared with the true states. Figures 6.13 and 6.14 show the payload swing angle, $\theta_p^{\mathcal{I}}$, and angular rate, $\dot{\theta}_p^{\mathcal{I}}$, when a 1 m position step in the North direction is commanded for 2 kg payload and 1 m cable.

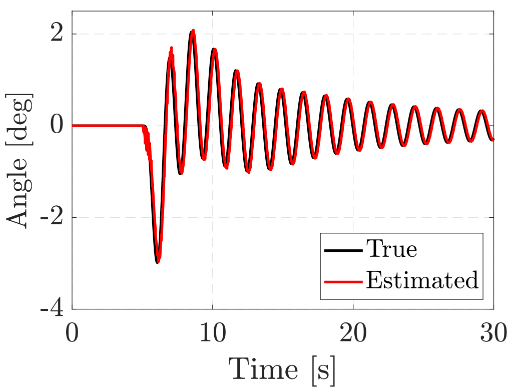


Figure 6.13: EKF estimate of the payload swing angle, $\theta_p^{\mathcal{I}}$.

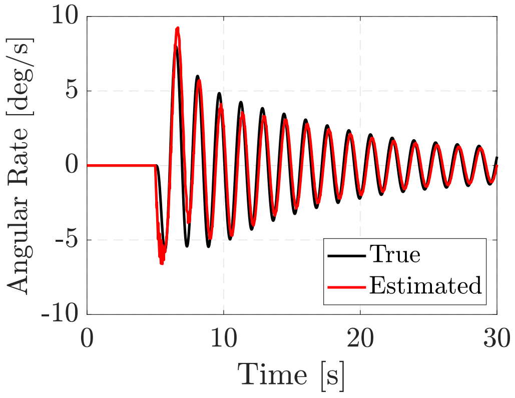


Figure 6.14: EKF estimate of the payload swing rate, $\dot{\theta}_p^{\mathcal{I}}$.

Commanding an East position step of 1 m shows similar results for the swing angle, $\phi_p^{\mathcal{I}}$, and angular rate, $\dot{\phi}_p^{\mathcal{I}}$ and are not presented here. The estimated states follow the true states very well. A slight delay is visible between the estimated and true states which is due to the delay of the measurements received as a result of the time required for the image processing. However, this small delay is not of concern, but may introduce problems if it increases. An increased delay may result in the full-state feedback controller amplifying the payload swing oscillations.

The EKF performance is investigated when errors in the payload mass and cable length estimates are introduced as it is dependent on the accuracy of these parameters. Figures 6.15 and 6.16 show a 1 m position step response in the North direction for a 2 kg payload attached with a 1 m cable when a 20% error is introduced in both the mass and cable

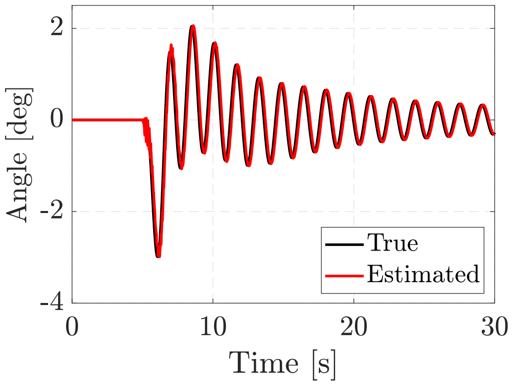


Figure 6.15: EKF estimate of the payload swing angle, $\phi_p^{\mathcal{I}}$, with a 20% parameter estimate error.

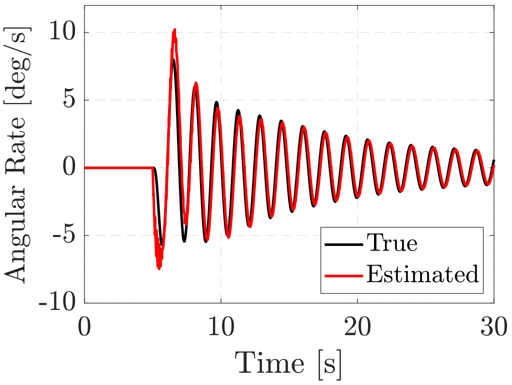


Figure 6.16: EKF estimate of the payload swing rate, $\dot{\phi}_p^{\mathcal{I}}$, with a 20% parameter estimate error.

length estimates. It is clear that the EKF is still able to accurately estimate the payload states with a 20% error in the parameter estimates. The direct swing angle measurements from the vision system reduces the effect that these errors have on the state estimates.

The performance of the EKF is also investigated in the event where no measurements are received from the vision system due to the tracking algorithm losing the payload or when it swings outside the FOV of the camera as a result of aggressive maneuvers. Figures 6.17 and 6.18 show the swing angle and swing rate estimates of a 2 kg payload with a 1 m cable length for an aggressive quadrotor maneuver. Figure 6.17 indicates when measurements are received and it can be seen that a period of 2 s passes near time $t = 60\ s$ where no measurements are received. It is evident that the EKF is able to accurately propagate the payload states. However, it should be noted that the phase delay increases during the time when no new measurements are received which is then quickly corrected with new measurements.

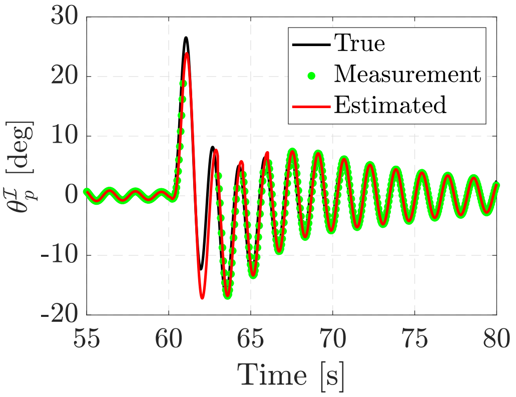


Figure 6.17: EKF estimate of the payload swing angle, $\theta_p^{\mathcal{I}}$, for aggressive maneuvers.

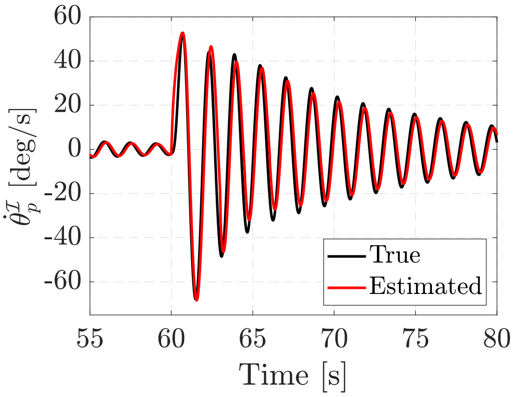


Figure 6.18: EKF estimate of the payload swing rate, $\dot{\theta}_p^{\mathcal{I}}$, for aggressive maneuvers.

6.6. Simulation Results

The full system identification phase and payload state estimation were verified in a SITL simulation in the Gazebo simulation environment for the quadrotor with a 2 kg payload and 1 m cable. High-frequency sensor noise, low-frequency sensor drift and sensor biases were added to the quadrotor model to replicate real-world scenarios. The results are presented in the order of the system identification steps.

The quadrotor was given an altitude reference and when reached, the payload mass was estimated with the result shown in Figure 6.19. The payload mass was estimated to be 1.98 kg. Thereafter, a gentle position step was commanded for the payload detection which initialized the payload tracking algorithm. The detected payload is shown in Figure 6.20. This is followed by a second position step command for the cable length estimation. The cable length, as shown in Figure 6.21, was estimated to be 0.95 m. The results indicate that the mass and cable length estimators are robust against sensor noise and are still able to produce accurate estimates.

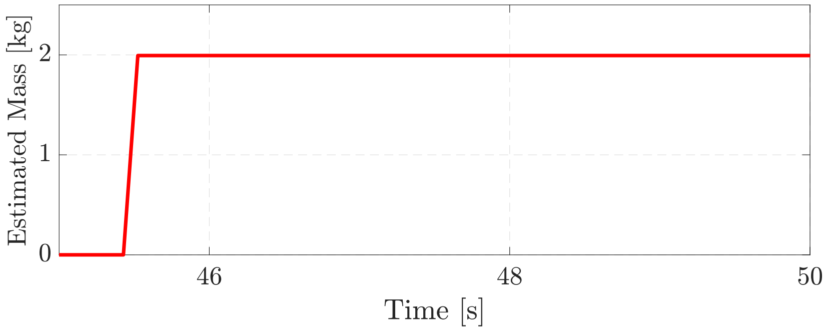


Figure 6.19: The payload mass estimation during the system identification phase.

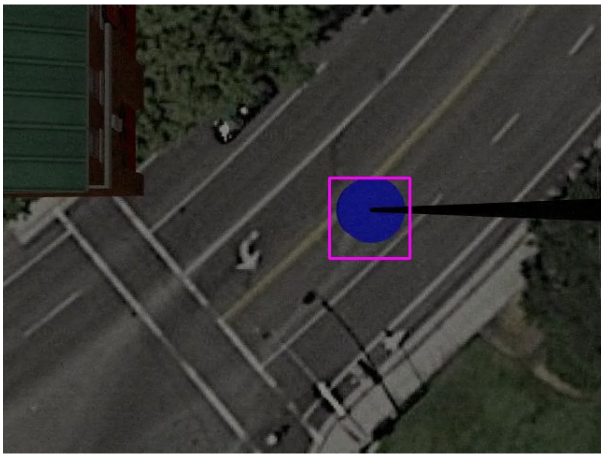


Figure 6.20: The detected payload.

After the payload parameters were estimated, the EKF was initialized and estimates of the payload swing angles and swing rates were produced as shown in Figures 6.22 to 6.25. It is clear that the estimates follow the true states well. The EKF is known to be robust against sensor noise as it is taken into account when producing estimates.

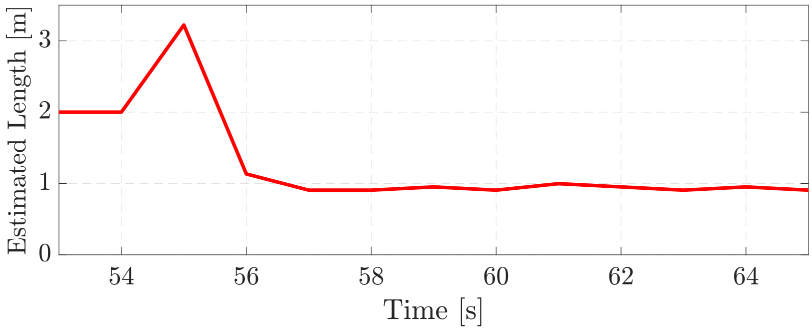


Figure 6.21: The cable length estimation during the system identification phase.

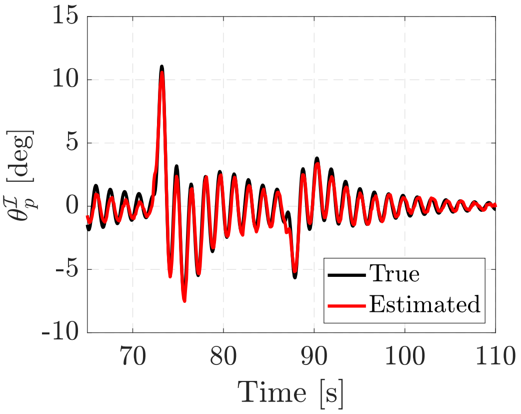


Figure 6.22: The $\theta_p^{\mathcal{I}}$ swing angle estimate.

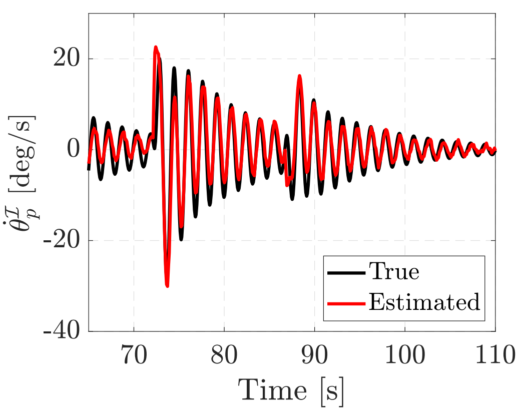


Figure 6.23: The $\dot{\theta}_p^{\mathcal{I}}$ swing rate estimate.

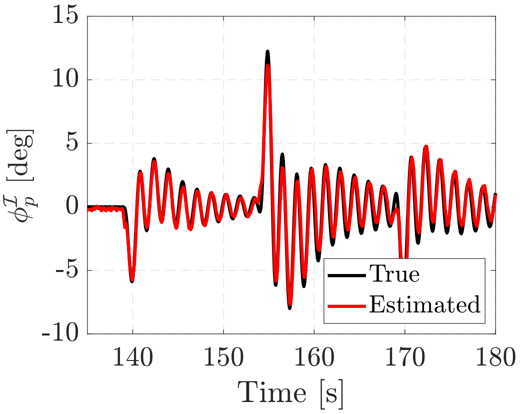


Figure 6.24: The $\phi_p^{\mathcal{I}}$ swing angle estimate.

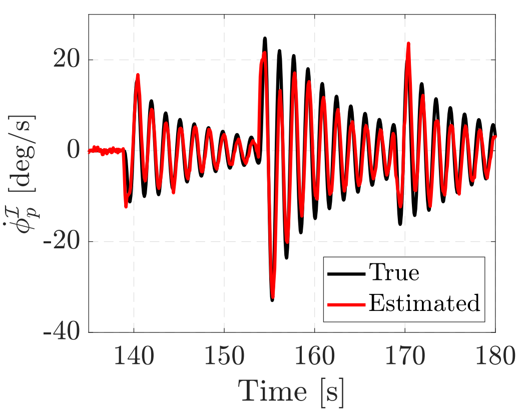


Figure 6.25: The $\dot{\phi}_p^{\mathcal{I}}$ swing rate estimate.

6.7. Summary

This chapter discussed the payload parameter estimation process, the vision system implemented to measure the payload swing angles and the estimator implemented to provide accurate payload state estimates for full-state feedback control.

The payload parameter estimation algorithms, which include the RLS algorithm for mass estimation and the sine wave estimator for cable length estimation, proved to be robust against sensor noise and produced estimates with small errors.

The vision system is able to detect and track a payload with unknown visual properties. Therefore, any arbitrary payload can be transported without the need for visual markers. The vision-based state estimation also eliminates the need for sensors on the payload.

The performance of the EKF was tested. The EKF proved to be robust against parameter estimation errors and was able to provide accurate payload state estimates. The EKF also showed the ability to accurately propagate state estimates during times when no measurements were received as a result of the vision system losing the payload when it swings outside the camera FOV. However, a phase delay becomes visible during these times. Therefore, long periods of time without measurements need to be avoided by using a camera with a greater FOV or avoiding aggressive maneuvers.

Chapter 7

Payload Swing Damping Control

This chapter discusses the velocity control system design for a quadrotor with an unknown suspended payload. As concluded in Chapter 5, the standard PID velocity controllers of the quadrotor are not adequate when a suspended payload is attached to the quadrotor. The payload significantly alters the dynamics of the quadrotor.

As it is assumed that the payload is attached to the center of mass of the quadrotor, only the translational dynamics are affected by the suspended payload. Therefore, the quadrotor horizontal velocity controllers will be redesigned to prevent payload oscillations and damp any residual payload oscillations still present in the system.

The chapter starts with the linearization of the quadrotor and suspended payload dynamics. This is followed by a discussion of the proposed control strategy with an architecture as shown in Figure 7.1, where r denotes the horizontal velocity input, u, the thrust control signal and y the plant output in the figure. The control strategy consists of a notch filter as a feedforward controller to minimize payload oscillations and a Linear Quadratic Gaussian (LQG) full-state feedback controller with integral action to damp residual oscillations in the system as well as externally imposed oscillations. The controllers are designed and tested in the MATLAB/Simulink environment. The designed longitudinal and lateral velocity controllers are identical and only the longitudinal velocity controller is represented.

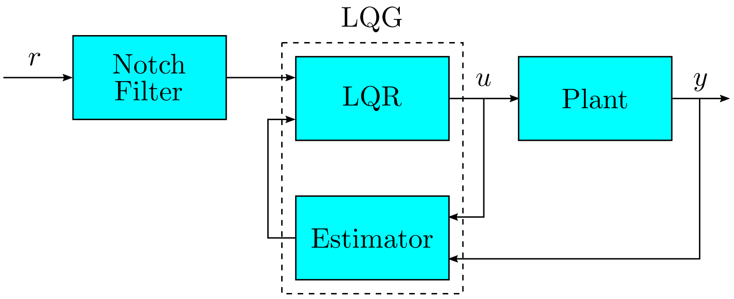


Figure 7.1: The proposed horizontal velocity control strategy.

7.1. Linearized Quadrotor and Suspended Payload Plant

The controller design is based on the linearized quadrotor and suspended payload dynamics. The equations of motion are derived with regards to the controller input and output. To ensure that steady state tracking errrors and drift caused by external disturbances are rejected, the longitudinal and lateral velocity of the quadrotor will be controlled.

The non-linear equations of motion derived in Section 3.4.2 are linearized around hover, $F_D = -g(m_q + m_p)$, by applying small angle approximations and ignoring the aerodynamic drag forces on the quadrotor and suspended payload given in Equations (3.33) and (3.34). Consider the model shown in Figure 7.2.

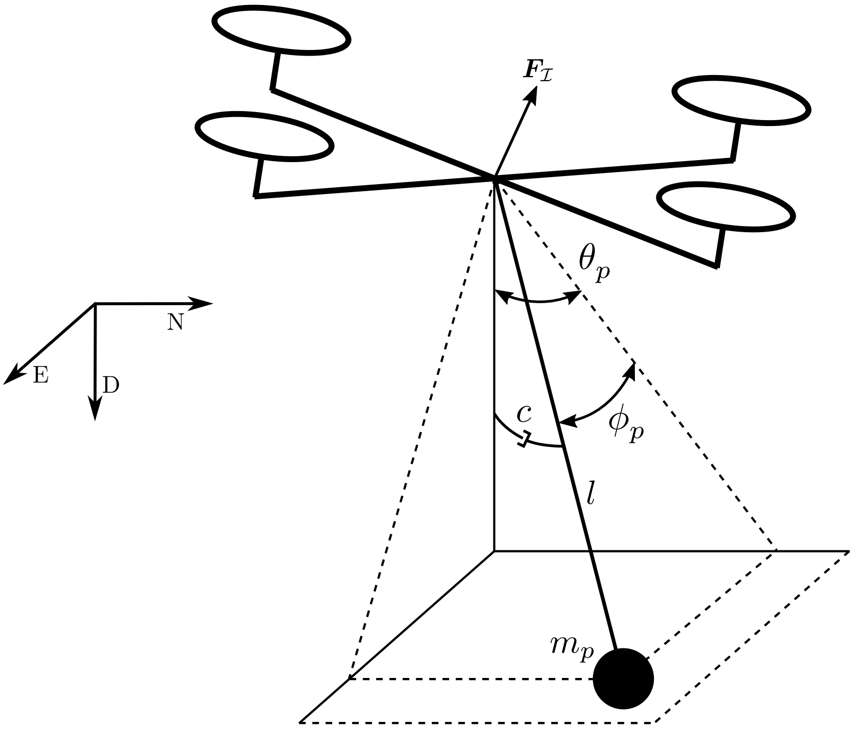


Figure 7.2: The quadrotor and suspended payload system.

The inertial force, $\boldsymbol{F}_{\mathcal{I}} = \{F_N,\ F_E,\ F_D\}$, represents the actual inertial force on the quadrotor as a result of the actuator outputs and serves as the input to the velocity controller plant. The longitudinal and lateral dynamics are decoupled and only the longitudinal dynamics are presented. Due to the quadrotor symmetry, the lateral dynamics are identical and not presented here. The linearized longitudinal equations of motion of the quadrotor and suspended payload system are written in state space form as

$$\dot{\mathbf{X}}_{long} = \mathbf{A}_{long}\mathbf{X}_{long} + \mathbf{B}_{long}\mathrm{U}_{long} \text{ and} \tag{7.1}$$

$$\mathrm{Y}_{long} = \mathbf{C}_{long}\mathbf{X}_{long}, \tag{7.2}$$

where

$$\mathbf{X}_{long} = \begin{bmatrix} V_N \; \dot{\theta}_p^{\mathcal{I}} \; \theta_p^{\mathcal{I}} \end{bmatrix}^T, \tag{7.3}$$

$$\mathbf{A}_{long} = \begin{bmatrix} 0 & \frac{c}{lm_q} & \frac{gm_p}{m_q} \\ 0 & -\frac{c(m_q+m_p)}{l^2m_qm_p} & -\frac{g(m_q+m_p)}{lm_q} \\ 0 & 1 & 0 \end{bmatrix}, \tag{7.4}$$

$$\mathbf{B}_{long} = \begin{bmatrix} \frac{1}{m_q} & -\frac{1}{lm_q} \end{bmatrix}^T, \tag{7.5}$$

$$\mathbf{C}_{long} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{7.6}$$

The input and output to the plant are given by $\mathrm{U}_{long} = F_N$ and $\mathrm{Y}_{long} = V_N$, respectively. The linear longitudinal velocity plant is then determined to be

$$G_{V_N}(s) = \frac{V_N(s)}{F_N(s)} = \mathbf{C}_{long}\left(s\mathbf{I} - \mathbf{A}_{long}\right)^{-1}\mathbf{B}_{long} \tag{7.7}$$

$$G_{V_N}(s) = \frac{1}{m_q} \cdot \frac{s^2 + \frac{c}{m_pl^2}s + \dfrac{g}{l}}{s\left[s^2 + \dfrac{c(m_q+m_p)}{m_qm_pl^2}s + \dfrac{g(m_q+m_p)}{m_ql}\right]} \tag{7.8}$$

It should be noted that the actual input to the longitudinal velocity controller plant is the commanded inertial force, $F_{N_{ref}}$, where the relationship between the actual force and commanded force is represented by the pitch angle dynamics, $G_{q2_{cl}}(s)$. However, as in [17], it is assumed that there is good time-scale separation between the inner pitch angle dynamics and outer longitudinal velocity dynamics.

Figure 7.3 shows the frequency responses of the pitch angle dynamics, the longitudinal velocity dynamics of the quadrotor and suspended payload system described by Equation (7.7) and the combined pitch angle and longitudinal velocity dynamics. The nominal payload has a mass of 2 kg and a 1 m cable length. The resonant pole and zero pairs added by the suspended payload are visible. The pole pair is at the natural frequency of the quadrotor and payload system which is given by Equation (6.23).

From Figure 7.3 it is also clear that the assumption made is reasonable as the bandwidth of the pitch angle dynamics is higher than the bandwidth of the longitudinal velocity dynamics. Therefore, the pitch angle dynamics are not considered when designing the velocity controllers.

7.2. Feedforward Control: Notch Filter

A common technique called input shaping is used for gantry cranes to prevent slung load oscillations [7]. A slung load attached to a gantry crane shows the same pendulum like behaviour as a suspended payload attached to a quadrotor. Therefore, similar techniques can be applied to the quadrotor and suspended payload system to prevent payload oscillations.

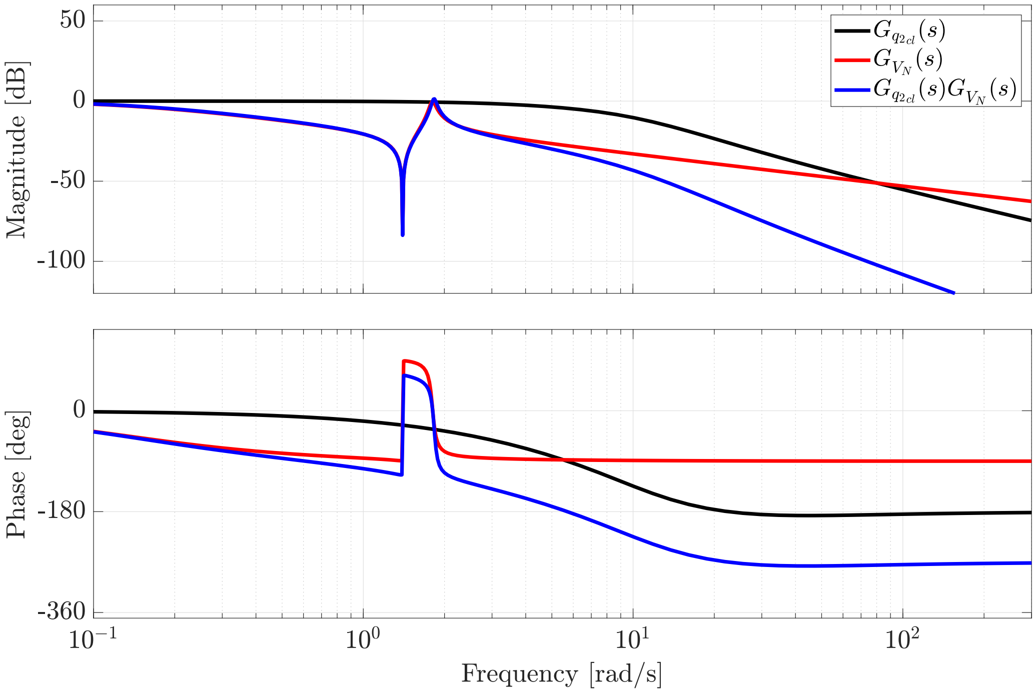


Figure 7.3: Bode plot of the pitch angle dynamics, $G_{q2_{cl}}(s)$, the longitudinal velocity dynamics of the quadrotor and payload system, $G_{V_N}(s)$, and the combined dynamics, $G_{q2_{cl}}(s)G_{V_N}(s)$.

Input shaping is a feedforward control technique that shapes a reference signal to avoid the excitation of lightly damped modes in a system. The need for input shaping is illustrated in Figure 7.4. The suspended payload introduces a lightly damped mode at the resonant pole pair in the longitudinal velocity dynamics. This results in the system to be underdamped. From Figure 7.4 it is also clear that the standard longitudinal velocity PID controller designed in Chapter 5 does not provide enough damping to reduce the effect of the lightly damped pole pair. Therefore, it is desired to implement an input shaper.

In most cases, the impulse based Zero Vibration and Derivative (ZVD) shaper is used for input shaping [56]. It involves the convolving of a command signal with a sequence of impulses to prevent the excitation of a vibration mode where the amplitudes and time locations of the impulses are dependent on the natural frequency of the system. The ZVD shaper is relatively robust to model uncertainties and therefore a popular technique to use. It is also possible to apply filters such as low pass and notch filters for input shaping. A comparison of the different input shaping techniques is given in [57].

As it is desired to specifically attenuate the lightly damped mode introduced by the suspended payload, it was decided to make use of a notch filter for input shaping. A notch filter, also known as a bandstop filter, is a filter that only attenuates a narrow band of frequencies with little effect on the rest of the system. The notch filter ensures that the specific mode is not destabilized by feedback control. As the natural frequency of the quadrotor and payload system are estimated, as discussed in Section 6.4, a notch filter can be designed online to specifically suppress the unwanted suspended payload oscillations.

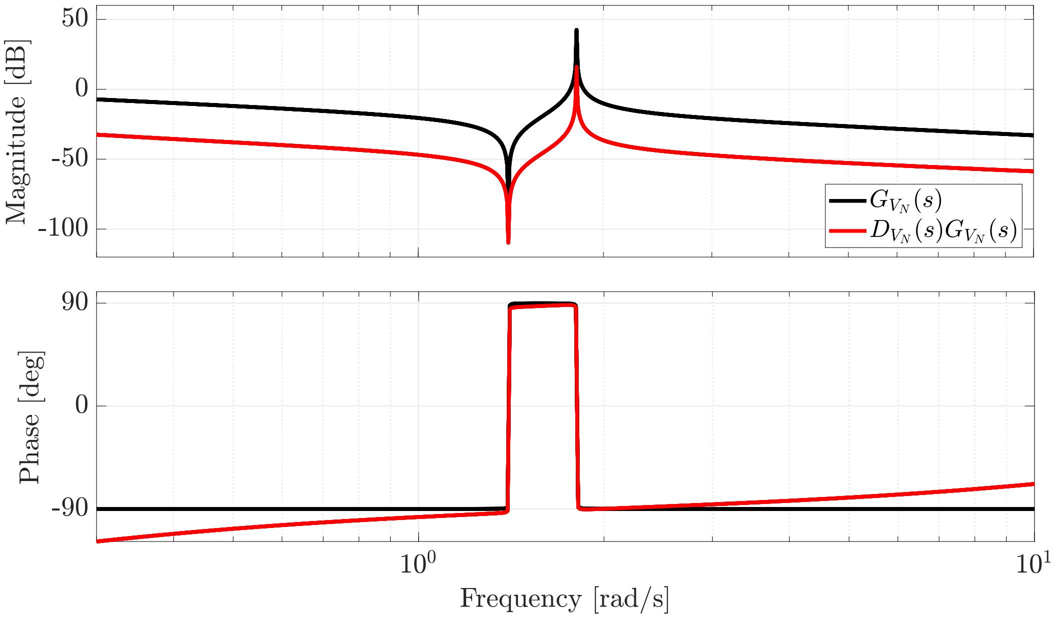


Figure 7.4: Bode plot of the open-loop longitudinal velocity dynamics, $G_{V_N}(s)$, and the combined longitudinal velocity and PID controller dynamics, $D_{V_N}(s)G_{V_N}(s)$.

The transfer function of a standard notch filter is given by

$$D_n(s) = \frac{s^2 + \omega_0^2}{s^2 + \omega_c s + \omega_0^2}, \tag{7.9}$$

where ω_0 is the central rejected frequency set equal to the natural frequency of the quadrotor and suspended payload system and ω_c is the width of the rejected band.

To accomodate for errors in the payload parameter estimates which are used to determine the natural frequency of the quadrotor and payload system, the notch filter is designed with a quality factor of $Q_n = 0.5$. The quality factor is related to the width of the rejected band as

$$\omega_c = \frac{\omega_0}{Q_n}. \tag{7.10}$$

By lowering the quality factor, the rejected frequency bandwidth is increased. This results in less attenuation at the target frequency. However, due to the possible payload parameter estimate errors, the notch filter is designed with an emphasis on robustness.

The frequency responses of the notch filter, $D_n(s)$, the open-loop longitudinal velocity and PID controller dynamics, $D_{V_N}(s)G_{V_N}(s)$ and the combined notch filter and longitudinal velocity and PID controller dynamics, $D_n(s)D_{V_N}(s)G_{V_N}(s)$, are shown in Figure 7.5. It is clear that the notch filter suppresses the control loop gain at the target natural frequency of the system.

The performance of the notch filter with the standard quadrotor PID controllers designed in Chapter 5 is investigated. Notch filtering is applied to the reference longitudinal velocity command , $V_{N_{ref}}$, of the quadrotor. The compensated and uncompensated longitudinal velocity and payload swing angle responses of the nominal quadrotor and suspended payload system for a 1 m/s longitudinal velocity step input are shown in Figure 7.6 and 7.7.

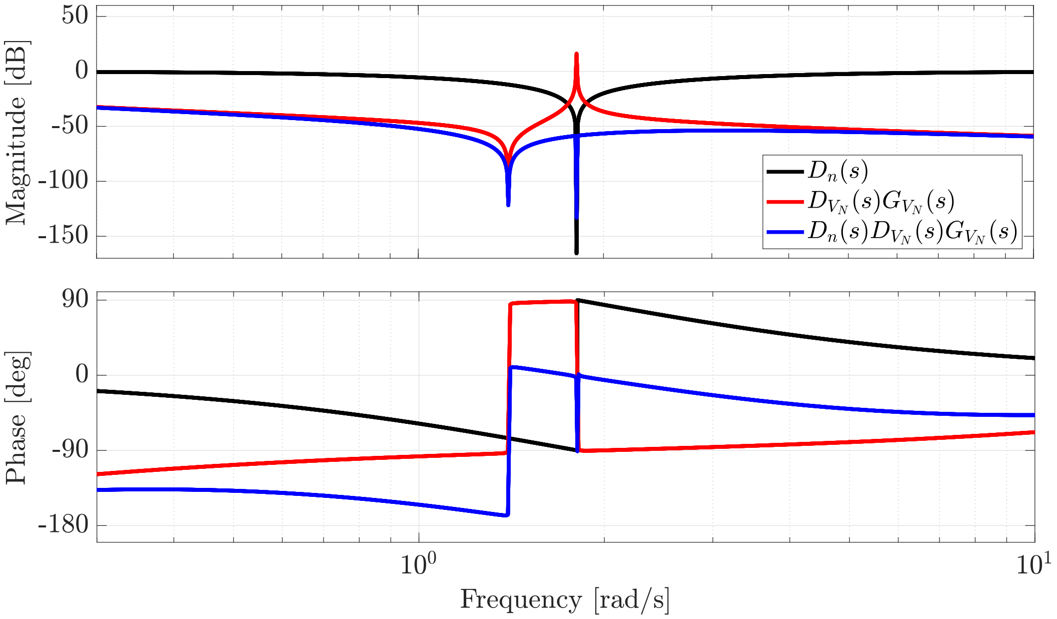


Figure 7.5: Bode plot of the open-loop longitudinal velocity dynamics and PID controller, $D_{V_N}(s)G_{V_N}(s)$, the notch filter, $D_n(s)$ and the combined system, $D_n(s)D_{V_N}(s)G_{V_N}(s)$.

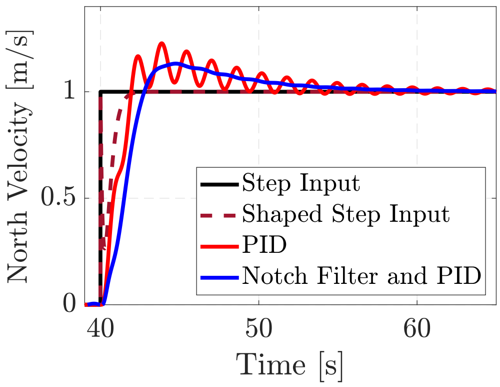


Figure 7.6: The quadrotor longitudinal velocity step response with and without notch filtering.

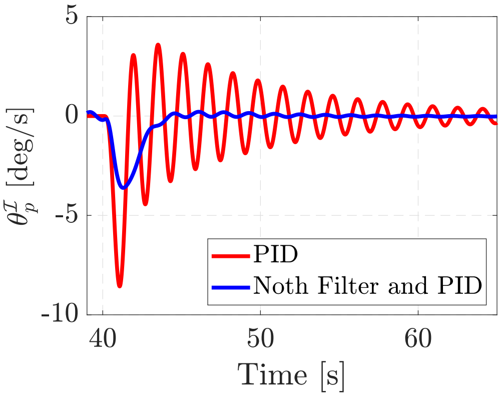


Figure 7.7: The $\theta_p^{\mathcal{I}}$ swing angle response with and without notch filtering.

The effect of the notch filter on the reference input signal is visible in Figure 7.6 and it is clear that the notch filter is able to attenuate the payload oscillations to near swing free motion and reduce the settling time.

The performance of the notch filter is dependent on the accuracy of the estimated payload parameters. Therefore, the system response for the same longitudinal velocity step input is shown in Figures 7.8 and 7.9 for 20% parameter estimate errors. The notch filter is still able to suppress the suspended payload oscillations with only a small deterioration in performance.

Lastly, it should be noted that the notch filter, and input shaping in general, does not provide any active damping, therefore, a full-state feedback controller is implemented to directly control the payload swing angles and provide active damping.

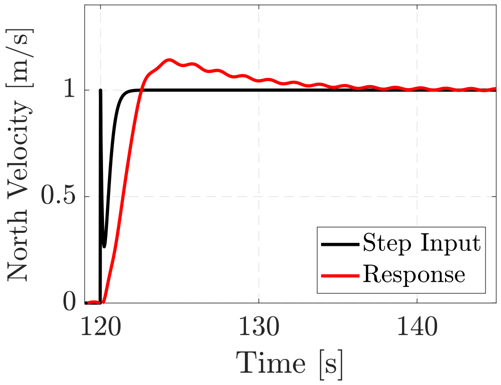


Figure 7.8: The quadrotor longitudinal velocity step response with notch filtering for payload parameter estimate errors of 20%.

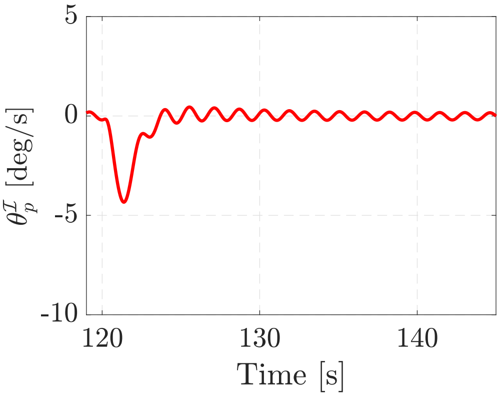


Figure 7.9: The $\theta_p^{\mathcal{I}}$ swing angle response with notch filtering for payload parameter estimate errors of 20%.

7.3. Full-State Feedback Control: LQG Control

To actively damp the suspended payload oscillations, a Linear Quadratic Gaussian (LQG) controller is implemented for full-state feedback control. An LQG controller consists of a Linear Quadratic Regulator (LQR) and an EKF. The LQR controller is implemented to simultaneously control the quadrotor longitudinal and lateral velocity as well as the suspended payload swing angles and swing rates. An EKF is required to estimate the payload states for full-state feedback control as the states cannot be directly measured. Furthermore, to ensure a zero steady-state tracking error, integral action is added to the LQR controller. The longitudinal state space equation is augmented with the additional integral state which is given as

$$\dot{V}_{N_I} = V_{N_{ref}} - V_N = V_{N_{ref}} - \mathbf{C}_{long}\mathbf{X}_{long}. \tag{7.11}$$

The augmented state space equation is then given by

$$\dot{\mathbf{X}}_A = \mathbf{A}_A\mathbf{X}_A + \mathbf{B}_A\mathrm{U}_{long} + \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} V_{N_{ref}}, \tag{7.12}$$

where

$$\mathbf{X}_A = \begin{bmatrix} V_{N_I} & \mathbf{X}_{long} \end{bmatrix}^T, \tag{7.13}$$

$$\mathbf{A}_A = \begin{bmatrix} 0 & -\mathbf{C}_{long} \\ \mathbf{0} & \mathbf{A}_{long} \end{bmatrix} \text{ and} \tag{7.14}$$

$$\mathbf{B}_A = \begin{bmatrix} 0 & \mathbf{B}_{long} \end{bmatrix}^T. \tag{7.15}$$

The resulting structure for the longitudinal velocity controller is illustrated in Figure 7.10.

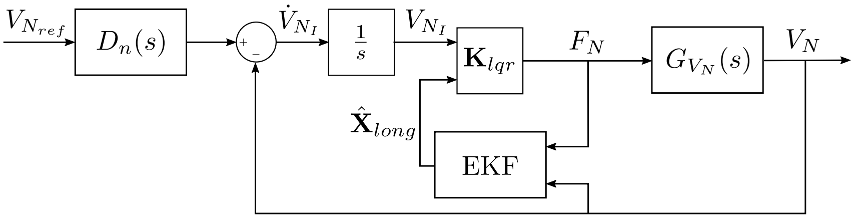


Figure 7.10: The proposed control strategy.

The control law of the LQR controller is given by

$$\mathrm{U}_{long} = -\mathbf{K}_{lqr}\mathbf{X}_A, \tag{7.16}$$

where $\mathbf{K}_{lqr}$ is the optimal control gains calculated by minimizing the cost function

$$J = \int_0^\infty \left(\mathbf{X}_A^T\mathbf{Q}\mathbf{X}_A + \mathrm{U}_{long}^2 R\right) dt. \tag{7.17}$$

$\mathbf{Q}$ is the matrix chosen to weigh the importance of the states on the the cost function and R is the scalar chosen to weigh the importance of the input on the cost function. These weights are chosen so that the controller meets the design requirements. The effect of the weights are explained as follows. Assigning a larger relative weight to R with respect to $\mathbf{Q}$ will result in the control effort to be kept small by allowing bigger deviations from the state references. Assigning larger relative weights to the elements in $\mathbf{Q}$ will result in a more aggressive controller to keep the corresponding state errors small.

The optimal gains are determined as

$$\mathbf{K}_{lqr} = R^{-1}\mathbf{B}_A\mathbf{W}, \tag{7.18}$$

where $\mathbf{W} = \mathbf{W}^T \geq 0$ is the unique positive semi-definite solution to the algebraic Riccati equation (ARE):

$$\mathbf{Q} - \mathbf{W}\mathbf{B}_A R^{-1}\mathbf{B}_A^T\mathbf{W} + \mathbf{W}\mathbf{A}_A^T + \mathbf{A}_A\mathbf{W} = \mathbf{0}, \tag{7.19}$$

and is determined via the eigendecomposition of the Hamiltonian matrix defined as

$$\mathbf{H}_A = \begin{bmatrix} \mathbf{A}_A & -\mathbf{B}_A R^{-1}\mathbf{B}^T \\ -\mathbf{Q} & -\mathbf{A}_A^T \end{bmatrix}. \tag{7.20}$$

The design requirements for the LQR controller are to damp the suspended payload oscillations and to ensure a zero steady-state tracking error. The controller is designed with an emphasis on damping and steady-state tracking performance at the expense of response time. Therefore, the state weighting matrix and input weight are chosen as follows:

- A relatively large weight is assigned to the integral state, V_{N_I}, to emphasize the steady-state tracking ability of the controller.
- A relatively small weight is assigned to the quadrotor velocity state to allow higher quadrotor velocities.
- A relatively large weight is assigned to the payload angular rates to increase damping

by penalising large angular rates.

- A very small weight is assigned to the payload swing angle to allow bigger swing angles. Bigger swing angles allow for higher bandwidth velocity control.
- The control input weight is chosen to ensure efficient actuator usage.

With this in mind, an iterative approach was taken to determine the values of the state weighting matrix and input weight. The final weights were chosen to be $\mathbf{Q} = \text{diag}\left([0.8,\ 0.11,\ 0.2,\ 0.05]\right)$ and $R = 0.06$.

Another reason for the small weight assigned to the payload swing angle state, $\theta_p^{\mathcal{I}}$, is to reduce the effect of the payload swing angle feedback control when flying at high speeds. Due to the aerodynamic drag on the payload at high speeds, the swing angle settles at a new non-zero trim state, $\theta_{p_0}^{\mathcal{I}}$. However, the controller is designed to damp the payload oscillations and not necessarily to maintain a zero degree swing angle.

The non-linear longitudinal velocity step response of the nominal LQR controller, designed for the nominal payload which has a 2 kg mass and a 1 m cable length is shown in Figure 7.11. The controller is able to damp the oscillations of the swinging payload and shows an improvement in performance compared to the standard longitudinal velocity PID controller. The system response shows an overshoot of 4.5%, a 2% settling time of 7.2 s and a bandwidth of 0.72 rad/s.

Figure 7.11 also shows the step responses of the nominal LQR controller for uncertain suspended payload systems. It is clear that the performance of the nominal LQR controller deteriorates for uncertain suspended payload systems and in some cases produce unstable responses.

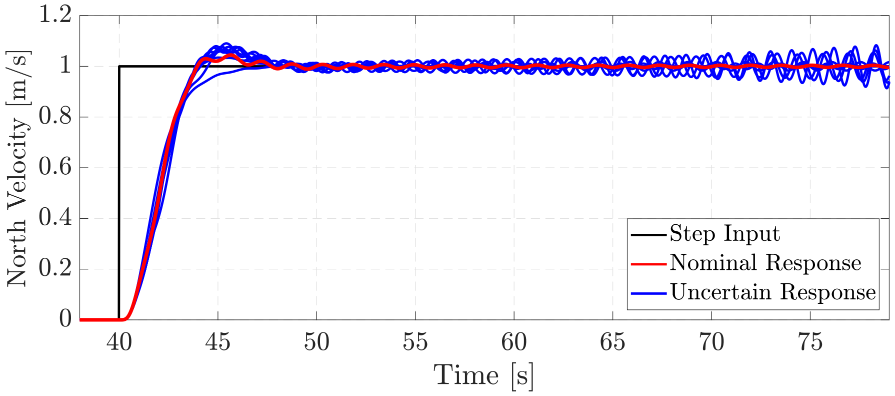


Figure 7.11: The quadrotor longitudinal velocity step responses with the nominal LQR controller for the nominal and uncertain suspended payload systems.

The controller attempts to reduce the effect of the lightly damped poles of the plant. However, as the suspended payload system changes, the lightly damped poles shift and the pole-zero cancellation effect of the LQR controller reduces. This confirms that the nominal LQR controller is not robust against parameter changes.

Therefore, an adaptive LQR controller is implemented. As the unknown payload parameters are estimated as discussed in Chapter 6, the LQR controller can be redesigned online, using the payload parameter estimates, by following the same process as described by

Equations (7.17) to (7.20). The state and input weights chosen for $\mathbf{Q}$ and R remains constant as these weights are chosen to obtain the desired controller performance as discussed.

Figure 7.12 shows the nominal LQR controller response for the nominal suspended payload system and the adaptive LQR controller response for different suspended payload systems. The adaptive LQR controller is redesigned online for each suspended payload system using the payload parameter estimates. It is clear that the responses of the adaptive LQR controller are similar to the response of the nominal LQR controller with the nominal suspended payload system. The adaptive LQR controller is able to damp the oscillations caused by the swinging payloads and eliminate unstable responses. Similar results are obtained in the lateral direction.

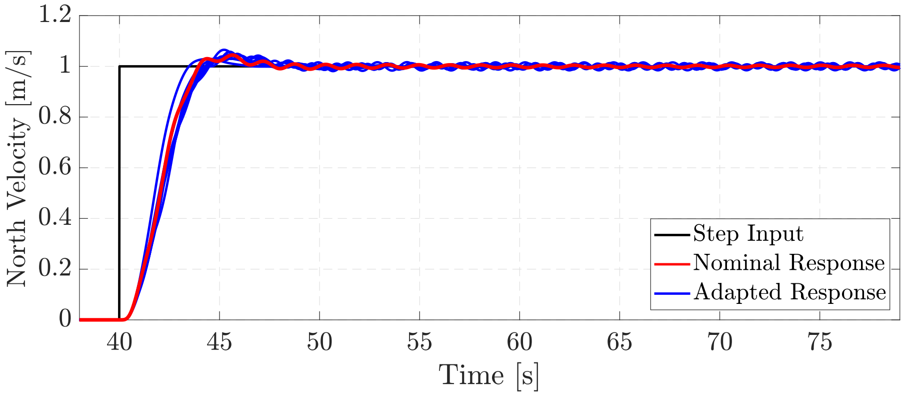


Figure 7.12: The quadrotor longitudinal velocity step response with adaptive LQR control for different suspended payload systems.

Robustness

The performance of the adaptive LQR controller when introduced to disturbances as well as the performance under the influence of sensor noise are investigated.

The steady-state tracking performance and disturbance rejection of the LQR controller for the ideal non-linear model is shown in Figure 7.13. A constant disturbance is introduced at time $t = 48\ s$ for the nominal quadrotor and suspended payload system. It is clear that the LQR controller is able to reject the disturbance while damping the payload oscillations.

To replicate real-world scenarios, high-frequency sensor noise, low-frequency sensor drift and sensor biases were added to the quadrotor model. The response of the nominal system is shown in Figure 7.14. The low-frequency sensor drift causes the quadrotor to drift which in turn induces small payload oscillations. However, the LQR controller still ensures a stable response by damping these oscillations and preventing them from increasing.

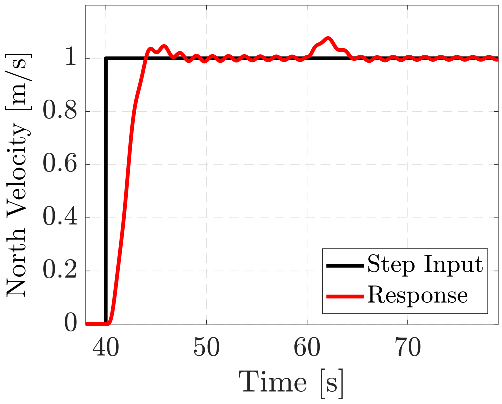


Figure 7.13: The longitudinal velocity disturbance rejection of the adaptive LQR controller.

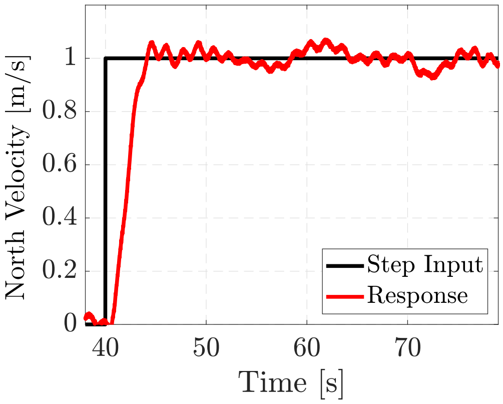


Figure 7.14: The quadrotor longitudinal velocity step response with sensor noise.

7.4. Simulation Results

The performance of the proposed swing damping velocity control system, which consists of a notch filter and an LQR controller is presented. The proposed velocity controller is compared to the standard PID velocity controllers designed in Chapter 5. The longitudinal velocity and payload swing angle responses of the nominal non-linear quadrotor and suspended payload model are shown in Figure 7.15 and 7.16, respectively.

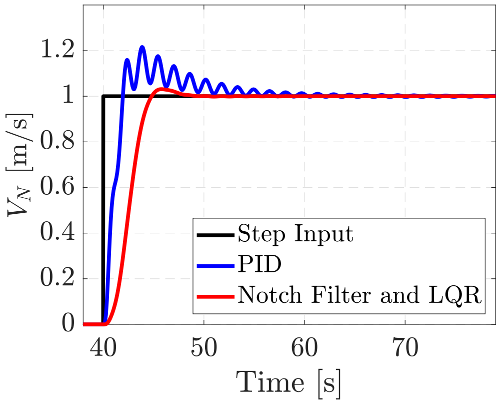


Figure 7.15: The quadrotor longitudinal velocity response with PID control and adaptive LQR and notch filter control.

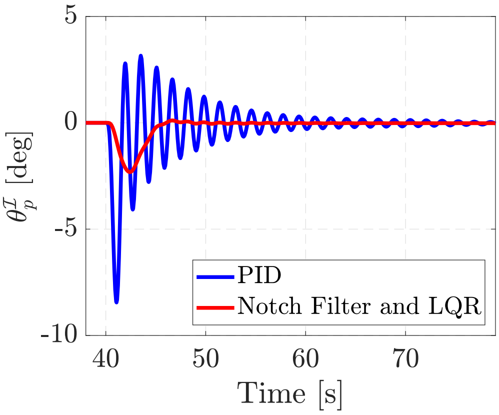


Figure 7.16: The $\theta_p^{\mathcal{I}}$ swing angle response with PID control and adaptive LQR and notch filter control.

The combined notch filter and LQR controller show an improvement in performance when compared to the velocity responses of the individual notch filter and LQR controllers in Figure 7.6 and 7.12. The proposed control strategy shows a significant improvement compared to the standard PID velocity controllers. The notch filter is able to prevent the initial excitation of the payload oscillations while the LQR controller is able to effectively damp the residual oscillations. An increase in the rise time of about 3.6 s is visible in Figure 7.15. However, the slower system response is acceptable as the controller is designed with an emphasis on its ability to prevent and damp payload oscillations. The same results are obtained in the lateral direction and not presented here.

Figure 7.17 and 7.18 show the system response under the influence of sensor noise for a velocity reference in both the longitudinal and lateral direction. It is clear that the performance in the longitudinal and lateral directions are very similar. Similar to the response shown in Figure 7.14, the sensor noise induces small payload oscillations. However, the controller is able to prevent the excitation of these oscillations.

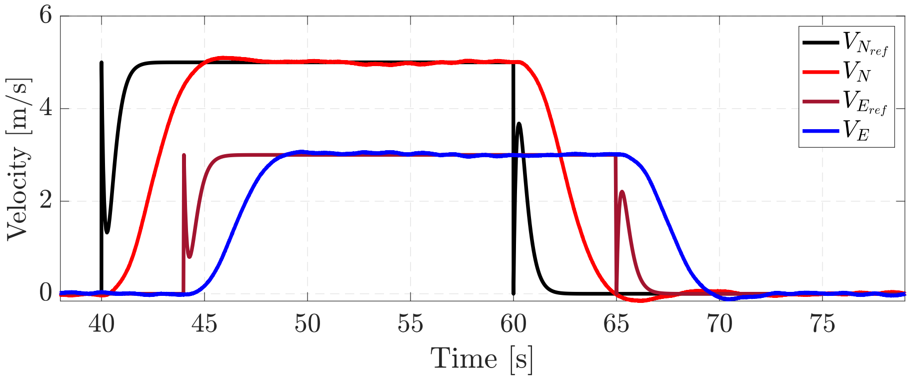


Figure 7.17: The quadrotor velocity response with notch filter and adaptive LQR control.

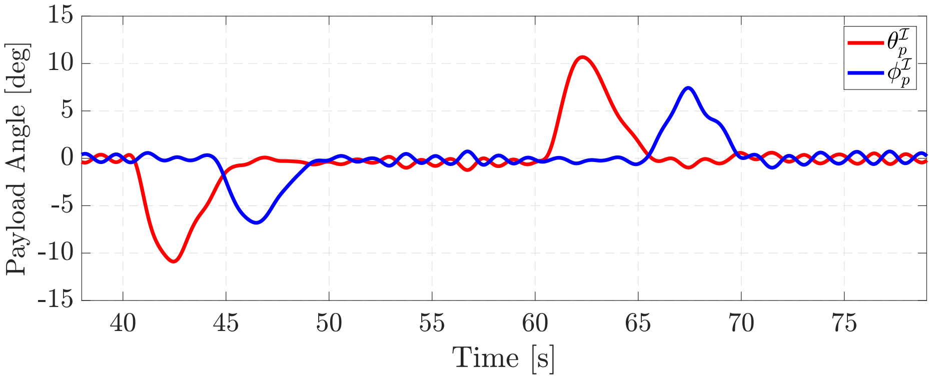


Figure 7.18: The payload swing angle response with notch filter and adaptive LQR control.

7.5. Summary

In this chapter, the architecture and design of a swing damping control system for a quadrotor with a suspended payload were presented. The quadrotor and suspended payload dynamics, derived in Chapter 3, were linearized and served as the basis for the controller design. The proposed controller consist of a feedforward and full-state feedback controller designed to make use of the unknown suspended payload parameter estimates and vision-based state estimator discussed in Chapter 6.

The feedforward controller is based on a technique called input shaping. A notch filter is designed and implemented to shape the longitudinal and lateral quadrotor velocity references to avoid the excitation of payload oscillations. Using the notch filter with the standard velocity PID controllers designed in Chapter 5 showed an improvement in performance. However, as this is a feedforward controller, no active damping is provided to damp residual payload oscillations.

A LQG full-state feedback controller is designed and implemented to actively damp the suspended payload oscillations by simultaeneously controlling the quadrotor's longitudinal and lateral velocity and the payload swing angles and swing rates. Integral action is added to ensure zero steady-state tracking errors. The LQG controller was initially designed for the nominal quadrotor and suspended payload system which has a mass of 2 kg and a cable length of 1 m. This controller was able to damp the oscillations caused by the nominal suspended payload system. However, for different suspended payload systems the controller performance deteriorated and showed underdamped behaviour and instability. This motivated the need for an adaptive LQG controller that can be redesigned online for the different suspended payload systems to ensure stability. The adaptive LQG controller proved to be robust and was able to damp payload oscillations.

The combined notch filter and LQG control architecture was implemented and exhibited nearly swing free motion. While the notch filter prevents the excitation of payload oscillations, the LQG controller is able to damp the residual payload oscillations as well as any externally imposed oscillations. The proposed swing damping control system proves to be a viable solution for the robust control of a quadrotor with an unknown suspended payload.

Chapter 8

Implementation and Results

This chapter focusses on the implementation and evaluation of the proposed control strategy in the PX4 flight stack as well as the evaluation of the custom ROS nodes on the Nvidia Jetson Nano onboard computer. HITL simulations are performed to ensure that the Pixhawk flight controller perform as expected with the modifications made to the PX4 flight stack and that the custom ROS nodes execute as intended on the Nvidia Jetson Nano onboard computer.

A quick overview of the control strategy implementation is given, followed by the evaluation of the control strategy in the PX4 and Gazebo simulation environment. The HITL simulation results are presented and discussed.

8.1. PX4 Implementation

The PX4 longitudinal and lateral velocity controller architecture is modified to implement the proposed control strategy consisting of the notch filter and LQG controller. Two parameters are created to activate the notch filter and switch between the PID and LQG controllers, respectively. A direct form 2 implementation is used for the discrete implementation of the notch filter. Custom PX4 modules are created for the RLS algorithm for the payload mass estimation and the EKF for the payload state estimation discussed in Chapter 6.

8.2. PX4 and Gazebo HITL Simulation Results

HITL simulations of the quadrotor with three different suspended payloads, as given in Table 8.1, are performed to evaluate the proposed control and estimation strategies. The hardware performance is also evaluated in the HITL simulations. The HITL configuration as illustrated in Figure 4.8 is used.

8.2.1. Hardware Performance Analysis

The performance of the Pixhawk 4 flight controller and Nvidia Jetson Nano onboard computer are evaluated in a HITL simulation and the results are presented.

System	Payload Mass [kg]	Cable Length [m]
1	1	0.6
2	2	1
3	3	2

Table 8.1: The quadrotor and payload systems considered for the HITL simulations.

Pixhawk 4 Flight Controller

The Pixhawk 4 has a 32-bit STM32F765 Cortex M7 processor with a clock speed of 216 MHz and 512 kB RAM [43]. Figure 8.1 shows the CPU load and RAM usage during a HITL simulation. At time $t = 90\ s$, the notch filter and LQR controllers are activated and the EKF is started. The CPU load remains relatively constant at 72% while a slight increase in the RAM usage can be seen at this point. The average RAM usage increases to 56%. Therefore, the modifications and additions to the PX4 flight stack has a negligible effect on the CPU load and RAM usage of the Pixhawk 4 flight controller.

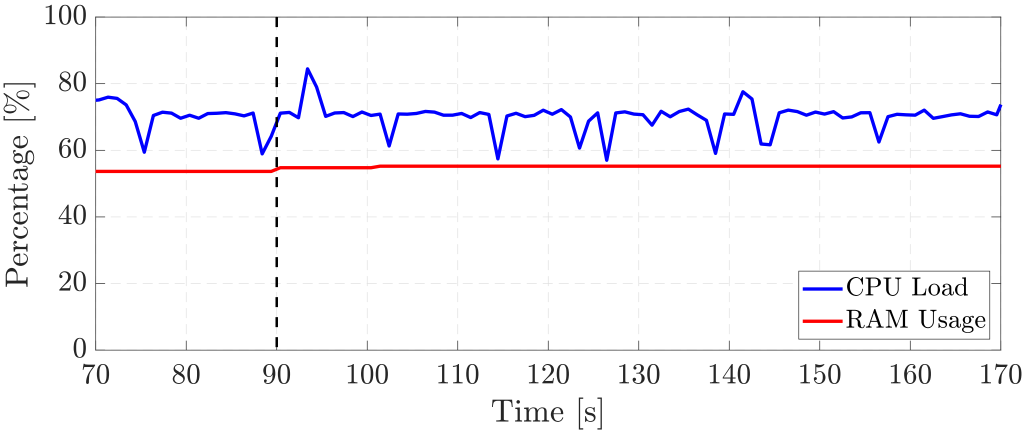


Figure 8.1: The Pixhawk CPU load and RAM usage during a HITL simulation.

Nvidia Jetson Nano Onboard Computer

The NVidia Jetson onboard computer is responsible for the execution of the ROS nodes which includes the image processing node, the payload parameter estimation node, LQG controller adjustment node and the MAVROS communication node. The image processing node is time critical as it is required to provide payload swing angle measurements in real time for full-state feedback control. The execution times of the detection and tracking algorithms of the image processing node are summarized in Table 8.2.

The Nvidia Jetson Nano is able to detect and track the suspended payload at a rate of 10 frames per second (fps) which is more than adequate for the range of payload parameters considered. A measurement frequency of 10 Hz is considerably more than the natural frequencies of the considered quadrotor and suspended payload systems.

However, the current image processing algorithms are implemented in the Python pro-

Vision System Component	**Execution Time [ms]**	**Percentage of Total Time [%]**
Background Motion Estimation and Compensation	92.7	94.6
Frame Differencing	0.03	< 0.01
Noise Filtering	0.35	3.5
ROI Identification	0.14	1.4
Total Time Detection	**98.0**	**100**
Tracking	94.8	100
Total Time Tracking	**94.8**	**100**

Table 8.2: The execution time of the payload detection and tracking algorithms.

gramming language and not fully optimized. It is recommended to implemented these algorithms in C++ to increase the measurement rate or so that it can be used with more resource constraint hardware. It is recommended that optimization starts with the first step of the detection algorithm as it is responsible for about 95 % of the total execution time. This should be followed by a C++ implementation of the tracking algorithm to ensure a higher payload tracking rate.

8.2.2. Payload Parameter Estimation

The full system identification phase as discussed in Chapter 6 is verified in the HITL simulations with the payload mass estimation and cable length estimation results shown in Figures 8.2 and 8.3, respectively.

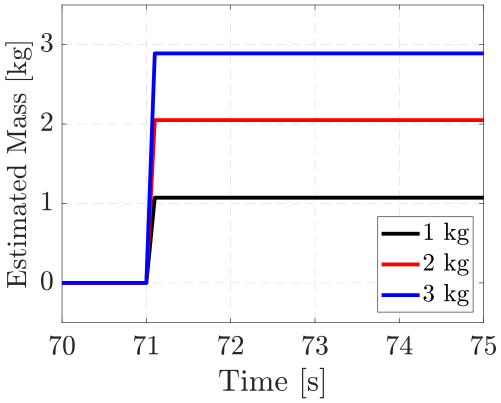


Figure 8.2: Payload mass estimation of the HITL simulations.

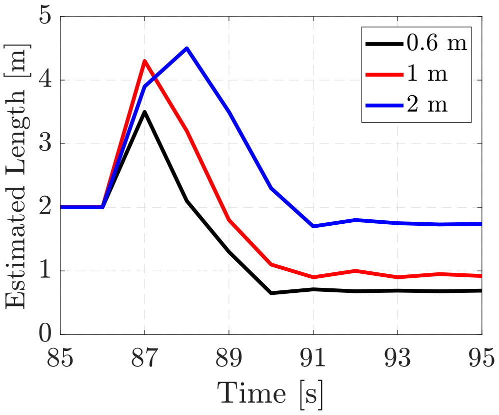


Figure 8.3: Payload length estimation of the HITL simulations.

The RLS algorithm quickly converges to payload mass estimates of 1.07 kg, 2.05 kg and 2.89 kg for the 1 kg, 2 kg and 3 kg payload masses, respectively. The RLS algorithm is able to accurately estimate the payload masses with negligibly small errors.

Greater estimate errors are observed for the cable length estimates. Cable length estimates of 0.69 m , 0.92 m and 1.74 m were obtained for the 0.6 m, 1 m and 2 m cables, respectively, with a maximum error of 15% for the 0.6 m cable. These estimate errors may be the result of a change in the oscillation frequency due to the quadrotor controllers.

To improve the accuracy of the cable length estimator, the cable length estimation step can be executed multiple times to obtain an average estimate which may be more accurate. However, the robustness of the proposed payload state estimation and control strategies against paremeter estimate errors does not justify the extra time required for multiple steps as is evident in Sections 8.2.3 and 8.2.4.

8.2.3. Payload State Estimation

Only the payload state estimation of the nominal quadrotor and suspended payload system, which has a payload mass of 2 kg and a cable length of 1 m, are presented. Similar results for the estimator performance were obtained for the other systems and are not presented. The EKF is able to accurately estimate the suspended payload states as shown in Figures 8.4 to 8.7. A slight delay is visible between the estimated and true states which is due to the delay of the swing angle measurements received from the vision system. However, the delay is small and not of concern. It is also clear that the EKF is robust against the parameter estimate errors.

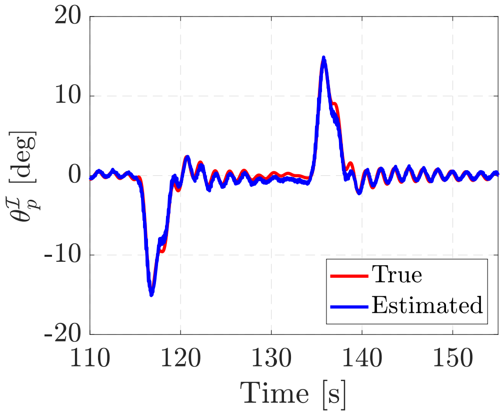


Figure 8.4: The $\theta_p^{\mathcal{I}}$ swing angle estimate of a HITL simulation.

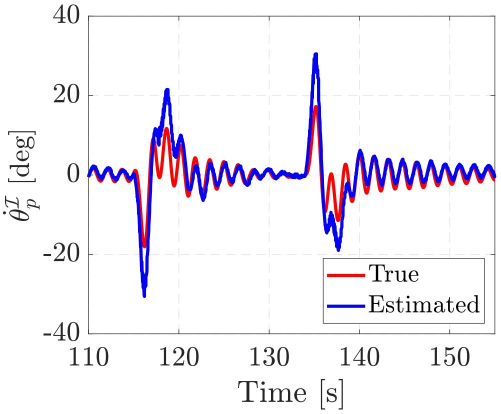


Figure 8.5: The $\dot{\theta}_p^{\mathcal{I}}$ swing rate estimate of a HITL simulation.

In Figures 8.5 and 8.7, it can be seen that the EKF overestimates the swing angle rates when a velocity reference is commanded. However, this does not pose any problems as the larger swing angle rate will result in a more aggressive controller.

8.2.4. Quadrotor and Payload with Swing Damping Control

The performance of the swing damping control strategy is evaluated in the HITL simulations where longitudinal and lateral velocity references are commanded. Figures 8.8 and 8.9 show the longitudinal and lateral velocity responses of the quadrotor with the suspended payloads as given in Table 8.1.

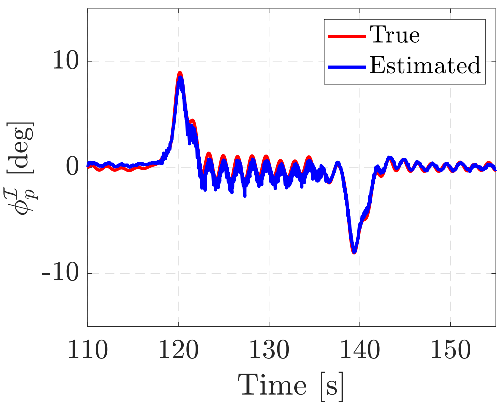


Figure 8.6: The $\phi_p^{\mathcal{I}}$ swing angle estimate of a HITL simulation.

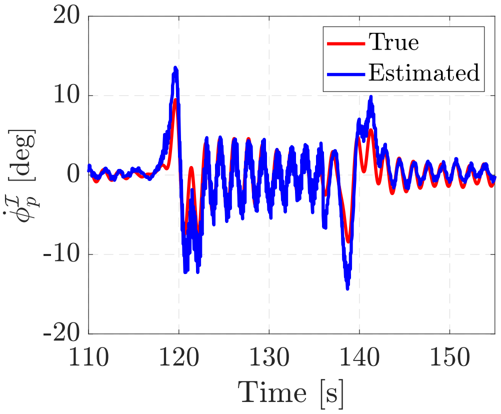


Figure 8.7: The $\dot{\phi}_p^{\mathcal{I}}$ swing rate estimate of a HITL simulation.

The same input references are commanded for the three quadrotor and suspended payload systems. The adaptive notch filter shapes the input references for the specific system as can be seen in Figure 7.17 in Chapter 7. However, only the unshaped input references are illustrated in Figures 8.8 and 8.9 to maintain clarity.

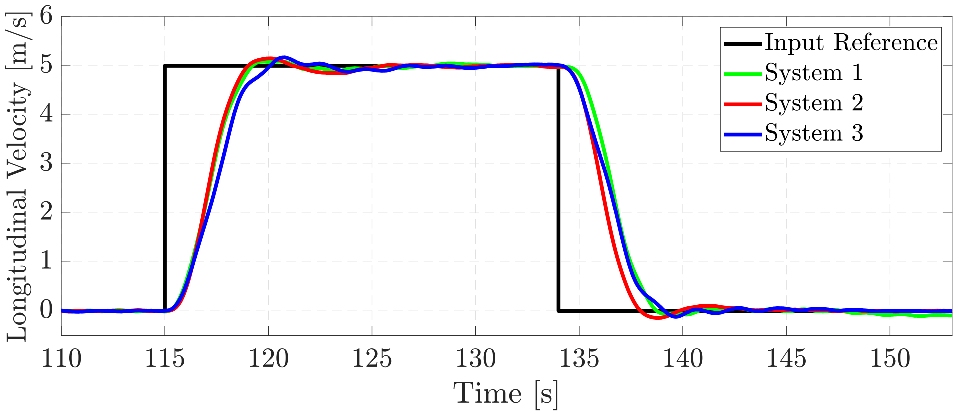


Figure 8.8: The quadrotor longitudinal velocity responses of the HITL simulations.

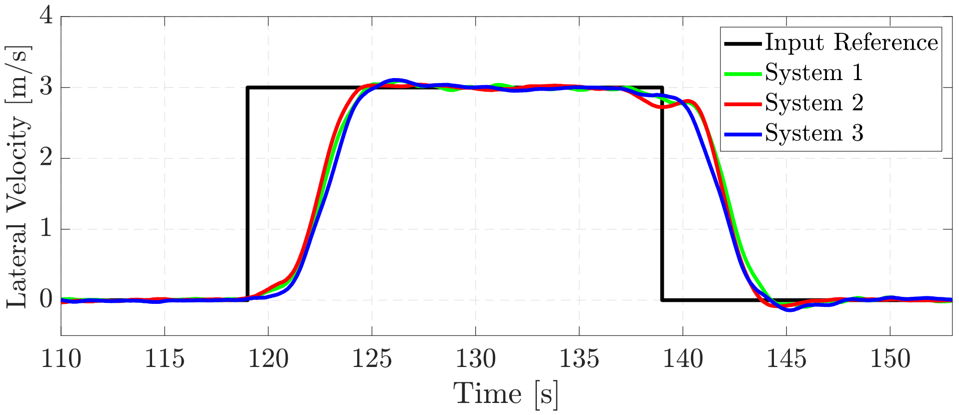


Figure 8.9: The quadrotor lateral velocity responses of the HITL simulations.

The simulation results show that the proposed controller is robust against sensor noise and parameter estimate errors and is able to damp the oscillations caused by the suspended payloads. This is also evident in the payload swing angle responses shown in Figures 8.10 and 8.11. Similar responses are obtained for the different systems which indicate that the controller is able to adapt to the unknown suspended payload. These results confirm that the controller objectives are reached.

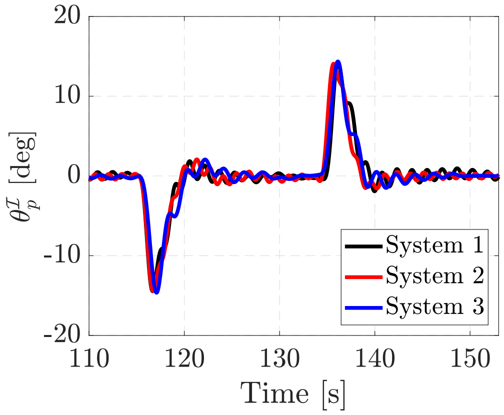


Figure 8.10: The $\theta_p^{\mathcal{I}}$ swing angle responses of the HITL simulations.

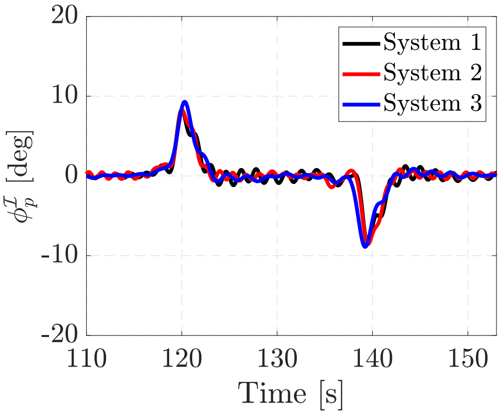


Figure 8.11: The $\phi_p^{\mathcal{I}}$ swing angle responses of the HITL simulations.

8.3. Summary

This chapter focussed on the implementation and evaluation of the proposed swing damping control strategy as well as the proposed payload parameter estimation and state estimation strategies. The control and estimation strategies were implemented in PX4 and ROS and evaluated in the PX4 and Gazebo simulation environment.

The parameter estimation strategies were able to estimate the unknown suspended payload parameters. The vision-based state estimator was able to provide accurate state estimates using payload swing angle measurements provided by the vision system. The proposed control strategy, consisting of a notch filter and LQG controller, was able to significantly damp the suspended payload oscillations.

The HITL simulation results prove that the proposed control strategy is practically feasible and can be implemented and verified in practical flight tests.

Chapter 9

Conclusion and Recommendations

9.1. Summary

In this thesis, a flight control system for a quadrotor with an unknown suspended payload is proposed. The suspended payload is unknown as its parameters are unknown and direct measurements of its states are not available. The suspended payload induces oscillations which significantly alters the flight dynamics of the quadrotor. A solution is proposed to damp these oscillations and ensure stable flight.

A literature study was conducted to identify proposed solutions for this problem. It was found that most solutions considered the problem not for completely unknown suspended payloads, but rather partially unknown suspended payloads. Adaptive and robust control techniques were proposed where adaptive control techniques estimate the unknown payload parameters and adapt accordingly and robust control techniques ensure stability for a range of parameter uncertainties. It was also observed that most of the control strategies assumed that the suspended payload states are known or can be measured using sophisticated external motion capture systems for feedback control. However, this is not practically feasible for real-world scenarios. A solution with practical feasibility in mind is therefore proposed.

A mathematical model of the quadrotor named Griffin was derived and implemented in the MATLAB/Simulink simulation environment for controller design and verification. The control system architecture of the PX4 flight control stack was adopted and a flight control system for the quadrotor without a suspended payload was designed. The designed control system was verified in the MATLAB/Simulink as well as the PX4 and Gazebo simulation environments. The mathematical model of the quadrotor with a suspended payload was then derived and implemented in the MATLAB/Simulink simulation environment. It was concluded that the standard quadrotor control system produces a lightly damped response due to the suspended payload oscillations and is not adequate for aerial payload transportation applications.

A control strategy was presented to damp the payload oscillations and ensure stable flight. The solution is a vision-based adaptive feedforward-feedback control strategy with an emphasis on payload swing damping. The assumption was made that the suspended payload is attached to the CoM of the quadrotor and only affects the translational dynamics of the quadrotor. Therefore, the proposed control strategy replaces the standard horizontal velocity controllers of the quadrotor.

The proposed control strategy requires estimates of the unknown payload parameters and states. Therefore, a system identification phase is implemented to estimate the unknown payload parameters and a vision-based state estimator is implemented to provide state estimates of the suspended payload. A RLS algorithm is applied to the vertical dynamics of the system to estimate the unknown payload mass. A dedicated sine wave estimator is used to estimate the oscillation frequency of the suspended payload which is proportional to the payload cable length. The estimation algorithms were able to provide parameter estimates with good accuracy in the presence of sensor noise.

A vision-based EKF state estimator is proposed to provide payload state estimates. The vision system consists of a downward facing camera attached to the bottom of the quadrotor that measures the suspended payload swing angles. The vision system is designed to be independent of visual markers and assumes no visual information about the suspended payload. A motion-based object detection technique is implemented which is able to detect any arbitrary payload. The vision system can therefore be used to provide swing angle measurements for any arbitrary multirotor and suspended payload system.

The control strategy has a feedforward controller consisting of a notch filter that functions as an input shaper to generate swing free trajectories by suppressing the unwanted payload oscillations. The estimated payload parameters are used to redesign the notch filter online for the specific suspended payload. The notch filter proved to be robust against parameter estimate errors of up to 20%. It can be concluded that the notch filter is a simple and effective alternative to more common input shaping techniques such as the ZVD input shaper.

In combination with the notch filter, a full-state feedback adaptive LQG controller with integral action was implemented to simultaneously control the quadrotor longitudinal and lateral velocities and actively damp the suspended payload oscillations. As with the notch filter, the LQG controller is redesigned online for the specific suspended payload.

The control strategy was implemented and verified in SITL and HITL simulations for different suspended payloads. The control strategy proved to be robust against sensor noise and external disturbances and is able to damp payload oscillations resulting in near swing free motion. The control strategy is therefore proved to be practically feasible.

9.2. Future Work

Future work includes the practical demonstration of the proposed vision-based control strategy. With the HITL simulations confirming the practical feasibility of the proposed solution, practical flight tests are the logical next step.

The proposed control strategy is composed of several components, including the payload parameter estimation algorithms, the vision-based state estimator, a notch filter and a LQG controller. These individual components can be used in combination with other control strategies for quadrotors with suspended payloads. Further improvements can also be made to the current components of the control strategy.

The recommended future work includes:

- The optimization of the payload detection algorithm for higher detection rates or to be used with resource constraint hardware.
- Explore alternative techniques such as convolutional neural networks (CNN) for the payload detection.
- The redesign of the EKF to compensate for the delayed swing angle measurements due to the image processing latency.
- The use of the payload parameter estimation algorithms with other adaptive control techniques for active swing damping.
- The use of the notch filter in combination with other adaptive and robust control techniques.
- The use of the vision-based estimator and payload parameter estimation algorithms in trajectory generation algorithms to obtain swing free motion.
- The use of the vision-based estimator and payload parameter estimation algorithms with trajectory generation algorithms for aggressive maneuvering through cluttered environments.

It is clear that the components of the proposed control strategy can be used individually to aid in the development of alternative solutions for the aerial payload transportation problem.

Bibliography

[1] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, “Design, modeling, estimation and control for aerial grasping and manipulation,” pp. 2668–2673, 2011.

[2] I. Palunko, P. Cruz, and R. Fierro, “Agile load transportation : Safe and efficient load manipulation with aerial robots,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[3] “Trendconfirmation - food drone delivery,” May 2019. [Online]. Available: https://www.fluxtrends.com/trendconfirmation-food-drone-delivery/

[4] M. Vahdanipour and M. Khodabandeh, “Adaptive fractional order sliding mode control for a quadrotor with a varying load,” *Aerospace Science and Technology*, vol. 86, 03 2019.

[5] B. J. Emran, J. Dias, L. Seneviratne, and G. Cai, “Robust adaptive control design for quadcopter payload add and drop applications,” pp. 3252–3257, 2015.

[6] C. Commander, “Drones for search and rescue missions,” Aug 2020. [Online]. Available: https://comparecommander.com/drones-for-search-rescue-missions/

[7] N. Singer, W. Singhose, and E. Kriikku, “An input shaping controller enabling cranes to move without sway.”

[8] S. Ichikawa, A. Castro, N. Johnson, H. Kojima, and W. Singhose, “Dynamics and command shaping control of quadcopters carrying suspended loads,” *IFAC-PapersOnLine*, vol. 51, no. 14, pp. 84 – 88, 2018, 14th IFAC Workshop on Time Delay Systems TDS 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318309492

[9] J. Potter, W. Singhose, and M. Costelloy, “Reducing swing of model helicopter sling load using input shaping,” in *2011 9th IEEE International Conference on Control and Automation (ICCA)*, 2011, pp. 348–353.

[10] M. Bisgaard, A. la Cour-Harbo, and J. Bendtsen, “Input shaping for helicopter slung load swing reduction,” 08 2008.

[11] K. Klausen, T. Fossen, and T. Johansen, “Nonlinear control of a multirotor uav with suspended load,” *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 176–184, 07 2015.

[12] S. Sadr, S. Ali, S. Moosavian, and P. Zarafshan, “Dynamics modeling and control of a quadrotor with swing load,” *Journal of Robotics*, vol. 2014, 12 2014.

[13] I. Palunko, R. Fierro, and P. Cruz, “Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach,” pp. 2691–2697, 2012.

[14] B. Xian, S. Wang, and S. Yang, "An online trajectory planning approach for a quadrotor uav with a slung payload," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6669–6678, 2020.

[15] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," pp. 4902–4909, 2013.

[16] T. Kusznir and J. Smoczek, "Sliding mode-based control of a uav quadrotor for suppressing the cable-suspended payload vibration," *Journal of Control Science and Engineering*, vol. 2020, pp. 1–12, 01 2020.

[17] C. C. Taylor and J. A. A. Engelbrecht, "Acceleration-based control of a quadrotor with a swinging payload," pp. 1–8, 2016.

[18] M. M. Nicotra, E. Garone, R. Naldi, and L. Marconi, "Nested saturation control of an uav carrying a suspended load," pp. 3585–3590, June 2014.

[19] M. M. de Almeida and G. V. Raffo, "Nonlinear control of a tiltrotor uav for load transportation," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 232 – 237, 2015, 11th IFAC Symposium on Robot Control SYROCO 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896315026634

[20] G. Raffo and M. Almeida, "A load transportation nonlinear control strategy using a tilt-rotor uav," *Journal of Advanced Transportation*, vol. 2018, pp. 1–20, 06 2018.

[21] H. M. Omar, "Designing anti-swing fuzzy controller for helicopter slung-load system near hover by particle swarms," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 223 – 234, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963813000667

[22] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. García, "Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6718–6723.

[23] S. Dai, T. Lee, and D. S. Bernstein, "Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass," pp. 6149–6154, 2014.

[24] S. Yang and B. Xian, "Energy-based nonlinear adaptive control design for the quadrotor uav system with a suspended payload," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2054–2064, 2020.

[25] A. l. M. Bisgaard and J. Dimon Bendtson, "Adaptive control system for autonomous helicopter slung load operations," *Control Engineering Practice*, vol. vol. 18.

[26] Y. Feng, C. A. Rabbath, S. Rakheja, and C. Su, "Adaptive controller design for generic quadrotor aircraft platform subject to slung load," pp. 1135–1139, 2015.

[27] X. Zhou, X. Zhang, J. Zhang, and R. Liu, "Stabilization and trajectory control of a quadrotor with uncertain suspended load," 2016.

[28] H. I. Lee, B.-Y. Lee, D. W. Yoo, G.-H. Moon, and M. J. Tahk, "Dynamics modeling and robust controller design of the multi-uav transportation system," *29th Congress of the International Council of the Aeronautical Sciences, ICAS 2014*, 01 2014.

[29] C. Taylor, “Robust control of a quadrotor with a suspended payload.” [Online]. Available: http://hdl.handle.net/10019.1/107858

[30] B. Xian, S. Wang, and S. Yang, “An online trajectory planning approach for a quadrotor uav with a slung payload,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6669–6678, 2020.

[31] S. Lee and H. Kim, “Autonomous swing-angle estimation for stable slung-load flight of multi-rotor uavs,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4576–4581, 2017.

[32] M. Zürn, K. Morton, A. Heckmann, A. McFadyen, S. Notter, and F. Gonzalez, “Mpc controlled multirotor with suspended slung load: System architecture and visual load detection,” pp. 1–11, 2016.

[33] A. Vargas and D. Anderson, “Computer vision technique to estimate the slung load dynamics when coupled to a multirotor unmanned aerial vehicle,” *RIIIT*, vol. ISSN 2007-9753, 05 2017.

[34] S. Tang, V. Wüest, and V. Kumar, “Aggressive flight with suspended payloads using vision-based control,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1152–1159, 2018.

[35] E. L. de Angelis, “Swing angle estimation for multicopter slung load applications,” *Aerospace Science and Technology*, vol. 89, pp. 264 – 274, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1270963818326282

[36] R. Jain and H. H. Nagel, “On the analysis of accumulative difference pictures from image sequences of real world scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 206–214, 1979.

[37] N. Paragios and G. Tziritas, “Adaptive detection and localization of moving objects in image sequences,” *Signal Processing: Image Communication*, vol. 14, pp. 277–296, 1999.

[38] A. S. A. Behrad and S. A. Motamedi, “A robust vision-based moving target detection and tracking system,” 2001.

[39] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1, pp. 185 – 203, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370281900242

[40] J. H. Blakelock, *Automatic control of aircraft and missiles*. Wiley, 1991. [Online]. Available: https://books.google.co.in/books?id=ubcczZUDCsMC

[41] A. Erasmus, “Stabilization of a rotary wing unmanned aerial vehicle with an unknown suspended payload.” [Online]. Available: http://hdl.handle.net/10019.1/107832

[42] “Camera module.” [Online]. Available: https://www.raspberrypi.org/documentation/hardware/camera/

[43] “Holybro pixhawk 4 (fmuv5),” (Accessed: 30/09/20). [Online]. Available: https://docs.px4.io/master/en/flight_controller/pixhawk4.html

[44] "Jetson nano developer kit," Feb 2020. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[45] "Raspberry pi camera v2 camera module, csi-2, 3280 x 2464 resolution." [Online]. Available: https://za.rs-online.com/web/p/raspberry-pi-cameras/9132664/

[46] "Px4 architectural overview." [Online]. Available: https://dev.px4.io/master/en/concept/architecture.html

[47] M. H. Dario Brescianini and R. D'Andrea, "Nonlinear quadrocopter attitude control," 2013.

[48] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[49] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," pp. 2564–2571, 2011.

[50] D. G. Lowe, "Object recognition from local scale-invariant features," vol. 2, pp. 1150–1157 vol.2, 1999.

[51] T. T. H. Bay, A. Ess and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 06 2008.

[52] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," vol. 24, no. 6, June 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[53] A. Lukežic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," pp. 4847–4856, 2017.

[54] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," pp. 2544–2550, 2010.

[55] M. S. D. Mellinger, Q. Lindsey and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," *2011 IEEE International Conference on Intelligent Robots and Systems*, pp. 2668–2673, 2011.

[56] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 76–82, 1990.

[57] A. Component, W. Singhose, N. Singer, and W. Seering, "Comparison of command shaping methods for reducing residual vibration," vol. 2, 03 1996.

Appendices

Appendix A

Quadrotor Control System Gains

This Appendix contains the quadrotor control system gains as designed in Chapter 5.

The angular rate controller gains are given in Table A.1.

Controller	**P Gain**	**I Gain**	**D Gain**
Roll Rate	0.086	0.020	0.003
Pitch Rate	0.0842	0.0196	0.003
Yaw Rate	0.035	0.03	0.0

Table A.1: The angular rate controller gains.

The angular controller gains are given in Table A.2.

Controller	**P Gain**
Roll Angle	3
Pitch Angle	3
Yaw Angle	1.2

Table A.2: The angular controller gains.

The velocity controller gains are given in Table A.3.

Controller	**P Gain**	**I Gain**	**D Gain**
Longitudinal Velocity	0.048	0.008	0.002
Lateral Velocity	0.048	0.008	0.002
Heave Velocity	0.1	0.01	0.0

Table A.3: The velocity controller gains.

The position controller gains are given in Table A.4.

Controller	**P Gain**
North Position	0.35
East Position	0.35
Down Position	0.9

Table A.4: The position controller gains.