

The Pennsylvania State University  
The Graduate School  
College of Engineering

**VISION-BASED RELATIVE DECK STATE ESTIMATION USED WITH TAU  
BASED LANDINGS**

A Thesis in  
Aerospace Engineering  
by  
William Holmes

© 2017 William Holmes

Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Master of Science

August 2017

The thesis of William Holmes was reviewed and approved\* by the following:

Jack W. Langelaan  
Associate Professor of Aerospace Engineering  
Thesis Advisor

Joseph F. Horn  
Professor of Aerospace Engineering

George A. Lesieutre  
Associate Dean for Research, College of Engineering

\*Signatures are on file in the Graduate School.

# Abstract

Landing a helicopter on a moving deck is a daunting task for even the most skilled pilot. The task involves controlling the vehicle, tracking the movements of the ship, and estimating the future positions of the deck. The difficulty of this task is the motivation for this thesis. In this thesis, a method of estimating the relative deck states necessary for autonomous landing as well as a method of generating valid trajectories to the deck surface are presented.

An unscented Kalman filter is presented which fuses measurements from a monocular vision system with an inertial measurement unit, both of which are carried on board the vehicle. Using the Kalman filter generated estimates, biologically inspired trajectories are generated to the deck which satisfy the vehicle dynamical constraints and several terminal landing constraints.

The Kalman filter and biological trajectories were used to land a mid-size quadrotor on a static and moving platform. Single run and aggregate performance of the Kalman filter are presented.

# Table of Contents

List of Figures	vii
List of Symbols	ix
Acknowledgments	xi
<b>Chapter 1</b>	
<b>Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 An Overview of a Self-Contained Landing System . . . . .	3
1.3 Problem Description . . . . .	4
1.4 Related Work . . . . .	4
1.4.1 Ship Landing . . . . .	4
1.4.2 Monocular Vision and Perception . . . . .	5
1.5 Contributions . . . . .	6
1.6 Readers Guide . . . . .	6
<b>Chapter 2</b>	
<b>The Ship Landing Problem</b>	<b>7</b>
2.1 Problem Statement . . . . .	7
2.2 System Models . . . . .	8
2.2.1 Reference Frames and Kinematic Relationships . . . . .	8
2.2.2 Vehicle Dynamics . . . . .	10
2.2.3 Landing Coordinate Systems . . . . .	11
2.3 Landing Constraints . . . . .	12
2.4 Sensor Models . . . . .	12
2.4.1 Vision Model . . . . .	12
2.4.2 Inertial Measurement Model . . . . .	14
2.5 Tau Theory . . . . .	14
2.5.1 Tau-Dot . . . . .	15
2.5.2 Tau-Guidance . . . . .	16
2.5.3 High Order Tau-Guidance . . . . .	18
2.6 Unscented Kalman Filter . . . . .	19
2.7 Summary: The Ship Landing Problem . . . . .	19

<b>Chapter 3</b>	
<b>Building a Unscented Kalman Filter</b>	<b>21</b>
3.1 Sigma Point Filter Steps . . . . .	21
3.1.1 Prediction Step . . . . .	22
3.1.2 Correction Step . . . . .	23
3.2 Choice of States . . . . .	24
3.3 Data Association . . . . .	24
3.4 Filter Initialization . . . . .	25
3.5 Assumed Gaussian Noise . . . . .	26
<b>Chapter 4</b>	
<b>Trajectory Generation and Control</b>	<b>27</b>
4.1 Trajectory Generation . . . . .	27
4.1.1 Touchdown Acceleration . . . . .	27
4.1.2 Deck Touchdown Prediction . . . . .	28
4.1.3 Terminal Approach Path . . . . .	29
4.1.4 Closing the Action Gaps . . . . .	29
4.1.5 Trajectory Gains . . . . .	31
4.1.6 Desired State . . . . .	31
4.2 Trajectory Following controller . . . . .	32
4.3 Attitude Commander . . . . .	33
4.3.1 Throttle Control . . . . .	33
<b>Chapter 5</b>	
<b>Hardware and Estimator Results</b>	<b>35</b>
5.1 Hardware . . . . .	35
5.1.1 Camera Calibration . . . . .	35
5.1.2 Camera Parameter Estimation . . . . .	38
5.1.3 Stewart Platform . . . . .	38
5.1.4 Vicon Motion Capture System . . . . .	39
5.2 Software Implementation . . . . .	40
5.2.1 Information Flow . . . . .	41
5.3 Estimator Results . . . . .	42
5.3.1 Repeatability and Flight Tests . . . . .	42
5.3.2 Filter Initialization . . . . .	43
5.3.3 Flights to a Static Deck . . . . .	44
5.3.4 Static Deck Results . . . . .	45
5.3.5 Moving Deck Results . . . . .	48
5.3.5.1 Hover Results . . . . .	48
5.3.5.2 Landing Results . . . . .	52
5.3.5.3 Aggregate Results . . . . .	54
5.3.6 Sources of Error . . . . .	58
5.4 conclusion . . . . .	59
<b>Chapter 6</b>	
<b>Conclusions</b>	<b>60</b>
6.1 Summary of Contributions . . . . .	60
6.2 Lessons Learned . . . . .	61

6.3	Future Work . . . . .	61
6.3.1	Future Estimation Work . . . . .	61
6.3.2	Ship Motion . . . . .	62
6.3.3	Motion Planning . . . . .	62
	<b>Bibliography</b>	<b>63</b>

# List of Figures

1.1	Lynx helicopter tracking a moving flight deck in rough seas. . . . .	2
1.2	UAVs face similar challenges to birds while landing: only measurements of relative state are certain to be available while the landing site may be driven by external forces (e.g, rough seas or wind). . . . .	2
1.3	A high level block diagram of the system. Green blocks are represent processes than occur on board the vehicle. . . . .	3
2.1	All frames used in estimation and path planning with the addition of the inertial frame. The axis of the IMU are not shown but are coincident with those of the vehicle. . . . .	9
2.2	The modified pinhole camera model which represents measurements to features as a set of bearing measurements from the optical axis of the camera. . . . .	13
2.3	Nominal position, velocity, and acceleration from tau-dot trajectories using varying values of $k$ . The trajectories start at 1 meter from the goal with a speed of 1 meter per second toward it. . . . .	16
2.4	Trajectories generated using tau-guidance starting at 1 meter from the goal with a trajectory duration of 5 seconds and varying values of $k$ . These trajectories are attractive because they start with zero velocity towards the goal and can be used for docking and landing. . . . .	17
2.5	Trajectories generated using higher order tau-guidance starting at 1 meter from the goal with a trajectory duration of 5 seconds and varying values of $k$ . These trajectories are attractive because they start with zero velocity and zero acceleration towards the goal and can be used for docking and landing. . . . .	19
4.1	This figure shows an alternate coordinate system to the $\lambda$ - $\gamma$ system for defining unit vectors between the level ship frame and vehicle. Action gaps are constructed using the $\alpha$ - $\beta$ system because they are well conditioned for level-deck landings. In this figure the vehicle (which is represented by the blue sphere) has some negative offset along each axis of the level-deck frame. . . . .	30
4.2	The desired accelerations of vehicle were found by summing the nominal accelerations from the trajectory with those generated by a virtual spring damper system fixed between the vehicle and the desired point on the trajectory. . . . .	32
5.1	The Y6 hexacopter carrying the Odroid and Machine Vision camera used to test the system presented in this thesis. . . . .	36

5.2	An unmodified picture from a camera mounted on the vehicle. The distortion in these images must be accounted for in order to convert pixel locations to bearing measurements. . . . .	36
5.3	The process of calibrating the camera using a checkerboard with known dimensions and the ROS package <i>camera_calibration</i> . . . . .	37
5.4	Multiple views of the landing platform from a typical position from which the vehicle would start a trajectory. . . . .	39
5.5	A representative sample of the motion of the deck. Each axis followed sinusoidal motion. . . . .	40
5.6	How information is passed around the system. . . . .	41
5.7	This figure shows the performance of the x position estimate of the deck in the vehicle frame. Regions where autonomy is one indicates the vehicle is flying autonomously. In some regions where the autonomy flag is low, the estimator is being reset multiple times a second which causes jaggedness of the estimate, error, and confidence interval. . . . .	42
5.8	Flights flown by the vehicle to the static deck. Trajectories were generated using three tau guidance gaps with a duration of 15 seconds with a $k$ value of .3. . . .	44
5.9	The norm of the estimate error as the vehicle approaches the deck. As expected, estimates get better as the vehicle approaches because bearing measurements contain more information relative to the noise. The error norms for individual flights are shown in grey while the average norm in 1cm bins is shown in black. . . .	45
5.10	The error in the position estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black. . . . .	46
5.11	The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black. . . . .	47
5.12	The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck. . . . .	48
5.13	The estimated Euler angles (in radians) of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck. . . . .	49
5.14	The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck. . . . .	50
5.15	The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck. . . . .	51
5.16	The estimated position and deck attitude during a flight to the moving deck. . . .	52
5.17	The estimated velocity and deck body rate during a flight to the moving deck. . .	53
5.18	Flights flown by the vehicle to the moving deck. Trajectories were generated using the method described in Chapter 4. . . . .	55
5.19	The norm of the error with individual errors shown in grey and the average error in 1cm bins shown in black for flights to a static deck. . . . .	56
5.20	The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black. . . . .	57
5.21	The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black. . . . .	57

# List of Symbols

- $\hat{\kappa}$  Estimated states will be indicated by a hat, p. 8
- $\mathbf{r}_{d/h}^I$  The position of the deck relative to the helicopter frame, p. 8
- $\mathbf{r}_d^I$  The position of the deck in the inertial frame, p. 8
- $\mathbf{r}_h^I$  The position of the vehicle in the inertial frame, p. 8
- $\mathbf{r}_{d/h}^h$  The position of the deck relative to the helicopter as seen from the helicopter frame, p. 8
- $\omega_h$  The body rates of the vehicle, p. 8
- $\omega_d$  The body rates of the deck, p. 14
- $\mathbf{T}_d$  The transformation from the inertial frame to the deck frame, p. 10
- $\mathbf{T}_h$  The transformation from the inertial frame to the vehicle frame, p. 8
- $\mathbf{T}_{cam}$  The transformation from the vehicle frame to the camera frame, p. 13
- $\mathbf{T}_g$  The transformation from the inertial frame to the vehicle frame, p. 14
- $\phi$  The third angle in a 321 Euler transformation, or roll p. 8
- $\theta$  The second angle in a 321 Euler transformation, or pitch, p. 8
- $\psi$  The first angle in a 321 Euler transformation, or pitch, p. 8
- $\mathbf{b}_i$  The position of the  $i^{th}$  feature in the deck frame, p. 13
- $\rho$  The position of the camera in the vehicle frame, p. 13
- $\mathbf{g}$  The acceleration from gravity in the inertial frame, p. 14
- $\mathbf{n}_c$  The Gaussian term used to model sensor noise , p. 13
- $\mathbf{v}$  The Gaussian term used to model system inaccuracies, p. 22
- $\tau$  The ratio between distance between the distance and velocity to a goal, p. 14
- $\mathbf{Q}$  A noise term added to the estimate covariance to capture inaccuracies in system dynamics , p. 23

**R** A noise term added to the estimate covariance to capture noise in measurements, p. 23

**P** A covariance matrix, p. 25

# Acknowledgments

There are so many people I would like to thank for making this effort possible. Jack Langelaan, my advisor, has been patient, understanding, and a constant source of guidance and wisdom. I want to thank my friends John Bird, Nate Depenenpush, and Junyi Geng. Thank you all for your companionship, patience, and always saving me the previous day's coffee.

There are countless others who have helped me along.

Thank you all.

# Dedication

This thesis is dedicated to the AVIA lab.

# Chapter 1 |

# Introduction

This thesis describes the development of a self-contained autonomous landing system that uses only a low-cost inertial measurement unit and a monocular camera. The research was motivated by the problem of landing an autonomous rotorcraft on a ship at high seas states.

Autonomous shipboard landing requires advances in flight control, relative position estimation, and trajectory planning. This thesis focuses on the problem of relative position estimation, that is, computing an estimate of the state of the landing deck (position, orientation, velocity, and angular rate) in the rotorcraft body frame. Combined with a trajectory planner and a flight control system that can follow the computed trajectory, the deck state estimator enables safe landing on a moving ship flight deck.

The problem of deck state estimation is difficult. First, the limited sensor suite means that only bearings to deck features are available. Second, deck motion is inherently stochastic, driven by the random motion of ocean waves. Third, the rotorcraft is itself moving, so there is no inertial reference frame. While the availability of an inertial reference such as GPS would add information to the deck state estimation problem, this thesis assumes that GPS is unavailable during the landing phase.

This thesis: (a) describes a framework for autonomous landing on a ship flight deck; (b) presents a solution to the deck state estimation problem based on a Sigma-point Kalman filter; (c) implements a landing trajectory generator based on tau-guidance; (d) presents results of hardware tests of landings on both static and moving decks using a small multi-rotor aircraft equipped with camera, on-board computer, and autopilot module.

## 1.1 Motivation

The task of landing on a ship is very challenging, even for human pilots. Not only is the landing deck moving in six degrees of freedom, with significant motion possible in each (see for example Figure 1.1), but in many cases the frequency and amplitude of deck motion is high enough that the vehicle's response characteristics are too slow for one to simply 'match motion' and land.

In addition, deck motion is driven by an uncertain forcing function, namely wave motion. Current practice by human pilots is generally dependent on waiting for a short quiescent period



Figure 1.1: Lynx helicopter tracking a moving flight deck in rough seas.



(a) autonomous firescout



(b) a bird

Figure 1.2: UAVs face similar challenges to birds while landing: only measurements of relative state are certain to be available while the landing site may be driven by external forces (e.g, rough seas or wind).

when touchdown can occur [1].

In the case of uninhabited aerial vehicles (UAVs) such as autonomous rotorcraft, the problem could be addressed using landing beacons or radio aids. However, this external infrastructure would preclude the ability to land on generic ships or at unprepared landing sites.

A UAV landing on an unmarked ship flight deck, shown in Figure 1.2a, faces challenges similar to a bird in the process of landing. In both cases, only relative measurements of the non-specialized landing site are certain to be available while inertial positions can not be relied upon.

## 1.2 An Overview of a Self-Contained Landing System

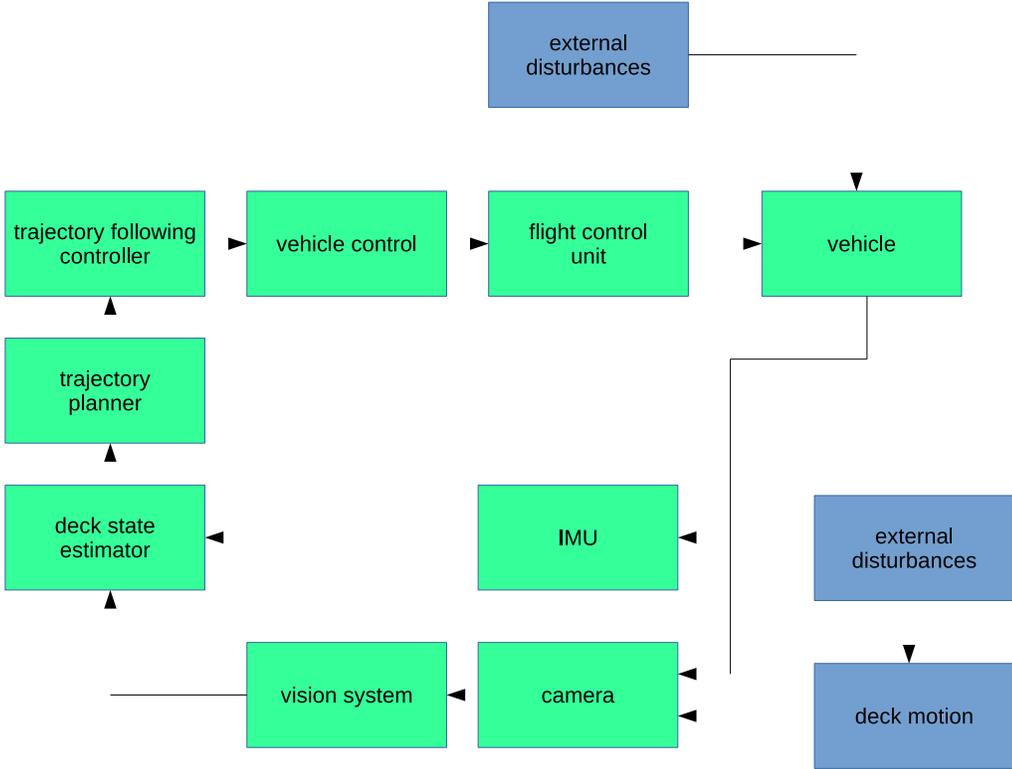


Figure 1.3: A high level block diagram of the system. Green blocks are represent processes than occur on board the vehicle.

An overview of the system presented in this thesis can be seen in figure 1.3. The IMU block represents an inertial measurement unit carried on the aircraft which generates measurements of attitude, acceleration, and body rate. The camera block, which is also carried on board the aircraft, generates measurements of the relative state of the aircraft and the deck. These images are then passed through vision processing to extract a limited set of salient features for tracking. Both sets of measurements are passed to an unscented Kalman filter which estimates the relative state of the deck and aircraft. These relative estimates are passed to the trajectory planner which generates a full desired state for the given aircraft-flight deck state. The trajectory following controller uses this desired state to generate a desired aircraft acceleration. The desired aircraft acceleration is passed to the vehicle controller which generates control inputs for the flight control unit. The flight control unit drives motors to match the desired control input from the vehicle control block. The motor forces and external disturbances drive the dynamics of the aircraft. The deck is driven largely by unknown wave motion. The ship may be controlled but here it is

assumed that ship motion is unknown.

### 1.3 Problem Description

Landing a helicopter on a ship driven by pseudo random waves is a challenging problem. The motion of the ship is largely unpredictable. Probabilistic searches of the relative state space is impracticable because of the large number of dimensions, the constantly changing starting conditions, and the short amount of time in which a trajectory would be need to be converged upon. This makes closed form trajectories, which are used in this thesis, attractive.

The other side of the problem, estimating the position of the ship, can often be complicated by lack of GPS. This creates a need for an estimator that isn't tied to the inertial frame like many other estimators that have been developed [2], [3], [4].

In addition to kinematic and dynamic constraints, there are landing constraints that trajectories must satisfy. For the vehicle to land safely the difference in roll and pitch must be small. This ensures that the large forces generated during landing will be spread across the landing gear of the vehicle. When hover attitude is deviated from, rotorcraft will accelerate in the direction in which they have tilted. This greatly complicates the problem of landing on a ship because for the vehicle to satisfy both the matching attitude constraint and the low velocity impact constraint, the vehicle must be accelerating away from the deck during landings on a non-level deck.

In addition to complicated and near unpredictable ship motion, there are often complex winds coming from the super structure of the ship.

### 1.4 Related Work

A significant amount of effort has been spent addressing the problem of autonomous ship landing during rough seas. There has also been considerable amount of work done on topics relating to tau based trajectory planning and vision based perception and state estimation.

#### 1.4.1 Ship Landing

Tau theory has previously been looked to for solving the ship landing problem. In [?], tau based control laws were developed to control the vehicle axis of a Bell 412 which resulted in natural landings while avoiding torque oscillations. Additionally, tau based control laws were developed for lateral movement for a helicopter similar to a UH60. Simulations of the lateral control laws showed they were within level 1 handling quality boundaries.

A tethered method of autonomously landing the vehicle on a deck was presented in [5]. This allowed for favorable coupling between the dynamics of the ship and helicopter. However, this approach has some drawbacks: it will require specialized decks and will be much more costly than a solution reliant that solely on estimation of ship states. Additionally, the task of attaching the tether to either the ship or vehicle introduces a complication not present in many other approaches.

An approach was presented in [6] estimated deck states using information from a camera and LIDAR mounted in a sensor suite bellow the vehicle. This approach showed that the use of both sensors resulted estimates. This was due to the complimentary nature of camera and LIDAR measurements.

Applications for vision based landings have been developed using stationary platforms [7, 8], as well as platforms with known constant velocity [9].

### 1.4.2 Monocular Vision and Perception

Vision based perception has many benefits, some of which was previously described in the previous motivation sections. The application of vison has been an of interest to the autonomous community for the use of navigation and object avoidance for some time. Benefits of vision based navigation is discussed in [10–13].

Vision based navigation is of particular interest in applications designed for GPS denied environments where navigation and control choices are designed around state information relative to fixed or dynamic environments, similar to the work presented in this thesis. The application of vision based navigation has been investigated in [14], [15], and [16]. In [17], a method of optical navigation is presented that is explicitly designed for vision based navigation.

The motivation behind this thesis, the potential gain in shipboard capabilities of UAVs and rotorcraft during high sea states, has also been explored in previous works. In [18], vision based navigation of a VTOL helicopter is explored in an airfield with methods designed to handle GPS failures and sensor latency. In [19], VTOL vehicle states relative to airport features are estimated by comparing vision and inertial measurements with knowledge of airport layouts. In [20], a fully autonomous quadrotor was presented capable of navigating indoor environments using information from a laser range finder, sonar, and inertial measurements.

A more complete solution to vision based autonomous landing was presented in [8]. This work used vision measurements to a set of known features with known locations to estimate vehicle relative states. Unlike the work presented in this thesis, [8] uses features that are known to be stationary.

In [21] and [22], the tracking and landing on a stationary and moving target by a UAV is presented using vision based navigation. Later, in [23], an approach estimating full deck motion is presented by identifying the heliport marking and extracting and Kalman filtering relative states.

Previously, Truskin, in [24], presented a work similar to this thesis where the ship contained a known set of features. Ship states where estimated in the inertial frame as it was assumed that GPS was available. The acceleration driving the ship motion was proven to be adequately captured by a zero-mean Gaussian distribution, the size of which was determined by the sea state and ship dynamics. This was tested against several methods.

This thesis uses the Munkres method of associating measured features with expected features. This algorithm was shown to have polynomial complexity in [25]. This method has previously been in sense and avoid applications [26]. It has been used to address the target-map problem [27], as well as landmark tracking [28].

## 1.5 Contributions

- **Estimator Design:** An unscented Kalman filter was designed that estimates the relative position of a ship flight deck. This system functions in GPS denied environments by fusing measurements of bearing and bearing rate to the deck with measurements of vehicle acceleration and body rates from an IMU carried on board the vehicle. The state vector being estimated consists of position, velocity, attitude, and body rate which is then used by the trajectory planner.
- **Tau Based Trajectory Planning:** Several tau based methods of generating trajectories are presented. These trajectories were first used to describe the landing trajectories followed by birds. One set of trajectories is then used to generate trajectories to the deck. A trajectory following controller is presented that commands desired acceleration. These commands are then fed to a lower level controller which calculates desired attitudes. These desired attitudes are passed to the autopilot controlling the vehicle.
- **Hardware Demonstration:** The estimator and trajectory planner were implemented on a vehicle. The behavior of the estimator was proven to be capable of providing enough information of allow for autonomous flight deck landings. 64 flights were flown to a static deck and 8 were flown to a moving deck.

## 1.6 Readers Guide

The remainder of this thesis will take the following form:

**Chapter 2:** The ship landing problem will be introduced along with all frames, kinematic relations, models, and algorithms used in later chapters.

**Chapter 3:** The construction of the estimator is presented. The problem of data association will also be addressed along with assumptions, and their validity, made while constructing the estimator.

**Chapter 4:** Trajectory Generation and Control discusses the tau based trajectories used to land on the platform, the trajectory following controller, and the landing controller.

**Chapter 5:** Results from the hardware implementation are presented. The behavior during the 64 flights to the static deck are discussed and analyzed along with the results from the flights to the moving deck. Challenges that arose while building the system are also address.

**Chapter 6:** A summary of research and the future areas of interest is discussed.

# Chapter 2 |

## The Ship Landing Problem

This chapter defines all models, tools, and methods needed to construct the system presented in this thesis. Kinematic models for motion are introduced in section 2.2. All frames necessary for estimation, trajectory planning, and control are introduced with a focus on the relative frames between ship and helicopter which are used for estimation and trajectory planning. Inertial frames will be used sporadically, and identified, but are only used due to their convenience in deriving the relations between non-inertial frames. Both camera and IMU models, which are used for estimation, are introduced in section 2.4. The method for generating trajectories, which is based on tau theory, is introduced in section 2.5. Several different tau derived methods of trajectory generation are introduced with varying starting and landing characteristics. Finally, the unscented Kalman filter will be briefly discussed in 2.6. Section 2.7 will summarize what was discussed in the chapter and will introduce the following chapter: The Design of an Unscented Kalman Filter.

### 2.1 Problem Statement

The goal of the unscented Kalman filter presented in this thesis is to generate estimates that will enable an autonomous landing of a helicopter on an uncooperative platform. The proposed method of generating trajectories presented in this chapter originates from tau theory, which describes the trajectories of natural beings and how they process visual information. The estimation and trajectory generation is accomplished by fusing information from multiple sensors carried on board the vehicle with some prior knowledge of the prominent visual features on the deck.

This problem is solved by optimally combining information from an inertial measurement unit, which provides measurements of the angular rate of the vehicle and the acceleration of the vehicle with the addition of gravity, and bearing and bearing rate measurements to the prominent deck features which are generated by a camera carried on board the vehicle and some computer vision software.

One aspect of the vision processing that is not seen in figure ?? is that assumption that all features seen by the computer vision software can be associated with features on the deck. Many approaches can be used to address this problem, known as the data association problem. In this thesis, the Munkres [25] algorithm is used to match seen features with expected features

where the cost of associating a measured feature and an expected feature is a function of the distance between measured features, expected features, and the uncertainty of the position of each expected feature. This will be discussed further in chapter 3 where the construction of the unscented Kalman filter is fully addressed.

To allow the system to function in a GPS denied environment, position, velocity, and attitude of the deck are estimated relative to the helicopter's rotating frame. The only non-relative state estimated was the body rate of the deck which is estimated in the deck's frame. This set of states, position, velocity, attitude, and body rate, is sufficient for the generation of safe trajectories to the deck under reasonable deck motions. The full state being estimated is

$$\hat{\mathbf{x}} = [x_d^h \ y_d^h \ z_d^h \ \phi_d^h \ \theta_d^h \ \psi_d^h \ u_d^h \ v_d^h \ w_d^h \ p_d^d \ q_d^d \ r_d^d]^T \quad (2.1)$$

where superscripts of states denote the frame in which they are being estimated, subscripts denote the frame that is being estimated, and  $h$  and  $d$  denote helicopter frame and deck frame respectively.

The information used to estimate the deck states is acquired from an IMU and camera on board the vehicle. It has been assumed that the translation and transformation between the vehicle and the camera is known. It is also assumed that there is no offset or transformation between the vehicle and the IMU. Neither of these assumptions is unreasonable; during the process of calibrating the camera, the pose of the camera was estimated in the vehicle frame. This will be addressed in the chapter 5. The assumption of the IMU being coincident with the vehicle frame only simplifies the appearance of measurement dynamics, it does not change their conditioning in any significant way.

This goal of this thesis is to present a method to estimate the states of a ship relative to a helicopter using a camera and an IMU, where both sensors are mounted on the helicopter in known orientations relative to the helicopter frame. The estimates are then to be used to follow trajectories generated from a tau theory to land on a moving ship flight deck. The purpose of the project is to create a self-contained system that allows a vehicle to land autonomously, in a safe manner, on a moving platform. To achieve this, an unscented Kalman filter was used to fuse information from both sensors which are then used for the task for trajectory generating and tracking.

## 2.2 System Models

### 2.2.1 Reference Frames and Kinematic Relationships

All states and reference frames estimated or used in derivations can be seen in figure 2.1. The position of the deck in the helicopter frame expressed in the inertial frame is

$$\mathbf{r}_{d/h}^I = \mathbf{r}_d^I - \mathbf{r}_h^I \quad (2.2)$$

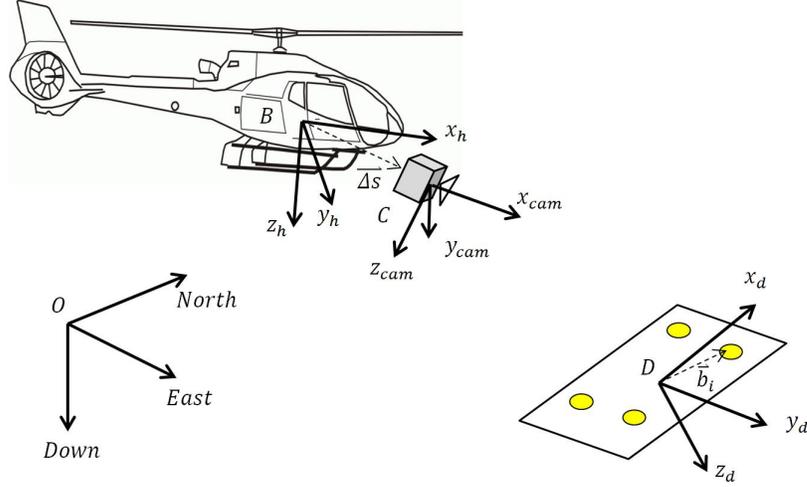


Figure 2.1: All frames used in estimation and path planning with the addition of the inertial frame. The axis of the IMU are not shown but are coincident with those of the vehicle.

Taking the derivative in the inertial frame yields

$$\dot{\mathbf{r}}_{d/h}^{\mathbf{I}} = \dot{\mathbf{r}}_d^{\mathbf{I}} - \dot{\mathbf{r}}_h^{\mathbf{I}} \quad (2.3)$$

The derivative of a vector in the rotating reference frame is

$$\dot{\mathbf{r}} = \dot{\mathbf{r}}^\omega + \omega \times \mathbf{r} \quad (2.4)$$

where the vector with the superscript  $\omega$  is the vector  $r$  as seen from the rotating frame. The velocity of the deck seen from the rotating helicopter frame is

$$\dot{\mathbf{r}}_{d/h}^{\mathbf{I}} = \dot{\mathbf{r}}_{d/h}^{\mathbf{h}} + \omega_{\mathbf{h}} \times \mathbf{r}_{d/h} \quad (2.5)$$

where  $\omega_{\mathbf{h}}$  is the body rate of the helicopter. Substituting in the rotating deck velocity in the rotating helicopter frames gives

$$\dot{\mathbf{r}}_{d/h}^{\mathbf{h}} = \dot{\mathbf{r}}_d^{\mathbf{I}} - \dot{\mathbf{r}}_h^{\mathbf{I}} - \omega \times \mathbf{r}_{d/h} \quad (2.6)$$

Taking the derivative again finding the acceleration of the deck in the rotating helicopter frame yields

$$\ddot{\mathbf{r}}_{d/h}^{\mathbf{h}} = \ddot{\mathbf{r}}_d^{\mathbf{I}} - \ddot{\mathbf{r}}_h^{\mathbf{I}} - \dot{\omega}_{\mathbf{h}} \times \mathbf{r}_{d/h} - 2\omega_{\mathbf{h}} \times \dot{\mathbf{r}}_{d/h} - \omega_{\mathbf{h}} \times (\omega_{\mathbf{h}} \times \mathbf{r}_{d/h}) \quad (2.7)$$

Note that the terms in this equation are from a mixed reference frame.

The transformation of a vector from the deck frame to the helicopter frame, assuming no

offset, is

$$\mathbf{r}^h = \mathbf{T}_h \mathbf{T}_d^T \mathbf{r}^d \quad (2.8)$$

where  $\mathbf{T}_h$  is the transformation from the inertial frame to the helicopter frame and  $\mathbf{T}_d$  is the transformation from the inertial frame to the deck frame. The transformations,  $\mathbf{T}_h$  and  $\mathbf{T}_d$ , are 3-2-1 rotations where

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.9)$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.10)$$

$$\mathbf{R}_3 = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

so

$$\mathbf{T} = \mathbf{R}_1 \mathbf{R}_2 \mathbf{R}_3 \quad (2.12)$$

The Euler angles defining the rotations are related the the body rates of the terminal frame by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.13)$$

The Euler angles that define the transformation from the deck to the helicopter are therefore related to body rates by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \left( \begin{bmatrix} p_h \\ q_h \\ r_h \end{bmatrix} - T_h T_d^T \begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} \right) \quad (2.14)$$

## 2.2.2 Vehicle Dynamics

In this thesis, hardware demonstrations were performed using a multi-rotor. For multi-rotors, the net thrust vector is always pointed in the same direction relative to the vehicle body frame (typically the frame is chosen so that  $T$  is aligned with  $-z$ .) Flight control thus occurs by tilting the vehicle (and thus the thrust vector) in the direction of desired acceleration and setting the magnitude of the thrust vector to a required value.

For the purposes of control and path planning, the vehicle in this project was treated as a second order system driven by the forces of gravity and throttle force. This assumption of vehicle

dynamics can be represented as

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ mg - m\ddot{z} \end{bmatrix} = \begin{bmatrix} -T \sin \theta \cos \phi \\ T \sin \phi \\ T \cos \theta \cos \phi \end{bmatrix} \quad (2.15)$$

Limiting angles to small values allows linearization:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ mg - m\ddot{z} \end{bmatrix} = \begin{bmatrix} -T\theta \\ T\phi \\ T \end{bmatrix} \quad (2.16)$$

Given a set of desired accelerations in x, y, and z, one can solve for the required roll, pitch, and net thrust.

In this form of the equations, a set of commanded angles and throttle can be solved for in closed form for a given set of desired accelerations.

### 2.2.3 Landing Coordinate Systems

In addition to the traditional body-carried and NED frames used in this project, two additional coordinate systems are used in the deck frame to define the approach path and desired position of the vehicle in the deck frame. The coordinate system used to define the terminal approach path of the vehicle is in the level deck body frame and is defined by azimuth, elevation, and distance. The azimuth and elevation are defined as

$$\lambda = \arctan \frac{y_{v/l_d}}{x_{v/l_d}} \quad (2.17)$$

$$\gamma = \arccos \left( \frac{z_{v/l_d}}{\sqrt{x_{v/l_d}^2 + y_{v/l_d}^2 + z_{v/l_d}^2}} \right) \quad (2.18)$$

where the distance component is simply the distance between the deck and the vehicle.

The coordinate frame used to define the trajectory of the vehicle consists of two angles,  $\alpha$  and  $\beta$ , and distance,  $\rho$ . The angles  $\alpha$  and  $\beta$  are defined respectively as the angle to the position in the x-z plane and the angle to the position in the y-z plane. The full definition of the coordinate system is

$$\alpha = \arctan 2(-z_{v/l_d}, x_{v/l_d}) \quad (2.19)$$

$$\beta = \arctan 2(-z_{v/l_d}, y_{v/l_d}) \quad (2.20)$$

$$\rho = \sqrt{x_{v/l_d}^2 + y_{v/l_d}^2 + z_{v/l_d}^2} \quad (2.21)$$

## 2.3 Landing Constraints

Due to the relation between vehicle attitude and vehicle acceleration, the terminal portion of a trajectory must satisfy several constraints to result in a safe landing. A safe landing must have the following three properties:

- There must be a small, negative velocity between the vehicle and the ship with it's speed limited by the landing gear.
- Landing occurs when the difference in position between the landing-gear of the vehicle and the desired landing location reaches zero.
- All landing gear of the vehicle comes into contact with the landing surface simultaneously within some tolerance.

For the vehicle to match attitudes with a non level surface during landing, the vehicle must be accelerating. The landing acceleration constraint can be related to the level body-fixed frame of the landing surface where the landing surface is out of level by angle  $\theta$ .

$$T = m(g - \ddot{z}_v) \quad (2.22)$$

$$\theta = -m \frac{\ddot{l}_v}{T} \quad (2.23)$$

$$acc_v = \sqrt{\ddot{l}_v^2 + \ddot{z}_v^2} \quad (2.24)$$

where  $acc$  is the total kinematic acceleration of the vehicle at touchdown and  $\ddot{l}_v$  is the lateral acceleration of the vehicle. For a landing scenario on a tilted surface, only the touchdown attitude of the vehicle,  $\theta$ , is given. The variables of  $T$ ,  $\ddot{x}$ , and  $\ddot{z}$  are currently under-determined. An additional constraint the total acceleration at touchdown and the terminal approach path of the trajectory can be defined.

## 2.4 Sensor Models

### 2.4.1 Vision Model

The camera is assumed to be offset from the body frame by the fixed known vector  $\rho$  and rotated from the helicopter frame by the known transformation  $\mathbf{T}_{\text{cam}}$ . The model used here, the modified pinhole camera model, represents features seen by the camera as bearings to the features from the optical axis of the camera. In this model, the x axis of the camera points out the optical center of the camera while the y and z axes point out the right and bottom of the camera respectively. This can be clearly seen in figure 2.2 which was taken from [14]. Therefore, the measurement of

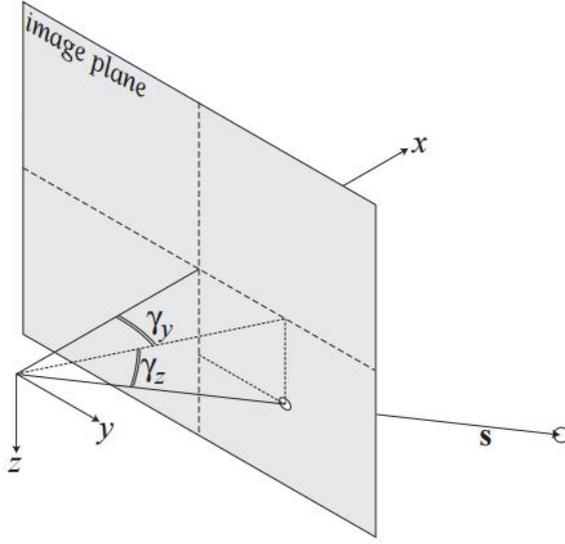


Figure 2.2: The modified pinhole camera model which represents measurements to features as a set of bearing measurements from the optical axis of the camera.

the  $i^{th}$  feature in the deck frame is

$$\mathbf{z}_s = \begin{bmatrix} \arctan\left(\frac{s_y^i}{s_x^i}\right) \\ \arctan\left(\frac{s_z^i}{s_x^i}\right) \end{bmatrix} + \mathbf{n}_c \quad (2.25)$$

where  $s_{(\cdot)}^i$  is the  $(\cdot)$  component of the vector  $\mathbf{s}^i$  which is defined by the position the  $i^{th}$  feature of the deck in the camera frame and  $\mathbf{n}_c$  is Gaussian noise corrupting the camera measurement.

The vector  $\mathbf{s}^i$  is

$$\mathbf{s}^i = \mathbf{T}_{\text{cam}}(\mathbf{r}_{d/h} + \mathbf{T}_h \mathbf{T}_d^T \mathbf{b}_i - \rho) \quad (2.26)$$

where  $\mathbf{b}_i$  is the vector to the  $i^{th}$  feature in the deck frame.

The model used for optical flow is found by taking the derivative of equation (2.25). The optical flow of the  $i^{th}$  feature is

$$\mathbf{z}_i = \begin{bmatrix} \frac{\dot{s}_y^i s_x^i - s_y^i \dot{s}_x^i}{s_x^{i2} + s_y^{i2}} \\ \frac{\dot{s}_z^i s_x^i - s_z^i \dot{s}_x^i}{s_x^{i2} + s_z^{i2}} \end{bmatrix} + \mathbf{n}_c \quad (2.27)$$

where  $\dot{\mathbf{s}}^i$  is found by taking the derivative of equation (2.26):

$$\dot{\mathbf{s}}^i = \mathbf{T}_{\text{cam}}(\dot{\mathbf{r}}_{d/h} + \omega_h \times \mathbf{r}_{d/h} - \mathbf{\Omega} \mathbf{T}_{h/d} \mathbf{b}_i) \quad (2.28)$$

where  $\mathbf{\Omega}$  is the skew symmetric cross product matrices of the relative body rates expressed in the helicopter frame

$$\mathbf{\Omega} = \text{skew}(\omega_{\mathbf{h}} - \mathbf{T}_{\mathbf{h}/\mathbf{d}}\omega_{\mathbf{d}}) \quad (2.29)$$

## 2.4.2 Inertial Measurement Model

The IMU consists of an accelerometer and a gyroscope. The accelerometer provides measurements of acceleration relative to free-fall, that is, acceleration relative to the (fixed) inertial frame with the addition of the perceived acceleration due to gravity. A measurement of acceleration will take the form

$$\mathbf{z}_a = \frac{d}{dt}\mathbf{u}_{imu} + \mathbf{T}_g\mathbf{g} + \mathbf{n}_{acc} \quad (2.30)$$

where  $\mathbf{u}_{imu}$  is the velocity of the IMU in the inertial reference frame,  $\mathbf{T}_g$  and  $\mathbf{g}$  are the transformation from the inertial frame to the vehicle frame, and  $\mathbf{n}_{acc}$  is Gaussian noise corrupting the measurement. The IMU is assumed not to be offset from the vehicle so the velocity of the vehicle is simply the velocity of the IMU. So the measured acceleration is therefore

$$\mathbf{z}_a = \frac{d}{dt}\mathbf{u}_h + \mathbf{T}_g\mathbf{g} + \mathbf{n}_{acc} \quad (2.31)$$

The gyroscope provides measurements of the helicopters body rates. The measurement model is

$$\mathbf{z}_\omega = \omega + \mathbf{n}_{gyro} \quad (2.32)$$

where  $\omega$  is  $[p \ q \ r]^T$  and  $\mathbf{n}_{gyro}$  is Gaussian noise corrupting the measurement.

These models assume that all biases and scaling factors in the sensors have been accounted for. This can be done through thermal calibration or some other method of estimation but this is not the focus of this thesis.

## 2.5 Tau Theory

The task of path planning often requires significant information about the environment, limitations and dimensions of the vehicle being planned for, and some measure of certainty of all information being used. This suite of information is not readily available to natural beings like birds, insects, and humans. The problem of planning paths is addressed with far less information, information that originates from the senses and information that was known prior. There is strong evidence in the literature [29] [30] [31] that all of those beings previously listed create and follow paths that are built around the variable

$$\tau = \frac{x}{\dot{x}} \quad (2.33)$$

where  $\tau$  is the ratio between the distance to the goal  $x$  and the velocity  $\dot{x}$  towards the goal. The literature often refers to the variable  $x$  as an "action gap." Note that the action gap is not necessarily a spatial distance: it is the distance between a current state and a final state, and this state could be position, velocity, an angle, or some other state of interest. The ratio  $\tau$  (which has

units time) is the time to contact with the goal.

The use of the scalar  $\tau$  is common in nature because:

- $\tau$  is observable from optical flow measurements;
- an animal can use  $\tau$  to safely perch or touch down without having well-defined knowledge of its state;
- speed is regulated smoothly as the distance to touchdown is reduced

is the case of autonomous vehicles, tau theory is attractive because trajectories can be written in closed form, leading to fast computation.

There are multiple types of trajectories generated by tau theory.

### 2.5.1 Tau-Dot

The a common form of  $\tau$  planning is known as tau-dot. This approach to path planning is used by pigeons [32], bats [30], and hummingbirds [29]. Trajectories are generated by setting the derivative of  $\tau$  to a constant

$$\dot{\tau} = k \tag{2.34}$$

where  $k$  is the constant that is used to tune the characteristics of the flight, and is most commonly used to describe the final state of the trajectory when the goal is reached. This differential equation, solved for position, velocity, and acceleration, yields:

$$x(t) = x_o \left( 1 + k \frac{\dot{x}_0}{x_o} t \right)^{\frac{1}{k}} \tag{2.35}$$

$$\dot{x}(t) = \dot{x}_0 \left( 1 + k \frac{\dot{x}_0}{x_o} t \right)^{\frac{1-k}{k}} \tag{2.36}$$

$$\ddot{x}(t) = \frac{\dot{x}_0^2}{x_o} (1 - k) \left( 1 + k \frac{\dot{x}_0}{x_o} t \right)^{\frac{1-2k}{k}} \tag{2.37}$$

where terms  $(\cdot)_o$  indicate the initial value of the state at the start of the trajectory. The value of  $k$  has great influence of the characteristics of the trajectory. Figure 2.3 shows the prescribed position, velocity, and acceleration for trajectories with varying values of  $k$  with an initial position of 1 meter from the goal and 1 meter per second toward it. For values of  $k$  less than 0 there is no touchdown time, the trajectory asymptotically approaches the goal. In real applications this would make for a very soft touchdown but potentially at an unknown time. This would likely to be driven by the nature of disturbances acting on the body and the capabilities of the disturbance rejection controller. For values of  $k$  between 0 and 0.5 the trajectories reach the goal in a finite time,  $t_{final} = -\tau_0/k$ , with zero acceleration and zero velocity. This region of  $k$  makes for smooth touchdowns. For values greater than 0.5, the trajectories end with either non-zero velocity or non-zero acceleration. These could be attractive if used in a docking type application where some small impact is needed to trigger a locking mechanism or activate some other device to secure the

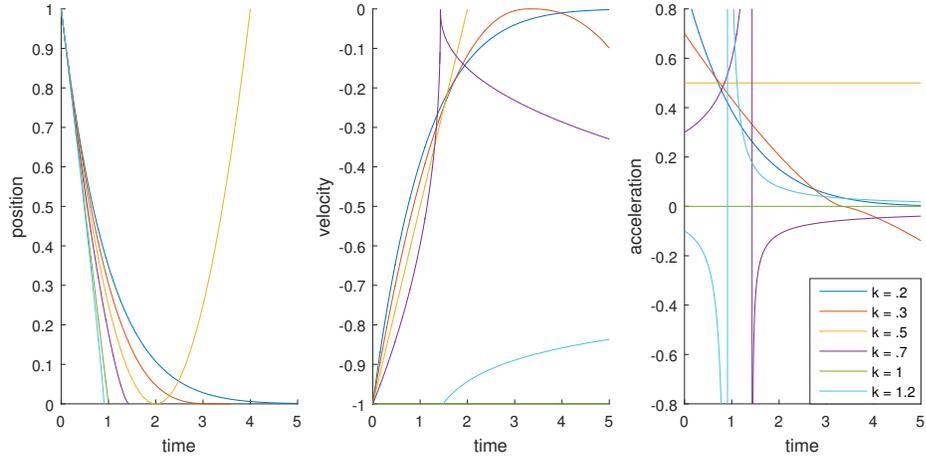


Figure 2.3: Nominal position, velocity, and acceleration from tau-dot trajectories using varying values of  $k$ . The trajectories start at 1 meter from the goal with a speed of 1 meter per second toward it.

object. For large values of  $k$ , much greater than 1, the trajectories end with large velocities and are not likely to be useful for any type of docking or landing mechanism.

While tau-dot trajectories are useful in nature and situations where probabilistic searches are not useful, they are largely used in situations with an object is attempting to come to a rest. These trajectories start with a non-zero velocity towards a goal. This is not the case with the helicopter ship landing problem, though one could imagine starting a sharp jump in acceleration towards the deck and then entering a tau-dot trajectory. However, this is an unnecessary complication when there are more fitting approach methods available. For the helicopter landing problem, a trajectory with an initial velocity of zero is needed.

## 2.5.2 Tau-Guidance

Another set of  $\tau$  based trajectories, proposed by Kendoul in [33], set the variable  $\tau$  equal to another gap that was closing. This had all the previously advantages of  $\tau$  but generated trajectories that started with initial velocities of zero. This form of equation is likely to be more useful in applications of docking or parking for vehicles that can hold a position. The trajectory proposed was

$$\frac{x}{\dot{x}} = \frac{k}{2} \left( t - \frac{T_g^2}{t} \right) \quad (2.38)$$

where  $k$  is the parameter used to tune the trajectories and the duration of the trajectory is defined by  $T_g$ . Solving the previous equations for position, velocity, and acceleration yields:

$$x(t) = \frac{x_0}{T_g^{\frac{k}{2}}} (T_g^2 - t^2)^{\frac{1}{k}} \quad (2.39)$$

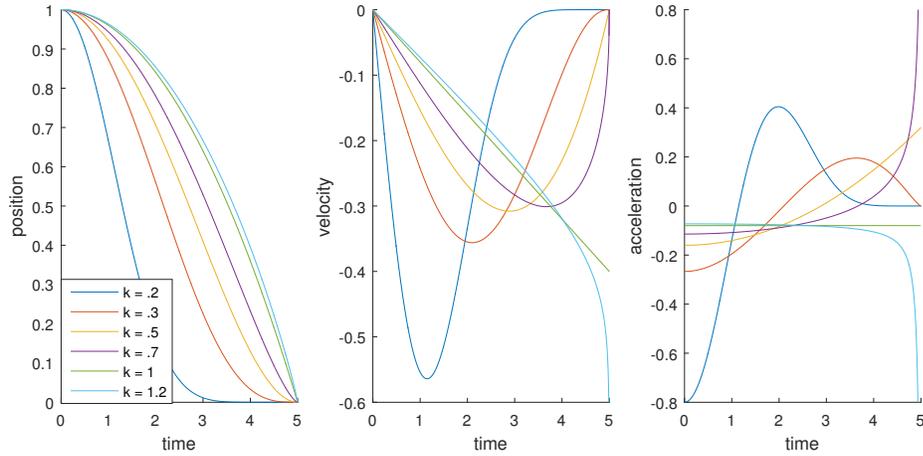


Figure 2.4: Trajectories generated using tau-guidance starting at 1 meter from the goal with a trajectory duration of 5 seconds and varying values of  $k$ . These trajectories are attractive because they start with zero velocity towards the goal and can be used for docking and landing.

$$\dot{x}(t) = -2 \frac{x_0 t}{T_g^{\frac{2}{k}} k} (T_g^2 - t^2)^{\frac{1-k}{k}} \quad (2.40)$$

$$\ddot{x}(t) = \frac{2x_0}{T_g^{\frac{2}{k}} k} \left[ \frac{2-k}{k} t^2 - T_g^2 \right] (T_g^2 - t^2)^{\frac{1-k}{k}} \quad (2.41)$$

For values of  $k$  less than zero, identically to tau-dot, there is no reaching of the destination. However, in this set of equations, there will not be a touchdown due to system imperfections because the vehicle is driven away from the destination. For values of  $k$  between 0 and 0.5, there is a touchdown, at  $t = T_g$ , with zero acceleration and zero velocity. For values of  $k$  equal to 0.5, touchdown occurs at  $t = T_g$  with zero velocity and a non-zero acceleration. For values of  $k$  greater than 0.5, touchdowns occur at time  $T_g$  with either accelerations or velocities approaching infinity. A series of trajectories were generated starting from 1 meter from there goal with zero initial velocity using varying values of  $k$  and a trajectory duration of 5 seconds. These can be seen in figure 2.4.

While tau-guidance trajectories can be more useful than the tau-dot trajectories explored in the previous section, they are still of little use during situations where a fast trajectory is needed and accelerations can not be instantaneously generated. Note that all trajectories start with a non-zero acceleration. For vehicles whose acceleration is largely determined by attitude like those used in this thesis, these instantaneous accelerations can not be achieved. For tasks like docking where there may be a limited window when the trajectory is useful, any lag in the system due to those higher order dynamics may endanger the vehicle at touchdown. For that reason, the following section will discuss higher order tau-guidance approach that starts with zero acceleration.

### 2.5.3 High Order Tau-Guidance

The trajectory proposed by Kendoul in [33] is attractive because it does not require an initial velocity. However, these trajectories prescribe an initial non-zero acceleration. For slow trajectories this may not be an issue where the settling time of the controller is fast compared to the total time of the trajectory. However, with more aggressive trajectories, with durations of the order of seconds, it may be problematic that the vehicle is not capable of perfect tracking throughout the trajectory. To address this, a new set of trajectories is presented in this thesis.

By slightly changing the gap closing proposed by Kendoul in [33], a new trajectory is proposed here which starts with zero velocity and acceleration that is trackable with vehicles that can not achieve near instantaneous changes in acceleration. The proposed trajectory generating relation is

$$\tau = \frac{k}{3} \left( t - \frac{T_g^3}{t^2} \right) \quad (2.42)$$

where  $k$  is the parameter used to tune for ideal touchdown parameters and  $T_g$  is the duration of the trajectory. Solving this relation for position, velocity, and acceleration yields:

$$x(t) = \frac{x_0}{T_g^{3/k}} (T_g^3 - t^3)^{\frac{1}{k}} \quad (2.43)$$

$$\dot{x} = -\frac{x_0}{T_g^{3/k}} \frac{3t^2}{k} (T_g^3 - t^3)^{\frac{1-k}{k}} \quad (2.44)$$

$$\ddot{x} = \frac{x_0}{T_g^{3/k}} \frac{3}{k} t (T_g^3 - t^3)^{\frac{1-k}{k}} \left[ 3t^3 \frac{1-k}{k} (T_g^3 - t^3)^{-1} - 2 \right] \quad (2.45)$$

Similar to the previous two sets of trajectories, different ranges of  $k$  have distinct touchdown characteristics. For values of  $k$  less than zero there is no theoretical touchdown. For values of  $k$  between 0 and .5 touchdowns have zero velocity and zero acceleration, and for values of  $k$  greater than .5 there is a non-zero velocity or acceleration, an impact, at touchdown. For trajectories with a  $k$  of .5, the touchdown velocity is zero but there is a finite touchdown acceleration. This is useful because landing on a non-level surface requires an acceleration when the acceleration of the vehicle is driven by the attitude of the vehicle. The acceleration prescribed at touchdown for trajectories where  $k$  is .5 is

$$\ddot{x} = \frac{x_0 18}{T_g^2} \quad (2.46)$$

So for a trajectory that started one meter above the destination and allowed 5 seconds for travel would have a nominal acceleration of  $.72 \frac{m}{s^2}$ . This is an entirely reasonable acceleration for many hovering vehicles. Assuming the vehicle is exerting the force needed to counteract gravity in hover, then a 4.2 degree offset from level will produce this acceleration.

Figure 2.5 shows trajectories for varying values of  $k$  with an initial distance of 1 meter to the goal with zero velocity and acceleration.

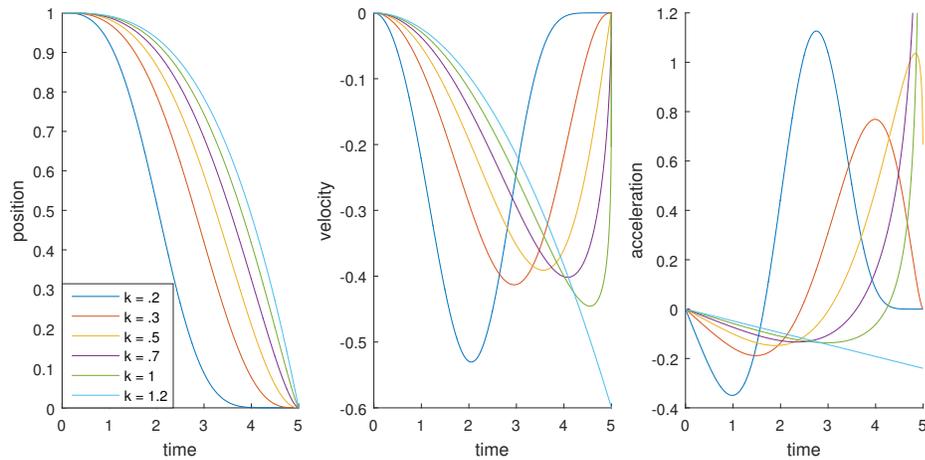


Figure 2.5: Trajectories generated using higher order tau-guidance starting at 1 meter from the goal with a trajectory duration of 5 seconds and varying values of  $k$ . These trajectories are attractive because they start with zero velocity and zero acceleration towards the goal and can be used for docking and landing.

## 2.6 Unscented Kalman Filter

In this project an unscented Kalman filter, also known as a sigma point Kalman filter, was used to estimate the state of the ship relative in the helicopter frame. The advantage of an unscented Kalman filter over a traditional Kalman filter and an extended Kalman filter is its native ability to handle nonlinear dynamics. The construction of the filter will be fully discussed in the following chapter. However, it is worth mentioning briefly here that the models which were previously discussed in sections 2.2.1 and 2.4 are used in the prediction and correction steps and are highly nonlinear.

The UKF is able to natively handle nonlinear dynamics by representing the uncertainty in the estimate as a set of points, known as sigma points. These points are passed through the nonlinear dynamics, after which, the Gaussian is recomputed by the new distribution of the points.

## 2.7 Summary: The Ship Landing Problem

The autonomous flight deck landing, like all autonomous systems, consists of two major components: estimation and control. In this chapter, all information needed to construct an unscented Kalman filter was presented. The kinematic relationship between the ship and the vehicle was developed in relative states. The sensors carried on board the vehicle, the camera and IMU, had models developed.

The task of planning trajectories was discussed. Tau theory, which is used to describe the manner in which natural beings using visual information for motion planning, was introduced. Tau-dot which produces trajectories that keep the derivative of the ratio between the distance to

a destination and the velocity towards it constant was discussed. Two other families of trajectories, both of which were generated from tau theory, were introduced. The first, tau-guidance, was introduced as a more practical solution than tau-dot in applications where a vehicle is transiting between two stationary states. The large drawback of this set of trajectories was the assumption that instantaneous changes in acceleration could be made. To address this, the higher order tau-guidance set of trajectories were introduced. These allowed transitions between two stationary states with zero starting velocity and acceleration. Additionally, by tuning a parameter, touchdown could be achieved with zero velocity and a small non-zero acceleration. This was found to be attractive as landing with a non-zero attitude for vehicles whose acceleration is driven by attitude (eg quadcopter, helicopter) necessitates an acceleration.

In the following chapter, the full unscented Kalman filter will be developed using the models presented in this chapter.

# Chapter 3 | Building a Unscented Kalman Filter

This chapter fully develops the UKF used to estimate the relative ship states. The UKF is a recursive algorithm that has two parts: a prediction step and a correction step. The prediction step uses the previous estimate and covariance to propagate the expected state forward through the system dynamics. In this application, this prediction step is triggered by a set of measurements from the IMU. The correction step uses measurements to correct the estimate from the prediction step. Here, the correction step uses measurements of bearing and bearing rate to the known prominent visual features on the deck. Both steps of the algorithm, along with the assumed inaccuracies and uncertainties of each step, will be fully developed in the chapter.

The choice of states being estimated will be discussed. Some consideration will be given to alternate sets of states with a brief qualitative comparison between advantages and disadvantages. Later, the problem of data association, which was briefly mention in chapter 2, will be fully discussed. The initialization of the filter will be covered. Finally, the validity of using a Gaussian distribution to represent the inaccuracies of models and errors corrupting measurements will be discussed.

## 3.1 Sigma Point Filter Steps

Kalman filters use a recursive process consisting of a prediction step followed by a correction. In essence, the state estimate produced by a Kalman filter is the weighted average of the prediction and correction, with the weight factor computed based on the relative confidence in each step.

Typically, the prediction step is computed using a dynamic model of the system and is driven by knowledge of the inputs. The correction step is done using measurements. In this particular case, the prediction step is driven by a kinematic model of the vehicle and measurements from the IMU are used in place of control inputs. The correction step is done using measurements from the vision system.

### 3.1.1 Prediction Step

The prediction step is triggered by a new set of measurement from the IMU carried on board the vehicle. These are used to integrate the state of the deck relative to the vehicle frame. The kinematic relationship between the vehicle and the deck, which was derived in chapter 2, is

$$\dot{\mathbf{r}}_{d/h}^h = \ddot{\mathbf{r}}_{d/h}^I - \dot{\boldsymbol{\omega}} \times \mathbf{r}_{d/h} - 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_{d/h} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{d/h}^h) \quad (3.1)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \left( \begin{bmatrix} p_h \\ q_h \\ r_h \end{bmatrix} - T_h T_d^T \begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} \right) \quad (3.2)$$

where  $\dot{\boldsymbol{\omega}}$  is found from taking the numerical derivative of the body rate measurements. In the previous equation, the term  $\ddot{\mathbf{r}}_{d/h}^I$  is the difference between the acceleration of the vehicle in the inertial frame and the acceleration of the deck in the inertial frame, or

$$\ddot{\mathbf{r}}_{d/h}^I = \ddot{\mathbf{r}}_d^I - \ddot{\mathbf{r}}_h^I \quad (3.3)$$

The term  $\ddot{\mathbf{r}}_h^I$  is being measured by the IMU with the addition of gravity. Therefore

$$\ddot{\mathbf{r}}_{d/h}^I = \ddot{\mathbf{r}}_d^I - (\mathbf{z}_{imu} - \mathbf{T}_g \mathbf{g}) \quad (3.4)$$

The only remaining unknown term is  $\ddot{\mathbf{r}}_d^I$ , or the ship acceleration in the inertial frame. To accurately represent this term, a significant amount of information is needed. The dimensions and shape of the hull of the ship, ship moments of inertial, with estimates of the surface of the ocean in contact with the ship are all needed to accurately predict ship motion. An alternate approach, which was presented by Truskin [24] and is used here, assumes zero ship acceleration. While this assumption is patently false, it does result in favorable characteristics of the system in terms of ease of implementation and general applicability. All that is needed to deploy the system is to quantify how the ships acceleration deviates from this assumption. This can be done with greater knowledge of the ship and sea state or by observing the ship motion for some time before starting the estimator. The goal of this approach is to "capture" the expected range of accelerations using the process noise. The performance of the estimator, which will be discussed in chapter 5, is not significantly hampered by this assumption. The resulting relation for relative acceleration is

$$\dot{\mathbf{r}}_{d/h}^h = \ddot{\mathbf{r}}_{d/h}^I - \dot{\boldsymbol{\omega}} \times \mathbf{r}_{d/h} - 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_{d/h} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_{d/h}^h) \quad (3.5)$$

The kinematics explored in the previous few equations can be rewritten in discrete form as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v} \quad (3.6)$$

where  $\mathbf{v}$  is the noise and uncertainty, assumed Gaussian, corrupting the measurement and capturing the uncertainty of the dynamics. The size of this term is driven by two sources: the IMU and ocean state. The IMU measurement, which triggers the prediction update, will have some

covariance associated with the signal. The influence this noise has on the system is a function of state of the system. By taking the Jacobian of the system with respect to the set of inertial measurements, the impact of the IMU covariance can be found.

The second term driving the uncertainty of the system dynamics is the wave motion forcing ship acceleration. As discussed, it is assumed the ship is undergoing zero acceleration. The covariance of ship acceleration is needed for an accurate understanding of how well these dynamics capture ship motion.

In Truskin [24], which looked at estimating ship position using prominent visual features in inertial space, the expected covariance of ship motion was derived from an assumption that ship motion could be described by sum of sinusoidal motion. For a given sea state, the nominal sinusoid had a magnitude and frequency, both of which had some deviation. By taking the second derivative of motion, the peak acceleration for each axis was found to be less than

$$(A + 3\sigma_A)(\omega + 3\sigma_\omega)^2 \quad (3.7)$$

Now the expected uncertainty in the system dynamics is defined, the process noise can be constricted:

$$\mathbf{Q} = \mathbf{F}_z \boldsymbol{\Sigma}_{\text{imu}} \mathbf{F}_z^T + \mathbf{F}_{\text{ship}} \boldsymbol{\Sigma}_{\text{ship}} \mathbf{F}_{\text{ship}}^T \quad (3.8)$$

### 3.1.2 Correction Step

The correction step is triggered by a new set of bearing and bearing rate measurements from the camera and vision processing software. The modified pinhole model that was described in chapter 2 was used as the measurement dynamics. This model can be described compactly as

$$\mathbf{z}_{\text{cam}} = \mathbf{g}(\mathbf{x}) + \mathbf{n}_c \quad (3.9)$$

where  $n_c$  is Gaussian noise,  $\mathcal{N}(0, R)$ , corrupting the measurements. The assumption of Gaussian noise relies on accurate knowledge of the camera pose in the vehicle frame and a proper calibration of the camera. The process of addressing both of these will be explained further in Chapter 5 which covers the results from hardware and the challenges that arose.

For any correction update, the set of measurements from the vision processing takes the form

$$\mathbf{z}_{\text{cam}} = \left[ \begin{array}{c} \left[ \begin{array}{c} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{array} \right]_1^T, \left[ \begin{array}{c} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{array} \right]_2^T, \dots, \left[ \begin{array}{c} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{array} \right]_n^T \end{array} \right] \quad (3.10)$$

for the measurement of  $n$  features. Features that were not seen in consecutive frames have zero measured optical flow. This information is used during the data association portion of the measurement update where the optical flow of new features is treated as an unmatched flow and

discarded.

The noise corrupting each bearing measurements is assumed to be  $\mathcal{N}(0, 4deg^2)$  and the noise corrupting each bearing rate measurement was  $\mathcal{N}(0, 4\frac{deg^2}{sec})$ .

## 3.2 Choice of States

The states estimated in this project was one possible set out of many. The choice to estimate the deck in the vehicle body frame was made for two equally important reasons. The task of converting estimated deck states in the vehicle fixed coordinate system into states that can be controlled around is straight forward. Alternate states and coordinate systems, such as estimating the deck in the vehicle frame using polar coordinates, can be significantly more complicated and error prone than the system used here. Using Cartesian coordinates to estimate the relative pose was chosen because the trajectory following controller controls the vehicle and paths planned in Cartesian space. Using a vehicle body fixed coordinate system was chosen because the IMU and camera used in the Kalman filter are fixed to the vehicle. A coordinate frame that is fixed relative to the sensors makes identifying errors in relative pose of the sensors and other sensor related errors much easier. A bias in a single axis of the accelerometer will result in a constant bias in the estimated position and velocity of the deck. The result of bias in a single axis of the IMU while using a deck fixed coordinate system would be a function of the relative pose of the deck. Using coordinate frames not fixed to the vehicle will make the symptoms of sensor related issues inherent to hardware less evident.

## 3.3 Data Association

Until this point the data provided by the vision processing was treated as usable by the UKF without any additional treatment. The need to match or associate the measured feature bearings with those belonging to the known set fixed to the flight deck will be addressed here. This process of matching expected data to measured data is called data association. For this task, the Munkres assignment algorithm [25] was implemented to match measured lights with expected lights. The Munkres assignment algorithm was attractive because it solves assignment tasks in polynomial time so a large number of features could be tracked without the need for a herculean computer. It also allows for a less than conservative set of features when the system is applied to ships; small features that may be unobservable for most of the flight may be included with no great cost incurred.

The assignment algorithm was implemented to handle any of three situations well. The first case is when the vision processing picks up the full set of known lights on the flight deck with zero extraneous features. The assignment algorithm needs to match the measured features correctly with the expected lights. The measured lights will be corrupted by noise (assumed to be Gaussian) and will be a measurement of the relative pose. This means the assignment algorithm must not reject lights within a reasonable distance from the expected lights. The second case is when the vision processing measures the correct number of features but one or more of the features is not a

measurement of the known set of lights fixed to the flight deck. Measured lights outside of this distance from expected lights must be rejected. The third case is when the vision processing measures of a set of features whose number does not match the number of expected measured features. The assignment algorithm must be able to handle a rectangular cost matrix.

In order for the Munkres assignment algorithm to be used, a cost of assignment was needed. The Mahalanobis distance was used to generate the cost of assigning an expected feature to a measured feature. To assign the  $i^{th}$  measured light to the  $j^{th}$  expected light the cost would be

$$C_{i,j} = (z_i - z_j)^T P (z_i - z_j) \quad (3.11)$$

where P is the covariance of the expected light. This means when the UKF is in the process of latching on to the relative deck states it is easier to match light measurements as long as the filter is started with a good initial starting state estimate and initial covariance than some cost function that was not informed by the state covariance. This allows the system to reject extraneous measurements robustly while there is a high confidence of the deck location while allowing the system to grab lights while the system is still converging.

To reject the incorrect light even while the number of expected lights matches the number of lights measured, a maximum Mahalanobis threshold was set. This was done by adding a dummy expected feature for each measured light the with maximum prescribed cost. With the dummy features, the full cost matrix passed to the Munkres algorithm looks like

$$\mathbf{C}_{\text{ost}} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{\text{dummy}} & 2 * C_{\text{dummy}} & 2 * C_{\text{dummy}} \\ C_{2,1} & C_{2,2} & C_{2,3} & 2 * C_{\text{dummy}} & C_{\text{dummy}} & 2 * C_{\text{dummy}} \\ C_{3,1} & C_{3,2} & C_{3,3} & 2 * C_{\text{dummy}} & 2 * C_{\text{dummy}} & C_{\text{dummy}} \end{bmatrix} \quad (3.12)$$

where the number of rows is the number of measured lights and the number of columns is equal to the number of lights plus the number of measured lights. The Mahalanobis distance and the dummy features allow the assignment system to function while meeting all requirements. Measured lights that go unmatched are discarded after the association process.

### 3.4 Filter Initialization

The initialization of an unscented Kalman filter used to estimate ship states in inertial space was explored in [24]. In that work it was assumed the ship would send a single message to the system. This message contains ship position, heading, velocity, and sea state. The starting estimate was

$$\hat{\mathbf{x}} = [x_d^I \ y_d^I \ z_d^I \ 0 \ 0 \ \psi_d^I \ u_d^I \ v_d^I \ 0 \ 0 \ 0 \ 0] \quad (3.13)$$

The states of deck body rate and vertical velocity are left unknown and initialized at zero.

The initial covariance used in [24] was driven by sea state with no assumed correlation between states. The initial covariance for each state was set to the root mean square of the expected

amplitude from wave motion. The initial covariance for the states position and attitude were

$$\mathbb{P}_{0(\cdot)} = \frac{A_{nom,(\cdot)}^2}{2} \quad (3.14)$$

while the initial covariance for velocity and angular rate was

$$\mathbb{P}_{0(\cdot)} = \frac{A_{nom,(\cdot)}^2 \omega_{nom,(\cdot)}^2}{2} \quad (3.15)$$

### 3.5 Assumed Gaussian Noise

Throughout this chapter the noise corrupting measurements has been presented as Gaussian or something whose distribution could be adequately represented as a Gaussian without resulting in constant biases cropping up in the estimates. There are several terms in the models which, if they were not accurately calibrated for or measured, would result in constant biases in the estimates. The most sensitive of these is the assumption that the pose of the camera is known in the vehicle frame. While the accuracy of the estimate is not terribly reliant of the position of the camera in the vehicle frame, it is very sensitive to errors in the cameras attitude with respect to the vehicle frame. When the vehicle is hovering about the deck, undergoing no acceleration relative to the inertial frame, there will be a constant bias in the estimates resulting from errors in the assumed attitude of the camera. While these errors do shrink as the vehicle approaches the deck, the more troubling result is the difference between expected measurements and real measurements during aggressive vehicle motions. If the estimate of the deck state converges, with low covariance, before a aggressive maneuver, there is a significant chance that some lights will be rejected by the assignment algorithm due to the errors in the measurement function.

Another potential source of error, which caused trouble during hardware tests, was biases in the IMU. When the vehicle is starting trajectories, biases in the acceleration measurements can cause constant bias in the estimates. Biases in the angular rate measurements are less problematic because the vision update provides very accurate measurements of the direction of the deck which is the information biases from the gyroscope would be corrupting. However, biases in the accelerator in the direction of the deck are able to significantly and consistently able to pull the estimate off of the truth. However, like the errors in the camera pose, these biases do shrink as the vehicle approaches the deck as the bearing measurements contain more information.

None of these sources of error invalidate the system presented in this chapter, this is discussed in the interested of pointing out potential sources of error and the consequences of assumptions made in this chapter.

# Chapter 4 | Trajectory Generation and Control

Generating trajectories to a moving deck is a complex problem. This is the result of several constraints of vehicle dynamics, touchdown state constraints, and the system's lack of knowledge of future deck state. This chapter introduces how trajectories were generated and the control strategy used to track trajectories.

In this chapter, a method of generating tau based trajectories will be presented. Decisions made designing the trajectory generation process will be discussed along with alternate choices that could have been made but were deemed unattractive. Additionally, controllers used to follow these trajectories will be presented.

## 4.1 Trajectory Generation

Trajectories generated to the moving deck were required to meet the following constraints:

- Trajectories must terminate with zero relative velocity between the vehicle and the deck.
- The terminal portion of the trajectory must allow the vehicle to achieve zero relative velocity while matching attitudes with the deck.
- Trajectories must have a method of predicting the attitude of the deck at touchdown.

The methods used to satisfy these constraints will be covered in the follow subsections.

### 4.1.1 Touchdown Acceleration

The constraint of matching attitudes between the deck and the vehicle leads to constraints of the nominal touchdown acceleration of the vehicle. This follows from the relation between vehicle attitude and vehicle acceleration laid out in Chapter 2. For a deck attitude out of level by angle  $\phi_{deck}$ , the set of relations between  $\phi_{deck}$  and vehicle accelerations are

$$T = m(g - \ddot{z}_v) \quad (4.1)$$

$$\phi_{deck} = -m \frac{\ddot{l}_v}{T} \quad (4.2)$$

$$acc_v = \sqrt{\ddot{l}_v^2 + \ddot{z}_v^2} \quad (4.3)$$

where  $\ddot{l}_v$  is the lateral acceleration of the vehicle,  $\ddot{z}_v$  is the vertical acceleration of the vehicle, and  $T$  is the commanded thrust. This relation shows the minimum touchdown acceleration to be a function of deck attitude; for a level deck the vehicle is capable of landing with matching attitudes with no acceleration while a vehicle landing on a non-level deck must have a non-zero lateral acceleration.

In this project the nominal touchdown acceleration of the vehicle was set to  $2.5 \frac{m}{s^2}$ . This allows the vehicle to match attitudes with decks that are out of level by  $15^\circ$  and satisfies the matching attitude constraint of a safe landing.

#### 4.1.2 Deck Touchdown Prediction

In order to compute a safe terminal approach path, an estimate of deck attitude at touchdown is needed.

This prediction is made by assuming that the deck's angular rate remains constant for a short period of time and then projecting the deck state forward for the time left in the trajectory. This is done using the Euler-Rodriguez formula. This formula converts an axis of rotation and a magnitude of rotation into a quaternion. The Euler-Rodriguez formula is

$$a = \cos(\phi/2) \quad (4.4)$$

$$b = k_x \sin(\phi/2) \quad (4.5)$$

$$c = k_y \sin(\phi/2) \quad (4.6)$$

$$d = k_z \sin(\phi/2) \quad (4.7)$$

where  $\phi$  is the magnitude of rotation and the vector  $[k_x, k_y, k_z]$  is the axis of rotation. In this application the magnitude of rotation was found to using

$$\phi = t * \|\omega_{deck}\| \quad (4.8)$$

where  $t$  is time until touchdown. The axis of rotation was found to be

$$k_i = \frac{\omega_{d,i}}{\|\omega_d\|} \quad (4.9)$$

where  $k_i$  is the  $i^{th}$  component of the axis of rotation.

### 4.1.3 Terminal Approach Path

Trajectories are planned from the nominal starting position to the projected desired terminal state using the acceleration constraint and the deck attitude prediction scheme. For a safe landing, the vehicle must match attitude with the deck and must land with near-zero relative velocity. For a given projected attitude of the deck, the relation between deck attitude and vehicle dynamics in the level deck frame is

$$\begin{bmatrix} \frac{-\ddot{x}}{T} \\ \frac{\ddot{y}}{T} \end{bmatrix} = \begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} \quad (4.10)$$

where  $\theta_d$  and  $\phi_d$  are the roll and pitch of the deck and  $T$  is the throttle force acting on the vehicle. Because the vehicle will reach the deck with near-zero velocity, the terminal acceleration must be opposite sign and in line with the terminal velocity of the vehicle. With these two relations, the azimuth of the terminal approach path,  $\lambda$ , can be defined:

$$\lambda = \arctan \frac{\ddot{y}}{-\ddot{x}} \quad (4.11)$$

or

$$\lambda = \arctan 2(\theta_d, \phi_d) \quad (4.12)$$

where the  $\lambda$  is undefined with a level deck as horizontal component of the trajectory is zero. With a known total acceleration at touchdown and the necessary lateral acceleration, the elevation of the terminal approach path,  $\gamma$ , can be found

$$\gamma = \arccos \left( \frac{\ddot{z}_v}{\ddot{l}_v} \right) \quad (4.13)$$

These two angles,  $\gamma$  and  $\lambda$ , define the terminal approach path of the vehicle for a safe landing.

### 4.1.4 Closing the Action Gaps

Action gaps are now used to generate trajectories to the deck using the nominal starting state and terminal approach path and terminal acceleration. Trajectories are defined using two angular distance gaps (i.e. the error between the vehicle current approach path and the desired approach path) and a distance gap (i.e. the distance from the vehicle to the touchdown point). In addition to closing the distance to touchdown, the distance gap is used to generate the terminal acceleration to ensure that the vehicle attitude matches deck attitude at touchdown.

The first set of angle gaps investigated were those that defined the terminal approach path,  $\lambda$  and  $\gamma$ . These were a natural first choice as it would be a simple extension from the touchdown path planner to generate the tau gaps. However, there are two disqualifying characteristics about these gaps. During the vertical landing case in which the deck is level and static,  $\lambda$  is not defined. The second, and more troubling characteristic is that the vehicle would clearly follow non-optimal trajectories. The tau gaps are only able to decrease. This would lead to cases where the deck crossed near level which would result in trajectories that pirouette the deck. A much shorter path would be to simply fly over the deck and to enter onto the final approach path. However this

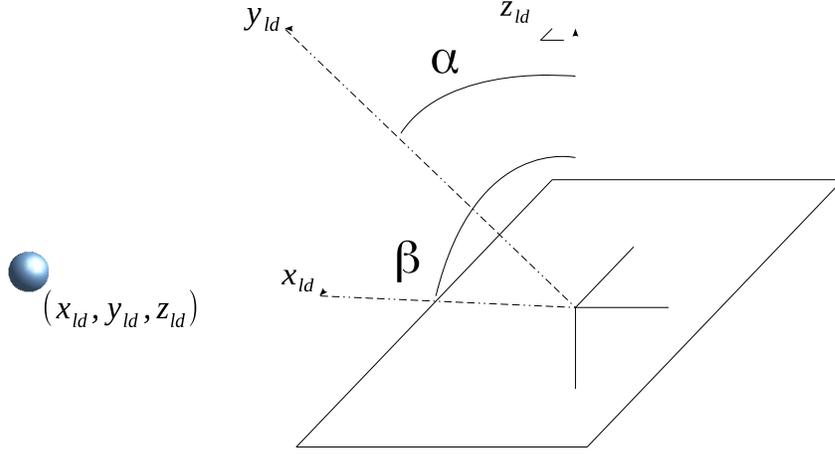


Figure 4.1: This figure shows an alternate coordinate system to the  $\lambda$ - $\gamma$  system for defining unit vectors between the level ship frame and vehicle. Action gaps are constructed using the  $\alpha$ - $\beta$  system because they are well conditioned for level-deck landings. In this figure the vehicle (which is represented by the blue sphere) has some negative offset along each axis of the level-deck frame.

would not be possible as the  $\gamma$  gap would shrink and then grow.

In this project, the three gaps that were closed using higher order tau trajectories were  $\alpha$ ,  $\beta$ , and distance to the deck. The angle  $\alpha$  is the angle to the vehicle when looking at the system from rear of the ship or

$$\alpha = \arctan 2(y_{v/ld}, -z_{v/ld}) \quad (4.14)$$

where  $\alpha$  is zero when the vehicle is straight above the deck. The second angle chosen to generate trajectories was the angle  $\beta$ . This is the angle to the vehicle from the deck when looking at the system from the port of the ship, or

$$\beta = \arctan 2(x_{v/ld}, -z_{v/ld}) \quad (4.15)$$

where  $\beta = 0$  when the vehicle is straight above the ship. The angular action gaps of  $\alpha$  and  $\beta$  are defined by the angular distance between the nominal angular starting position and the terminal angular position. The angles  $\lambda$  and  $\gamma$  are still used to calculate the terminal flight path, however due to their poor action gap characteristics, action gaps are constructed in the  $\alpha$   $\beta$  space.

Now that the final position and approach path of the trajectory is known, full trajectories can be determined.

### 4.1.5 Trajectory Gains

The three gaps defining the trajectory to the deck are  $\alpha$ ,  $\beta$ , and distance. The  $k$  values for these trajectories define how aggressively the vehicle is commanded to approach the deck. The distance gap has a predetermined  $k$  of 0.5 due to the acceleration during landing. However, the values of  $k$  for the  $\alpha$  and  $\beta$  gaps are only constrained to  $k = (0, 0.5)$ . This ensures that the velocities and accelerations of these gaps will be zero at touchdown. In Chapter 2 it was shown that as the  $k$  value becomes bigger, the gap closes more aggressively but closes closer to the end of the trajectory. In this project, the gaps  $\alpha$  and  $\beta$  used  $k_{\alpha, \beta} = 0.45$  to ensure that the projected touchdown attitude of the deck early in the landing process would not have as significant an impact on the trajectory which the vehicle followed. This also ensured the the vehicle would travel less distance as inaccurate deck predictions would not greatly affect the trajectory early on. However, the larger values of  $k$  meant that there were higher prescribed velocities and accelerations of the vehicle compared to lower values of  $k$ . It also meant that the vehicle was likely to track the trajectory better as is was happened over a smaller amount of time with greater information of the touchdown deck attitude.

### 4.1.6 Desired State

Using the gaps of  $\alpha$ ,  $\beta$ , and  $\rho$ , the desired state of the vehicle can now be calculated. In the following equations, gap distances, velocities, and accelerations are a function of time elapsed since the start of the trajectory. The relation between time elapsed and gap state can be found in Chapter 2.

To generate desired position, initially a unit vector pointing to the desired position is constructed in the level deck frame

$$u_{unit} = [x, y, z] / \|[x, y, z]\| \quad (4.16)$$

Now the desired position in the level deck frame is constructed

$$x_{des,ld} = u_{unit} * gap(t) \quad (4.17)$$

Desired velocity of the vehicle in the level deck frame is

$$v_{des,ld} = u_{unit} * \dot{\rho}(t) + \omega \times x_{des,ld} \quad (4.18)$$

where  $\omega$  is the changing angles of the trajectory in the level deck frame, or

$$\omega = [\dot{\alpha}(t), \dot{\beta}(t), 0] \quad (4.19)$$

The desired acceleration of the vehicle in the level deck frame is

$$a_{des,ld} = \ddot{\rho}(t) * u_{unit} + 2\omega \times \dot{\rho} * u_{unit} + \omega_d \times x_{des,ld} + \omega \times (\omega \times x_{des,ld}) \quad (4.20)$$

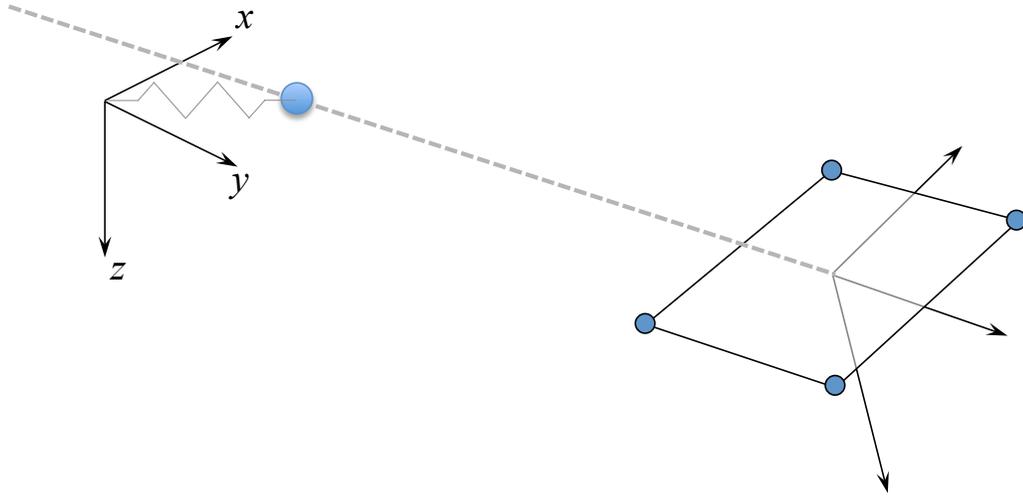


Figure 4.2: The desired accelerations of vehicle were found by summing the nominal accelerations from the trajectory with those generated by a virtual spring damper system fixed between the vehicle and the desired point on the trajectory.

where  $\omega_d$  is

$$\omega = [\ddot{\alpha}(t), \ddot{\beta}(t), 0] \quad (4.21)$$

Now desired states are converted to deck frame.

$$x_{des,d} = T_{ldtd} x_{des,ld} \quad (4.22)$$

$$v_{des,d} = T_{ldtd} v_{des,ld} - \omega_{deck} \times x_{des,d} \quad (4.23)$$

$$a_{des,d} = T_{ldtd} a_{des,ld} \quad (4.24)$$

The states from these equations are then passed to the trajectory following controller.

## 4.2 Trajectory Following controller

A trajectory following controller was constructed by attaching a virtual spring damper system between the vehicle and the nominal position on the trajectory. This was attractive because it allowed for the turning of the controller, by means of choosing the spring and damper constants, so that the controller had suitable settling times and oscillations. Figure 4.2 depicts the trajectory following controller used here. The controller also had a feed forward component in addition to the feedback component. The feedforward component was the nominal acceleration at the desired position on the trajectory found in equation 4.24. So the final acceleration command took the

form

$$\ddot{x}_{cmd} = K(x_{des} - x_{est}) + B(\dot{x}_{des} - \dot{x}_{est}) + \ddot{x}_{nom} \quad (4.25)$$

$$\ddot{y}_{cmd} = K(y_{des} - y_{est}) + B(\dot{y}_{des} - \dot{y}_{est}) + \ddot{y}_{nom} \quad (4.26)$$

$$\ddot{z}_{cmd} = K(z_{des} - z_{est}) + B(\dot{z}_{des} - \dot{z}_{est}) + \ddot{z}_{nom} \quad (4.27)$$

where all terms have been transformed from their original level-deck frame to the body frame of the vehicle.

### 4.3 Attitude Commander

The commanded accelerations from the previous section must now be achieved. The vehicle used in this thesis can be thought of as a disk with an acceleration vector. Thus the relation between the acceleration of the vehicle and the attitude of the vehicle is

$$\begin{bmatrix} m\ddot{x}_{cmd} \\ m\ddot{y}_{cmd} \\ mg - m\ddot{z}_{cmd} \end{bmatrix} = \begin{bmatrix} -T \sin \theta \cos \phi \\ T \sin \phi \\ T \cos \theta \cos \phi \end{bmatrix} \quad (4.28)$$

Assuming small angles, which is not unreasonable ( the vehicle in testing was limited to attitudes that deviated less than  $15deg$  away from level), the relation between acceleration and attitude simplifies to

$$T_{cmd} = m(g - \ddot{z}_{cmd}) \quad (4.29)$$

$$\phi_{cmd} = m \frac{\ddot{y}_{cmd}}{T_{cmd}} \quad (4.30)$$

$$\theta_{cmd} = -m \frac{\ddot{x}_{cmd}}{T_{cmd}} \quad (4.31)$$

Therefore, given a set of desired accelerations from the trajectory following controller, a set of desired attitudes can be generated. The heading of the vehicle, which did not appear in any of the previous equations, was always commanded toward the deck. This would ensure that the features on the deck would be observable. Once a desired attitude was calculated, it the was sent to the autopilot for tracking.

#### 4.3.1 Throttle Control

A throttle controller was designed to track the desired vertical acceleration produced by the trajectory following controller. The relation between throttle and vertical acceleration (assuming level) was:

$$\ddot{z} = A(T + T_0) - g \quad (4.32)$$

where  $T_0$  is hover throttle,  $g$  is the acceleration of gravity,  $A$  is the relation between vertical acceleration and throttle around hover, and  $T$  is the commanded deviation from hover throttle.

The gain  $A$  is a function of wind, ground effect, body rate, battery voltage, and other aerodynamic effects.

A PI controller was designed to track the commanded acceleration generated by the trajectory following controller. The controller took the form

$$T = k_p (\ddot{z}_{cmd} - \ddot{z}_{meas}) + k_i \int (\ddot{z}_{cmd} - \ddot{z}_{meas}) dt \quad (4.33)$$

where  $k_p$  is the proportional gain and  $k_i$  is the integral gain. In order to take into account ground effect, the requisite amount of throttle was recorded at 1m and at ground level. The hover throttle used between these two distances was found using a linear interpolation. For heights greater than 1 meter, the hover throttle of 1 meter was used.

# Chapter 5 |

## Hardware and Estimator Results

In order to validate the strategies presented in the previous two chapters, a small semi-autonomous hexacopter was outfitted with an additional computer. This vehicle used the strategies presented in the previous two chapters to land on a static platform as well as a moving platform. In this chapter, the challenges that arose while constructing the vehicle will be discussed including the calibration of the camera carried by the vehicle and converting the theoretical algorithms to code. Additionally, this chapter will discuss the performance of the estimator with some attention spent on assumptions made in earlier chapters and their validity.

### 5.1 Hardware

The vehicle used to test the system was a Y6 hexacopter bought from 3d Robotics, shown in figure 5.1. The camera used was a Bluefox Machine Vision camera with a wide field of view lens. The camera was mounted off the front of the vehicle so the deck was within view during the whole duration of the flights. The vehicle carried two computers: an Odroid XU4 computer and Pixhawk. The Odroid is a small, single board computer that uses a Samsung chip designed primarily for use in cell phones. The Odroid was responsible for all vision processing, estimation, trajectory planning and following. The Pixhawk is a small autopilot which was responsible for converting commanded attitude and throttle percentage into commands sent to the motors. In figure 5.1 the Odroid XU4 is the grey translucent box sitting on the top of the vehicle while the Pixhawk is out of sight on the level below.

The following few sub-sections will discuss the challenges inherent to this kind of hardware and how they were addressed in this project.

#### 5.1.1 Camera Calibration

The camera and lens used in this project produced images with a significant amount of distortion. This distortion must be accounted for in order to convert the pixel locations of features into bearing measurements. An example image from the BlueFox camera with a wide angle lens is shown in figure 5.2, with lens distortion clearly visible.



Figure 5.1: The Y6 hexacopter carrying the Odroid and Machine Vision camera used to test the system presented in this thesis.

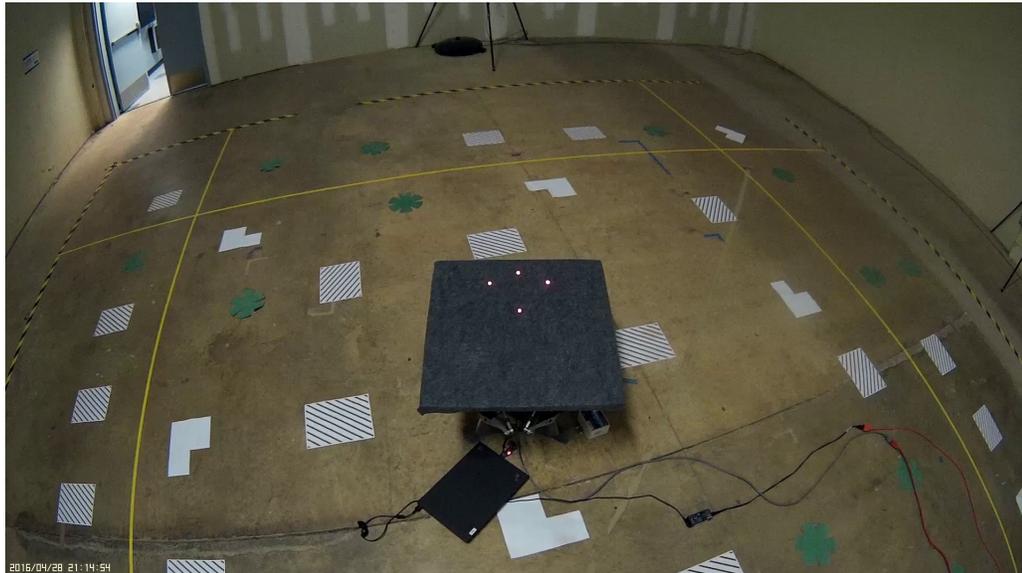


Figure 5.2: An unmodified picture from a camera mounted on the vehicle. The distortion in these images must be accounted for in order to convert pixel locations to bearing measurements.

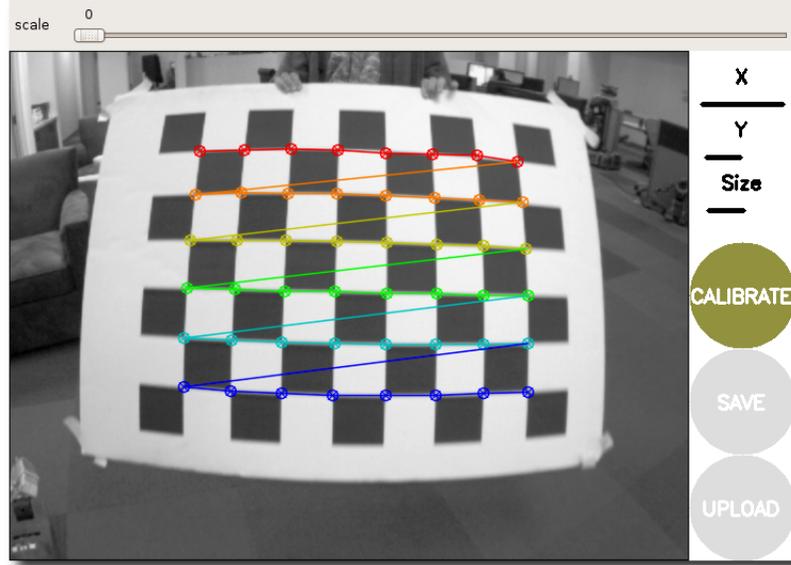


Figure 5.3: The process of calibrating the camera using a checkerboard with known dimensions and the ROS package *camera\_calibration*.

The distortion of the camera was measured using the ROS package *camera\_calibration*. This process records images of a checkerboard, seen in use in figure 5.3, from a range of relative positions. With knowledge of the size and shape of the checkerboard the parameters that define lens distortion (the camera's intrinsic parameters) can be calculated from a series of images.

The model used to represent the distortion present in the camera was

$$x' = x/z \quad (5.1)$$

$$y' = y/z \quad (5.2)$$

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \quad (5.3)$$

$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_2 x' y' + p_1 (r^2 + 2y'^2) \quad (5.4)$$

$$r = x'^2 + y'^2 \quad (5.5)$$

where  $x$ ,  $y$ , and  $z$  are the position of the feature in the camera frame, parameters  $k_1$  through  $k_6$  are radial distortion parameters, and parameters  $p_1$  and  $p_2$  are tangential distortion parameters. The values  $x'$  and  $y'$  are the nominal position of a feature on the undistorted image. These are the inputs to the distortion model where the output is the location of the feature on the distorted image. The radial coefficients found via camera calibration were

$$[k_1, k_2, k_3, k_4, k_5, k_6] = [-.188, -.1631, .1631, .131, -.366, .2436] \quad (5.6)$$

and the radial coefficients were

$$[p_1, p_2] = [.00136, .001605] \quad (5.7)$$

### 5.1.2 Camera Parameter Estimation

The pose of the camera in the vehicle frame, which was assumed known in Chapter 3, is needed for accurate estimation of the deck. With inaccurate camera pose information, there is a risk of light rejection during fast vehicle motions due to large differences between expected measurements and real measurements. To estimate and verify both camera pose in the vehicle frame and the camera calibration, the measurements from the camera were compared to the measurements expected given the true relative pose between the vehicle and deck. In order to fine tune the position of the camera in the vehicle frame, expected bearing measurements were compared to measured. This was done by calculating the position of each light in the camera frame, using information from the Vicon tracking system then projecting that onto a live video of the set of lights. Because of the high resolution of the camera and the accuracy of the Vicon system, the relative pose camera in the vehicle frame was tuned with great confidence. Additionally, this process validated the camera calibration process, as Vicon derived bearings were passed through the distortion model and drawn on the distorted image. The distance between drawn lights and measured lights remained small as the distance between the lights and the center of the image increased. The states tuned using this method were

$$\hat{x} = [x \ y \ z \ \phi \ \theta \ \psi] \quad (5.8)$$

where this state defines the translation and rotation to the camera from the vehicle frame. It was found the pose of the camera in the vehicle frame is

$$\hat{x} = [10.1cm \ .8cm \ 110cm \ 0.0rad \ -.76rad \ -.06rad] \quad (5.9)$$

### 5.1.3 Stewart Platform

Motion representative of a ship flight deck was created using a Stewart platform. The Stewart platform is capable of six degree of freedom motion(surge, sway, heave, roll, pitch, yaw).

Deck features are defined using a set of four light emitting diodes(LEDs), so that feature tracking could be done by first performing an intensity segmentation on an image, identifying "blobs" in the segmented image and then tracking centroids of each blob. LED locations in the deck frame were

$$\mathbf{b} = \begin{bmatrix} -12.8cm & 0cm & 0cm & 5.8cm \\ 0cm & -10.2cm & 10.2cm & 0cm \\ 0cm & 0cm & 0cm & 0cm \end{bmatrix} \quad (5.10)$$

A set of representative images of the dfeck are shown in figure 5.4. Note that even though there is significant difference in the attitude between pictures of the deck, the positions of the LEDs in the picture does not change significantly.

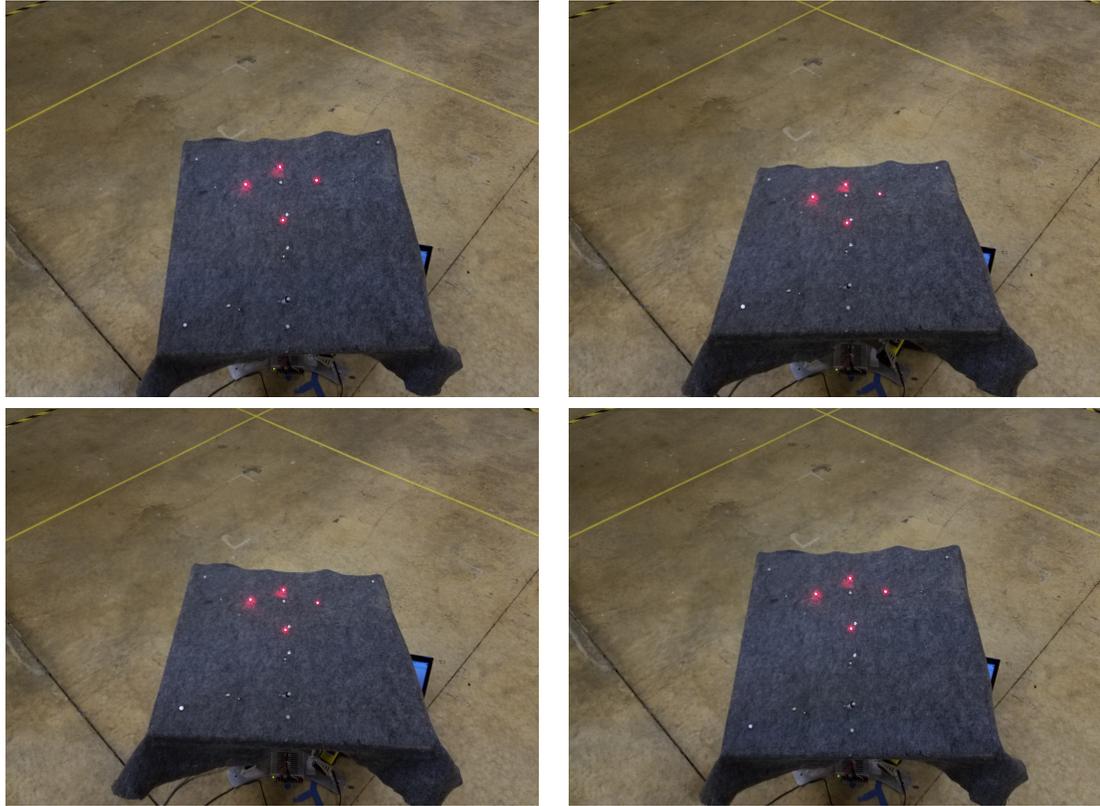


Figure 5.4: Multiple views of the landing platform from a typical position from which the vehicle would start a trajectory.

A laptop computer running a Matlab user interface is responsible for actuating the Stewart platform. Each of the deck's degrees of freedom can be excited independently (although actuation of one degree of freedom may affect the limits of actuation of another degree of freedom).

To mimic ship motion, each degree of freedom is driven with sinusoidal motion so that

$$d_{(\cdot)} = \sum_{n=1}^N a_{n,\cdot} \sin((\omega_{n,\cdot} t + \varphi_{n,\cdot})) \quad (5.11)$$

where  $d_{(\cdot)}$  denotes displacement in a degree of freedom.

The Stewart platform used here can use up to three sinusoids in each degree of freedom. Figure 5.5 show a sample of representative deck motion measured using the Vicon motion capture system-Vicon data is used as truth in assessing accuracy of the relative deck state estimator.

#### 5.1.4 Vicon Motion Capture System

The vehicle was flown in a room outfitted with twelve Vicon motion capture cameras. The cameras captured infrared images of the vehicle and Stewart platform. The vehicle and the

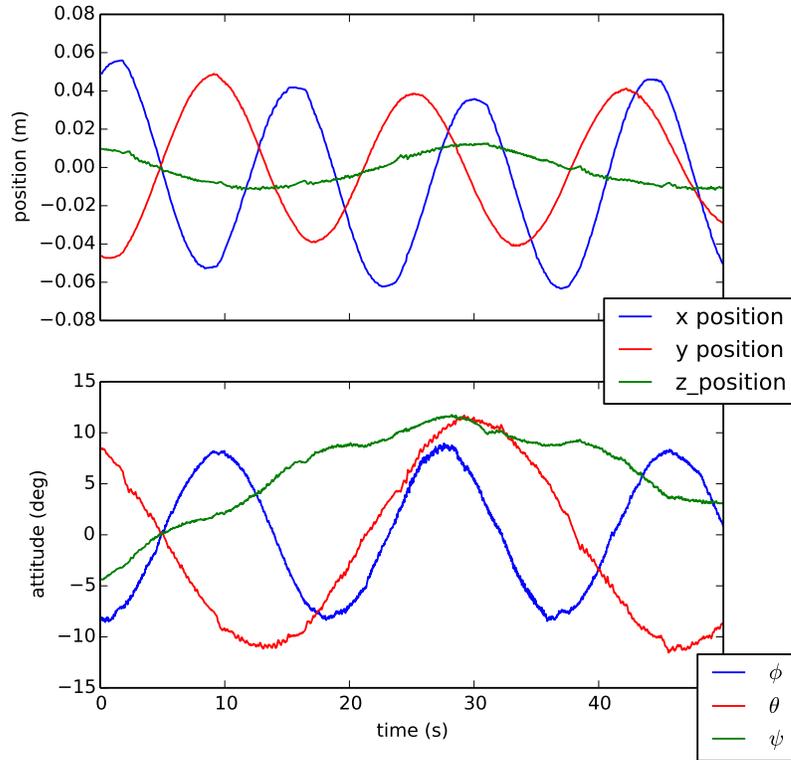


Figure 5.5: A representative sample of the motion of the deck. Each axis followed sinusoidal motion.

Stewart platform were carrying infrared reflectors which can be seen in figures 5.1 and 5.4. Vicon software was running on a ground computer which resolved the position and attitude of both the vehicle and Stewart platform at 100Hz. These state measurements were passed to the ROS ground station computer. The Vicon messages were then recorded using ROS logging software. Logging the Vicon measurements, which was treated as truth, allowed for measuring the performance of the UKF. In the following sections, all error plots are based off the measurements supplied by the Vicon system.

## 5.2 Software Implementation

The estimator, trajectory generator and control strategy presented in earlier chapters were implemented in software on an Odroid XU4 computer which was carried on the vehicle. These applications were written in C++ and Python and communicated using Robotics Operating System, or ROS, which is publish-subscribe based middleware.

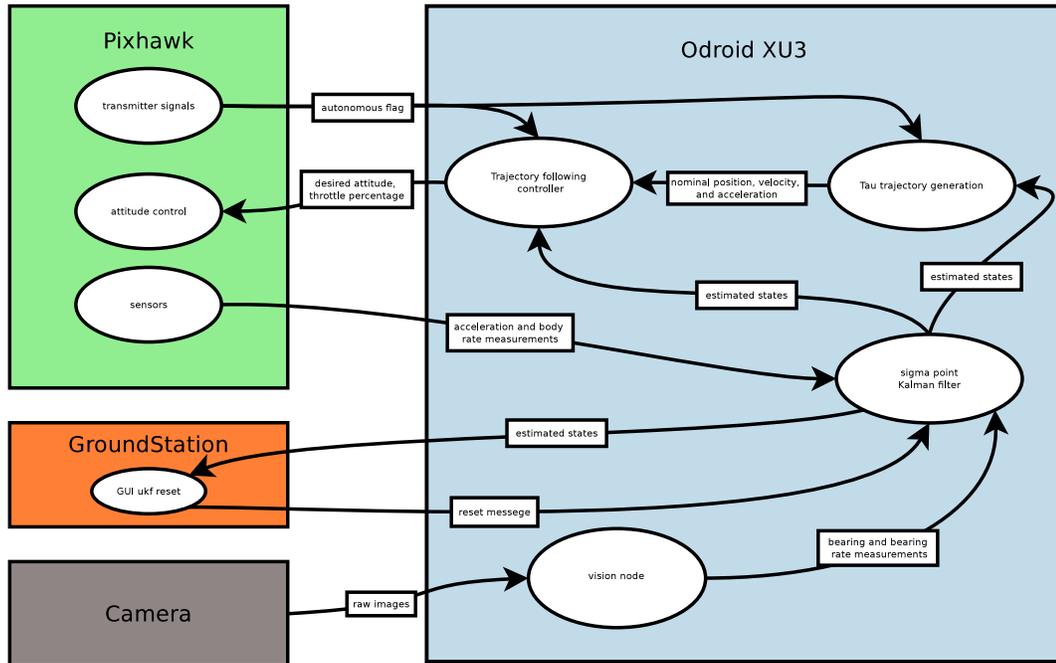


Figure 5.6: How information is passed around the system.

### 5.2.1 Information Flow

All computation needed to fly the vehicle autonomously was flown on board the vehicle. The Odroid XU4 computer was responsible for all computation relating to vision processing and deck state estimation. The Odroid was also responsible for all trajectory planning as well as the high level control; desired attitudes and throttle percentages were passed to the Pixhawk. The Pixhawk was responsible for tracking desired attitudes from the Odroid and estimating the roll and pitch of the vehicle relative to the inertial frame. The ground station computer was only used to monitor the estimator and desired trajectories to ensure that the Odroid was never given control while there was a faulty estimate or an issue with the system health.

The processes on the Odroid were broken into four major systems: estimation, trajectory generation, trajectory following, and vision processing. The vision processing node extracted bearing and bearing rate measurements to features and passed these measurements to the estimation node. The estimation node used these measurements as well as messages from the Pixhawk containing IMU measurements for state estimation. The trajectory generation used information from the estimation node to generate trajectories. The control node combined information from the trajectory generation node and the estimation node. Figure 5.6 depicts the manner in which information migrates through the system.

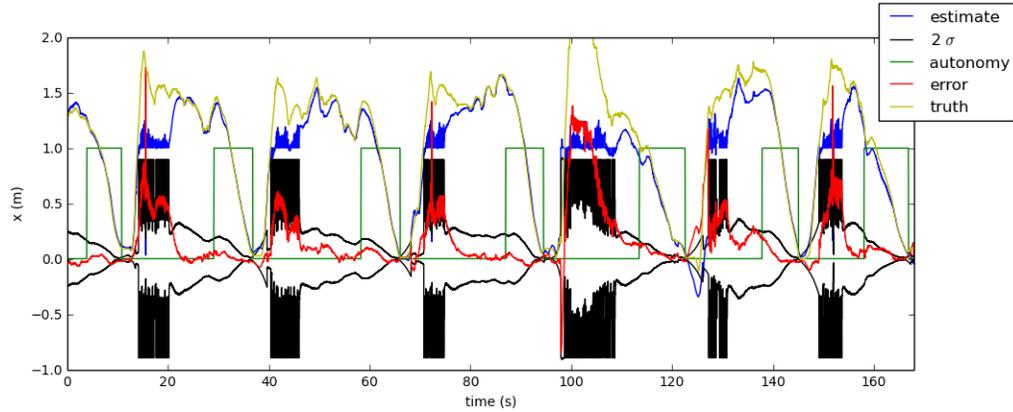


Figure 5.7: This figure shows the performance of the x position estimate of the deck in the vehicle frame. Regions where autonomy is one indicates the vehicle is flying autonomously. In some regions where the autonomy flag is low, the estimator is being reset multiple times a second which causes jaggedness of the estimate, error, and confidence interval.

## 5.3 Estimator Results

The estimator was tested on the vehicle using two types of approaches: flights to a static deck, where the deck was stationary and level, and flights to a moving deck during which all six degrees of deck freedom were excited. During this section, the performance of the estimator during both sets of tests will be presented and discussed following a brief discussion of the robustness and repeatability of the estimator.

### 5.3.1 Repeatability and Flight Tests

The structure of the system presented in the thesis is robust to large errors in the initial estimated state of the deck. It was found that the system still functioned as biases developed in the IMU. This robustness allowed for rapid testing of the vehicle with a minimal amount of time spent between autonomous flights tuning or calibrating sensors. This can be clearly seen in figure 5.7 where the processes of starting the estimator, landing, and resetting to starting position happens seven times over the course of three minutes. This plot is representative of all flights to the static deck; this data was not cropped to occlude an unflattering flight. The batteries used to power the motors of the vehicle are only flown for 5 minutes at most. Past five minutes of flight the full commanded throttle in manual mode is not capable to accelerating the vehicle to a satisfactory degree if something were to go wrong (e.g. the Odroid crashed). The data shown in figure 5.7 is the estimated x position of the deck in the vehicle frame during seven autonomous flights. The green line, when at a value of one, denotes fully autonomous flight. The blue line is the estimated position while the yellow and red lines are the true relative position and estimate error respectively. The two black lines represent the confidence of the estimate, they are plus and minus two standard deviations about zero. In the region following autonomous flights, where

the confidence interval and estimate are discontinuous, the vehicle is being manually flown back to near the starting location. During this period the estimator is constantly being reset which causes the discontinuities in the estimate, error, and confidence lines. The estimator was allowed to start once the vehicle reached a reasonable position above the deck. After this, the person in charge of the ground station would verify that the estimates had converged to a state close to the true state of the system. Note that the system converges before each flight and there is never a convergence to a state that is not close to the true state of the system. After the estimate had converged to a state that the individual manning the ground states deemed reasonable, a new autonomous flight would begin. During all seven flights the error of the estimator is within the two sigma bounds and is close to zero and approaches zero as the vehicle gets close to the deck.

### 5.3.2 Filter Initialization

Before the flight to the static deck is able to start, the filter must have an initial estimated state and covariance. During flights to the static deck, the position of the deck was assumed to be

$$\hat{\mathbf{x}} = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (5.12)$$

or one meter below and in front of the vehicle with all velocities and rates assumed to be zero. This initial state estimate represents the nominal starting position of the vehicle for a flight to the static deck. This nominal case starts with the vehicle one meter above and behind the deck with no difference between the attitudes or velocities between the vehicle and deck. The initial covariance associated with this estimate was necessarily large compared to the covariance of the converged state due to the large uncertainty in the initial conditions. The process of estimating the deck state started while the vehicle was being manually flown above the deck. In the case where the pilot is attempting to handover the vehicle while the vehicle is at the nominal starting position, it is unlikely the pilot would be able to relinquish control with the vehicle reliably within 10cm of the starting position. Additionally, the many flights were intentionally conducted with initial conditions that differed greatly from the nominal case. This was done to explore how the estimator handled initial states that differed greatly from the initial estimate and how the system performed at large distances from the deck. During flights to the static deck the initial estimated covariance for each state was

$$\mathbf{P}_{\mathbf{0},(\mathbf{xyz})} = (.5m)^2 \mathbf{I} \quad (5.13)$$

$$\mathbf{P}_{\mathbf{0},(\phi\theta\psi)} = (20deg)^2 \mathbf{I} \quad (5.14)$$

$$\mathbf{P}_{\mathbf{0},(\dot{\mathbf{x}}\dot{\mathbf{y}}\dot{\mathbf{z}})} = \left( .05 \frac{m}{sec} \right)^2 \mathbf{I} \quad (5.15)$$

$$\mathbf{P}_{\mathbf{0},(\dot{\phi}\dot{\theta}\dot{\psi})} = \left( 2 \frac{deg}{sec} \right)^2 \mathbf{I} \quad (5.16)$$

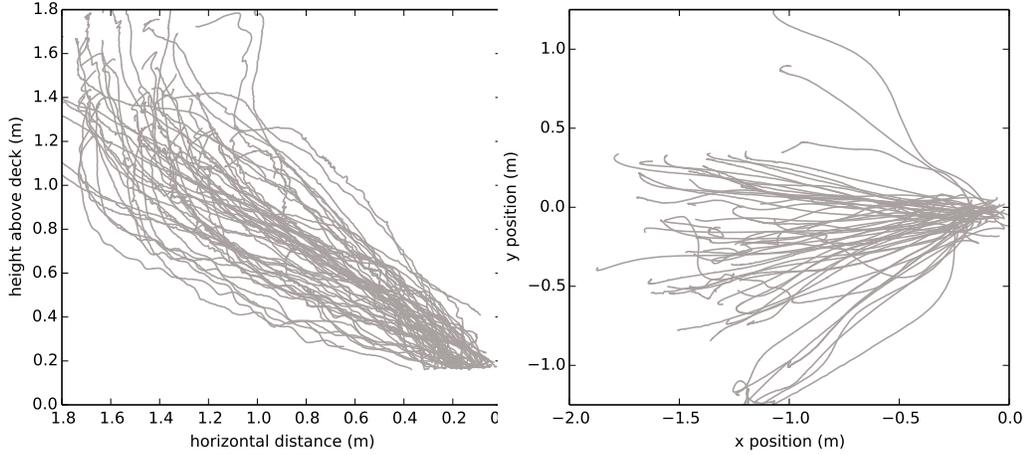


Figure 5.8: Flights flown by the vehicle to the static deck. Trajectories were generated using three tau guidance gaps with a duration of 15 seconds with a  $k$  value of .3.

and the total initial covariance was

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{P}_{0,(xyz)} & 0 & 0 & 0 \\ 0 & \mathbf{P}_{0,(\phi\theta\psi)} & 0 & 0 \\ 0 & 0 & \mathbf{P}_{0,(\dot{x}\dot{y}\dot{z})} & 0 \\ 0 & 0 & 0 & \mathbf{P}_{0,(\dot{\phi}\dot{\theta}\dot{\psi})} \end{bmatrix} \quad (5.17)$$

where all uncertainties were assumed to be uncorrelated.

### 5.3.3 Flights to a Static Deck

64 flights were flown to a static deck using tau guidance trajectories. Each trajectory consisted of three Tau guidance gaps, one for each axis of the deck. For each trajectory, a flight duration of 15 seconds was used along with a  $k$  value of .3 for each axis of the trajectory. This resulted in straight trajectories to the static deck with no commanded acceleration at touchdown. The purpose of these flights was to verify the system's ability to following trajectories based on estimated data. The trajectories were flown from varying initial conditions above the deck, with starting distances often exceeding 2 meters and initial relative headings often off by  $45^\circ$ . The paths flown by the vehicle to the deck has been broken into two plots: the height of the vehicle shown against the horizontal distance to the deck and the position of the vehicle in the  $xy$  plane of the deck, both can be seen in figure 5.8.

Both plots show a variety of paths flown to the deck with variations in relative position and velocity. The jagged nature of some of the trajectories flown was likely caused by energetic initial conditions at the moment when the Odroid was given authority over the vehicle. Prior to this, the vehicle was being flown manually. The pilot is largely focused on keeping the vehicle near

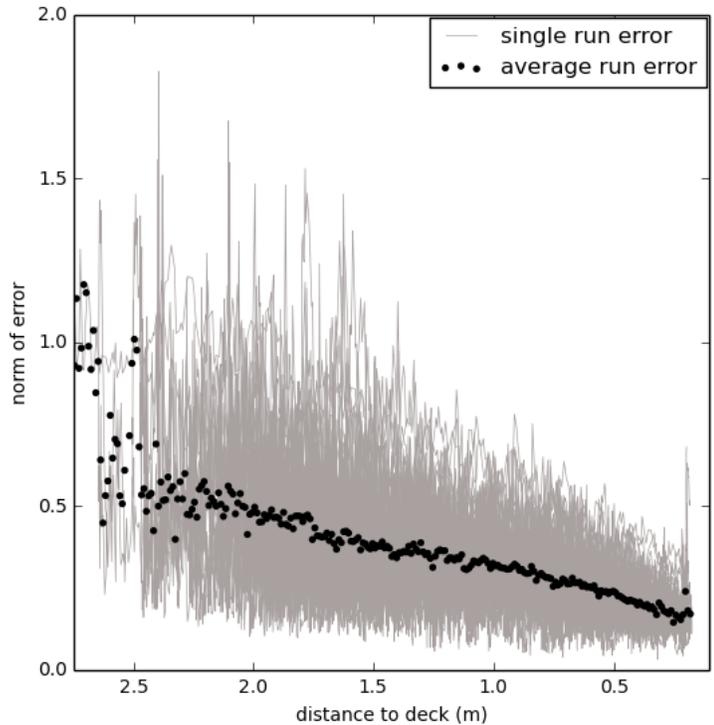


Figure 5.9: The norm of the estimate error as the vehicle approaches the deck. As expected, estimates get better as the vehicle approaches because bearing measurements contain more information relative to the noise. The error norms for individual flights are shown in grey while the average norm in 1cm bins is shown in black.

the nominal starting position but must also be aware of the person running the ground station computer. This often results in initial conditions where the vehicle is not level and is accelerating relative to the deck. At the end of many flights seen in figure 5.8, the vehicle never reaches (0,0). This is to be expected as the vehicle body frame is 17cm above the landing gear of the vehicle. Additionally, the trajectory generated does not intercept the deck perpendicularly, the vehicle would nominally close the gap of each axis at the same point and closes the gaps at the same rate. Therefore, the trajectory is a straight line to the goal. This is not meant to be used in the more demanding scenario of the dynamic deck but was sufficient for flights to a static deck where the purpose is to verify the estimator. This is why the distance between the vehicle and the goal rarely closes to precisely zero.

**5.3.4 Static Deck Results**

The hex was flown for over sixty flights to verify the results from the estimator. This was possible because of the small amount of overhead required for each flight after the initial calibration of the

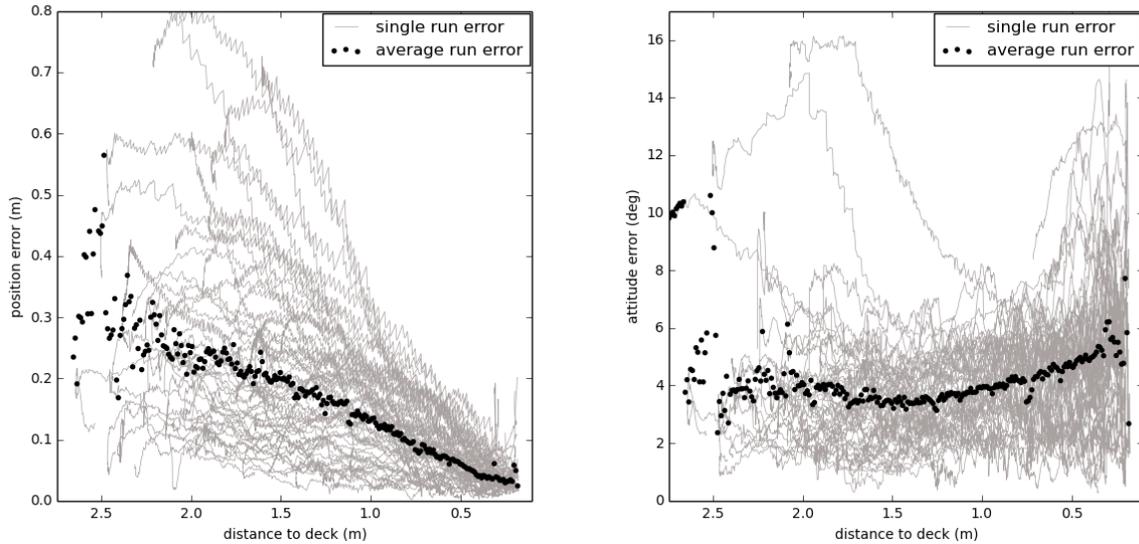


Figure 5.10: The error in the position estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black.

vehicle had taken place. The main takeaway from the flights can be seen in figure 5.9. This figure shows the norm of all states estimated using units of meters, radians, meters per second, and radians per second. This plot shows that the accuracy of the estimator improves as the vehicle approaches the deck. This is not surprising as the pose of the deck becomes more observable as the distance to the deck decreases. It should also be noted that the width of the lights on the deck are driving that observability: a distance of two meters away is effectively ten deck widths away from the deck. The large spikes in error seen at distances greater than 1 meter from the deck are errors in the estimated velocity, this is likely caused by biases in the IMU. These biases have greater corrupting effects at larger distances from the deck due to the decreased observability of the deck at larger distances. The sharp rise in error at the point of touchdown was driven by errors in the attitude estimates and will be discussed later on.

The left plot in Figure 5.10 shows the position error over the course of the 64 runs. This follows the same trend of converging towards truth as the vehicle approaches the deck. This error was likely caused by biases in the IMU. These would manifest in the estimate by pulling the estimate toward the bias. As the vehicle approached the deck and deck observability increased, camera measurements would have more weight and pull the estimate back towards the truth. Single runs with large amounts of error have error profiles that are far more jagged than those that performed better through the run. This is likely because the UKF was fed conflicting information from the camera and the IMU. In each run the error decreases towards zero suggesting that this approach is robust to large amounts of sensor noise. In the final moments of some flights there is a spike in error. This is likely caused by lights leaving the field of view of the camera. At the end of the flight these measurements would be the most susceptible to errors in the camera calibration.

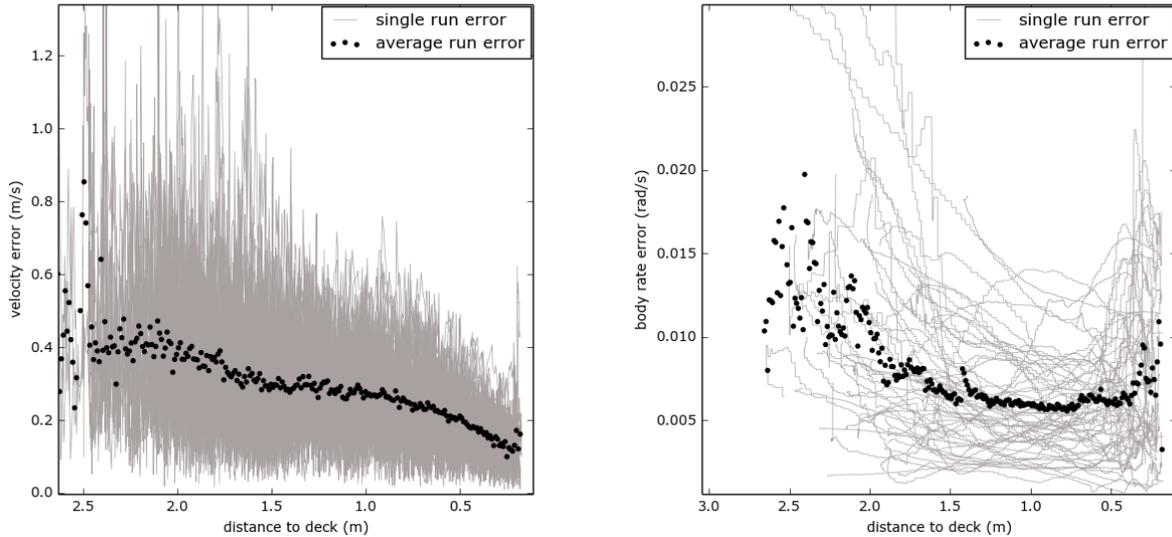


Figure 5.11: The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black.

The limited number of flights with this characteristic suggests that these flights lost more than one light. If all lights were lost due to occlusion or some other cause, the estimates would solely be driven by the IMU. At that point, any biases in the IMU would drive the error up without any correcting measurements.

the left plot in figure 5.10 shows the attitude error over the course of 64 runs. This plot is unique in that the average error reaches a minimum well before the end of the runs. This rise in error is likely driven by errors in the camera calibration. Errors in roll and pitch could increase as the vehicle approached the deck and lights strayed farther from the optical center where the calibration is most accurate. In the last few centimeters of flight there are several flights that have a spike in attitude error. This is likely caused by one or more features leaving the camera's field of view. This would feed less information to the system which would explain the increase in error.

The left plot in figure 5.11 shows the velocity error of the course of 64 runs. These estimates like those mentioned previously do better as the distance to the deck decreases. These estimates are more easily corrupted as the measurements from the IMU have a greater affect.

The right plot in figure 5.11 shows the estimated rate of the deck platform over the course of 64 rungs. These estimates are only calculated using the optical flow as the IMU is only proving information in the vehicle frame and can not provide information on the deck frame. The estimates perform well.

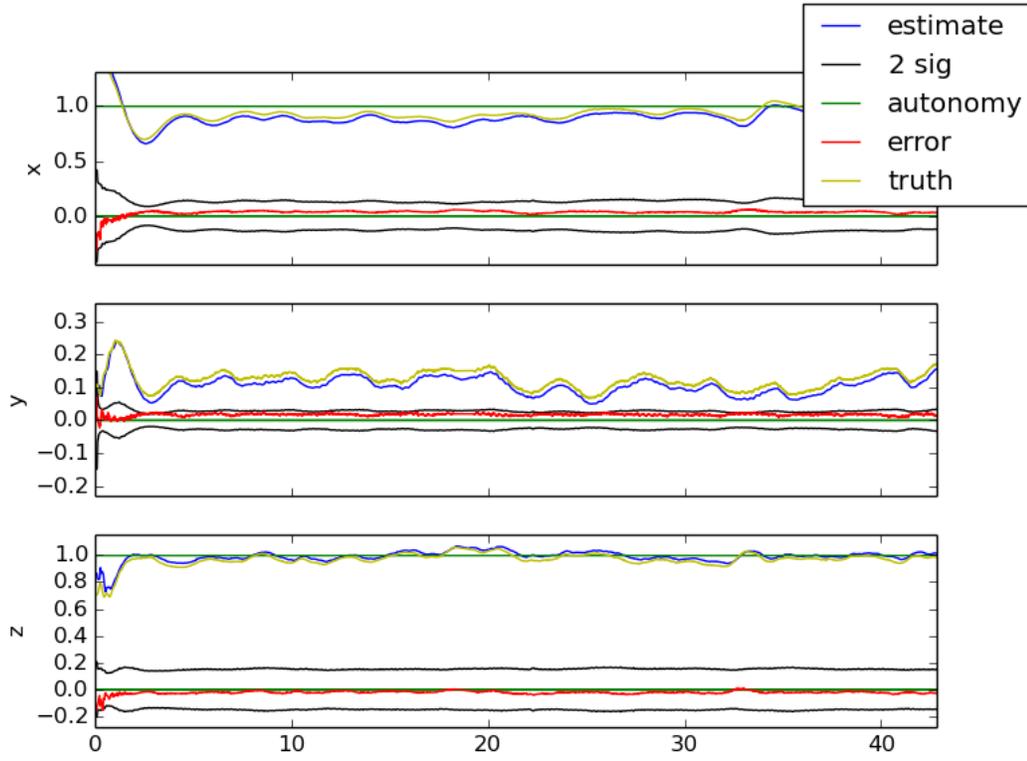


Figure 5.12: The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck.

### 5.3.5 Moving Deck Results

#### Hover Results

In order to validate the approach presented in this thesis, the vehicle was commanded to hover autonomously above the moving landing platform. The intent of this task was to investigate the steady state performance of the system while being run on a flying vehicle while tracking a moving platform. This test was run for over 40 seconds to gather a representative sample of the system while tracking the moving deck while in a hovering state. The behavior of the position estimates of the deck can be seen in figure 5.12. In figure 5.12 the estimated position of the deck in the vehicle frame is plotted in blue, the true position of the deck in the vehicle frame is plotted in yellow, the error between estimated position and true position is shown in red, and the confidence of the estimate is plotted as  $\pm 2\sigma$  about zero. During this commanded hover state, the total estimated error never exceeds 10cm with estimate errors never exceeding the two  $2\sigma$  bounds.

The estimated error never exceeding the  $2\sigma$  confidence intervals. The lack of confidence in the estimated state is the product of how the uncertainties in deck motion is represented.

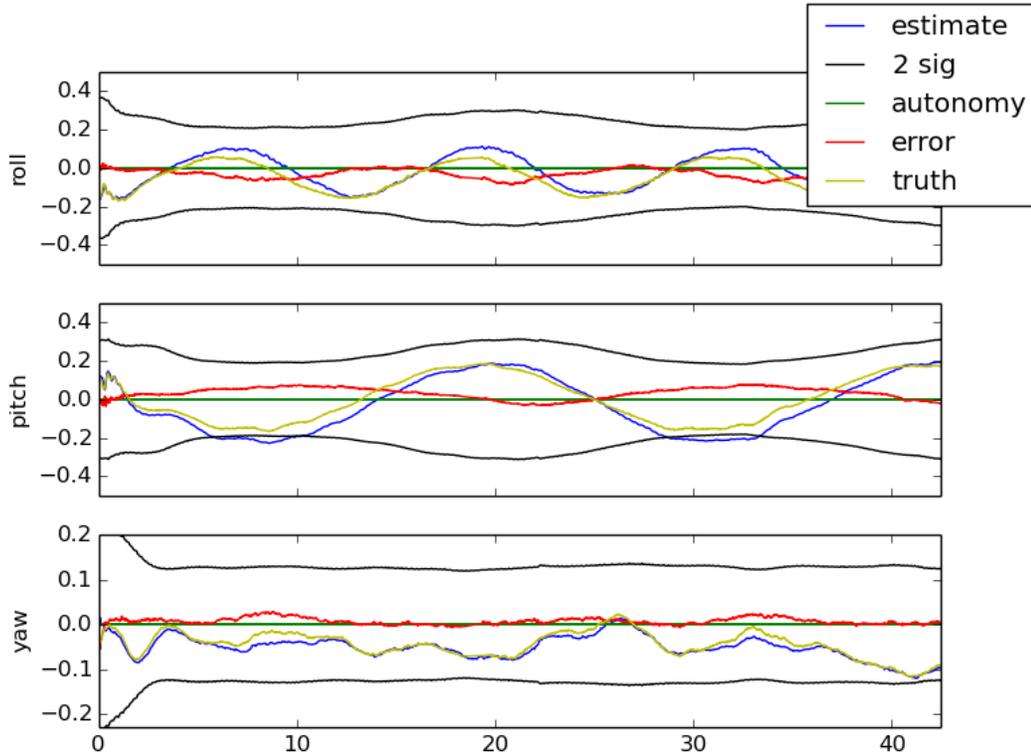


Figure 5.13: The estimated Euler angles (in radians) of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck.

As described in chapter 2, the ship is modeled as a second order system driven by external forces (i.e. waves and wind). Because the system is assumed to be constant velocity, the range of forces acting on deck must be captured by the process noise. The process noise, which is used to capture the inaccuracies in the model, is a function of the nominal amplitude and frequency of ship motion. The simplicity and ease of implementation of this model comes with a cost; it is not an accurate representation of ship motion. This is apparent in the estimates of deck position in figure 5.12. The second source of the inflated confidence interval is the process noise associated with IMU drift. As temperatures and other environmental variables change while the vehicle is flying, drift in the IMU was identified. A small component of process noise was added to the system to account for the changing nature of the acceleration measurements.

The behavior of the estimated attitude of the deck can be seen in figure 5.13 where all aspects of the data are presented in similar fashion to 5.12. In the same manner as the position estimates, the errors in attitude estimates never exceed the  $2\sigma$  confidence interval. This lack of confidence for this set of states is only caused by the assumptions made while constructing the component of process noise associate with ship motion. The state most accurately measured out of the

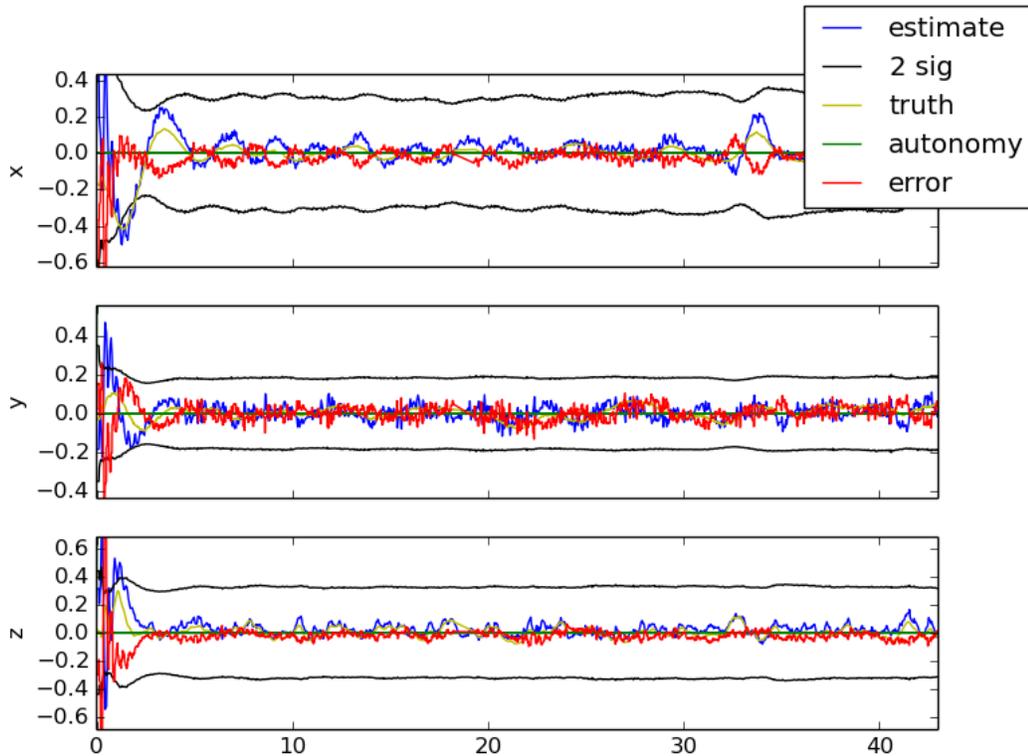


Figure 5.14: The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck.

three presented in 5.13 is yaw. This is the result of a yaw state that is far more observable than roll or pitch; that a small change in yaw results in the greatest change in the position of the lights on the optical plane of the camera compared to the two other states. An aspect unique to the yaw estimate is the constant nature of the  $2\sigma$  confidence interval. This is the result of the constant observability of the yaw state. Unlike the  $2\sigma$  bounds of yaw, the confidence interval of pitch varies significantly with pitch. This result is driven by the observability of the state. Consider the case where the landing platform is parallel to the optical plane of the camera and all features of the landing platform are contained within that plane (which is the case in this system). An incremental change in pitch or roll of the platform will result in a minuscule change in the projected position of the features on the optical plane of the camera. As the deck approaches that state with larger values of pitch, the observability of the state decreases and the confidence interval of the estimated state increases. Overall, estimates of attitude were accurate to within 5 degrees.

The behavior of the estimated velocity is shown in figure 5.14 where all aspects of the data are presented in the same manner as figure 5.12. Similar to both the position and attitude estimates, the errors from this period of hover never exceeds the two sigma bounds. This is due to the

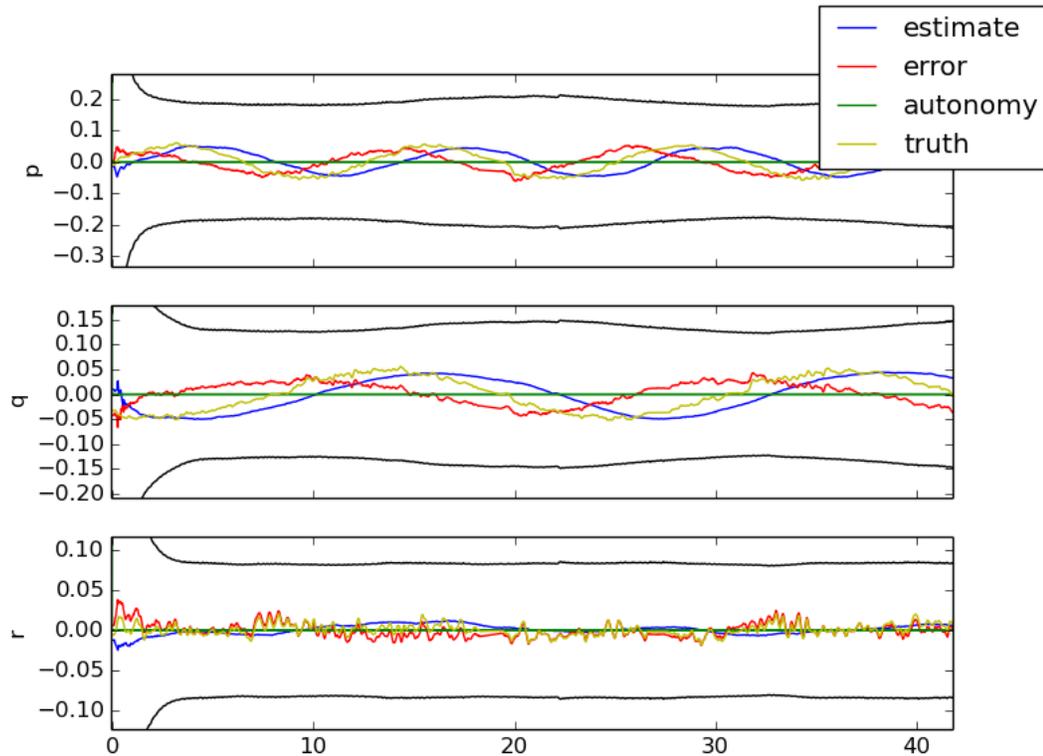


Figure 5.15: The estimated position of the deck in the vehicle frame. During this period, the vehicle was autonomously hovering over the moving deck.

large uncertainty from the noise model of the ship motion which was described in chapter 2. Much like the changes in observability which drove confidence intervals for attitude, there are some components of deck motion that are more observable than others. Velocities toward that camera, which would consist of velocities in the  $x$  and  $z$  axis of the vehicle, would be far less observable than motions that are perpendicular to the optical axis of the camera. This variable observability is the cause for the larger confidence bounds in the  $x$  and  $z$  axis which would contain the motion parallel to the optical axis of the camera. Velocity estimates were within five centimeters per second of the truth. Estimated states converged to real states in a few seconds.

Results from the estimated body rate of the deck can be seen in figure 5.15 where all data is presented in the same manner as figure 5.12. Body rate measurements about the  $x$  and  $y$  axes of the deck are poorly observable. This explains the lagging behavior seen in figure 5.15. Measurements of body rate about the  $x$  and  $y$  axes of the deck do not result in significant changes in the projected image plane. The error of the estimated deck rate never exceeds the  $\pm 2\sigma$  bounds. This is caused by the same factors which resulted in an identical behavior for attitude estimates. The general model of ship motion which was used to create the process noise results in greater uncertainties in estimates than a process model that was better informed of the dynamics of the

ship motion.

### Landing Results

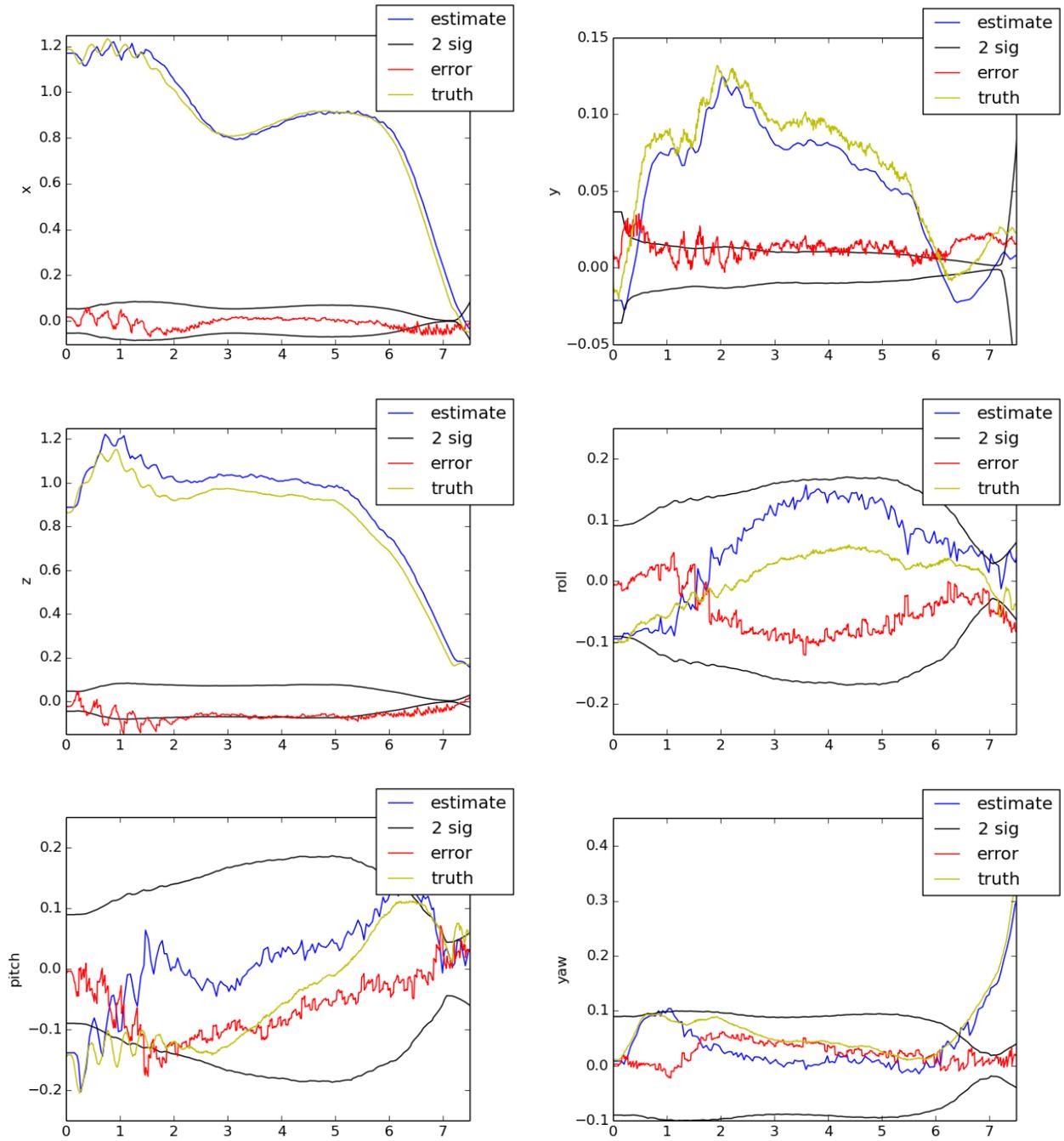


Figure 5.16: The estimated position and deck attitude during a flight to the moving deck.

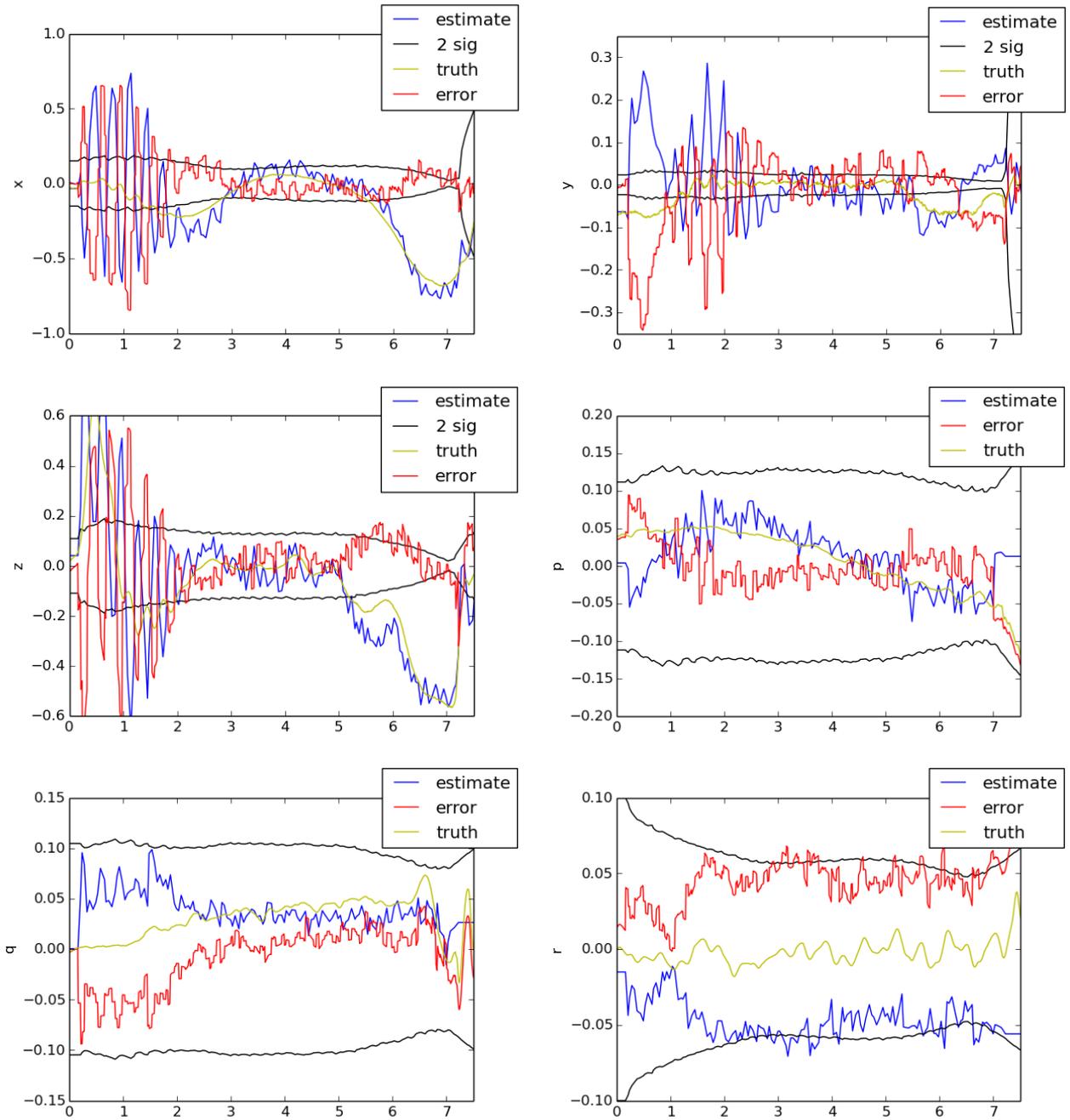


Figure 5.17: The estimated velocity and deck body rate during a flight to the moving deck.

Estimating the deck state during a flight to the moving deck is far more demanding of the system than holding a hover above the deck. Aspects relating to vehicle vibration, camera measurements, and trajectory tracking all play a role in the final behavior of the system. Compared to a hover, tracking the trajectory will entail far greater motion of the lights in the camera frame

if the touchdown angle  $\lambda$  is significantly greater than zero. Though the heading of the vehicle was always commanded towards the deck, all rolling and pitching of the vehicle would result in much greater bearing rate measurements. These dynamics were not present during the commanded hover and were largely absent from the landings on the static deck due to the long periods of flight. During the landings to the moving deck the resulting roll rate and pitch rate will have significant impact on the measured bearing rates of the lights.

The estimated position of the deck in the vehicle frame is presented in figure 5.16 along with estimated attitude. Over the course of the flight the position estimates behave well. Initial state estimate confidence is relatively high with error in the estate within a handful of centimeters for each component of position. As the vehicle begins the landing process, the estimates remain accurate and converge to zero error as the distance to the deck decreases. The attitude estimates behave somewhat poorly during the hover period of the flight with individual errors exceeding .1 radians. During this portion of the flight the vehicle is autonomously hovering above the deck with the deck pitched away from the vehicle. This relative attitude results in poorer observability. As the relative attitude changes the states become more observable and estimates become more accurate. This is somewhat expected as the observability of these states given the two dimensional nature of the light configuration. The estimated yaw angle is consistently more accurate as this state is clearly observable from the images from the camera. As the vehicle begins the process of landing on the deck, generally the angle estimates get better. As the vehicle enters the second half of the trajectory, there is some deterioration of the state estimates. This is due to some loss of accuracy in the camera calibration. As the lights stray from the center of the image, the accuracy of the light measurements diminishes. However, as the errors due to the inaccuracies in the camera model are on the order a degrees, the estimated relative yaw angles between the vehicle and deck remains accurate. The estimated velocity of the deck and deck body rate is present in figure 5.17. The estimated velocity near the beginning of the flight is accurate. Biases in the x and z components of the velocities are likely to be caused by some bias in the IMU. Any bias that is in line with the vector to the deck has an outsize effect on the estimate. However, the estimates do get better as the vehicle follows the trajectory to land on the deck. Errors are better because the system is generally more observable as the system gets closer. Deck body rate estimates, unlike all other states, get less accurate as the vehicle approaches the deck. This is caused by inaccuracies in the camera calibration model. When the lights are at great distances from the center of the image, there is likely to be greater error in the suspected location of the pixel in the undisturbed camera image. Taking the derivative of this causes significant error.

### **Aggregate Results**

In order to evaluate the range of the estimator behavior while landing on the deck, the vehicle was autonomously landed on the platform 20 times. The results of the estimator and the flight paths taken will be presented here. For each flight, the starting position of the vehicle was nominally one meter above and meter behind the deck. While flight paths to the deck are a function of deck attitude and deck rate for each flight, these flights are repeated with constant nominal positions which allows for repeatability.

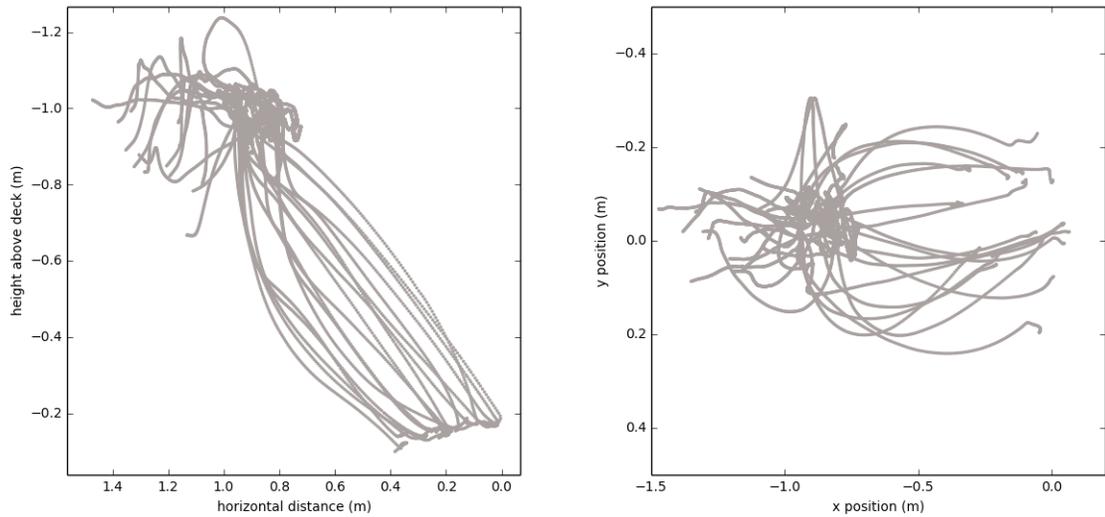


Figure 5.18: Flights flown by the vehicle to the moving deck. Trajectories were generated using the method described in Chapter 4.

During flights to the moving deck, unlike flights to the static deck, the system was handled by a single person. Before the onboard computer was given authority over the system, the pilot flew the vehicle to near the nominal starting position of the trajectories. Once the pilot was satisfied with the state of the vehicle, the Odroid computer was given authority over flight control. At that time, the estimator would be started using initial states from the Vicon tracking system in the flight space. The initial estimated covariance of the estimates was small, on the order of centimeters for position and several degrees for attitude. To allow the estimated states to converge and to ensure reasonable flight paths to the deck, the vehicle entered a hover state above the deck for a minimum of five seconds. The minimum hover period of five seconds allowed for any transients in the dynamics from the handover to die out and to ensure that trajectory following behavior would only be driven by the controllers on the vehicle. After the minimum hover period, the vehicle would start the deck landing process once the vehicle was within a certain distance of the nominal starting position and the projected touchdown trajectory was reasonable. The condition relating to relative position was to ensure that the vehicle would not attempt to land while there were still transients in the system left over from the handover from manual flight. This condition was rarely unmet due to the minimum five second commanded hover at the nominal starting position above the deck. The conditions relating to the projected deck attitude was to limit the trajectories which the vehicle would follow. The angles used in these condition,  $\lambda$  and  $\gamma$ , where chosen so the vehicle would not have to approach the deck from a heading more than 45 degrees different from the initial heading. The angle  $\gamma$  was limited to angles greater than 20 degrees from vertical so there would not be a step change in desired  $\lambda$  at touchdown. Once all conditions were met, the vehicle would track the generated trajectory to a landing on the platform.

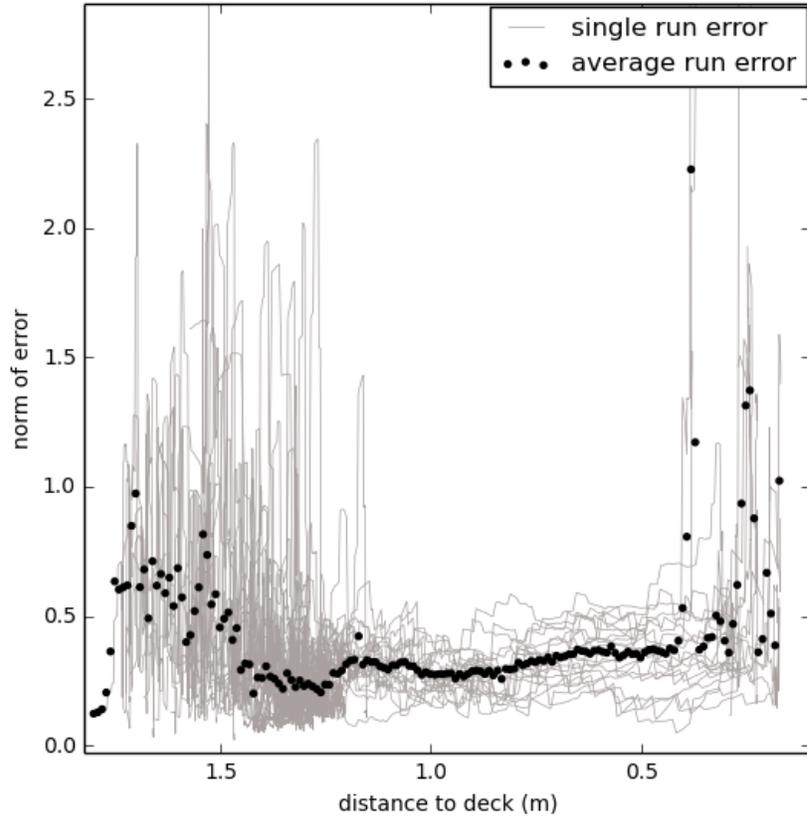


Figure 5.19: The norm of the error with individual errors shown in grey and the average error in 1cm bins shown in black for flights to a static deck.

The behavior of the position estimates can be seen in left plot in figure 5.20 where the norm of the position error as a function of distance to the deck is plotted in grey for individual flights and black for averages. This plot shows that estimates in position error generally get better as the vehicle approaches the deck. As the vehicle enters the portion of the flight, where the features of the platform are no longer contained within the center of the image, there is occasional loss in estimate accuracy. This loss of accuracy is likely driven by lights straying significantly from the center of the optical center of the camera or the complete loss of one or more lights. Throughout the flights, the norm of the error in estimated position generally stays below 10cm.

The behavior of the attitude estimates can be seen in the right plot in figure 5.20 where results from the flights are presented in the same manner as the norm in position error plot. Attitude estimates are unique among the estimated states because there is a clear distance from the deck at which the system is able to generate the most accurate estimates. At the beginning

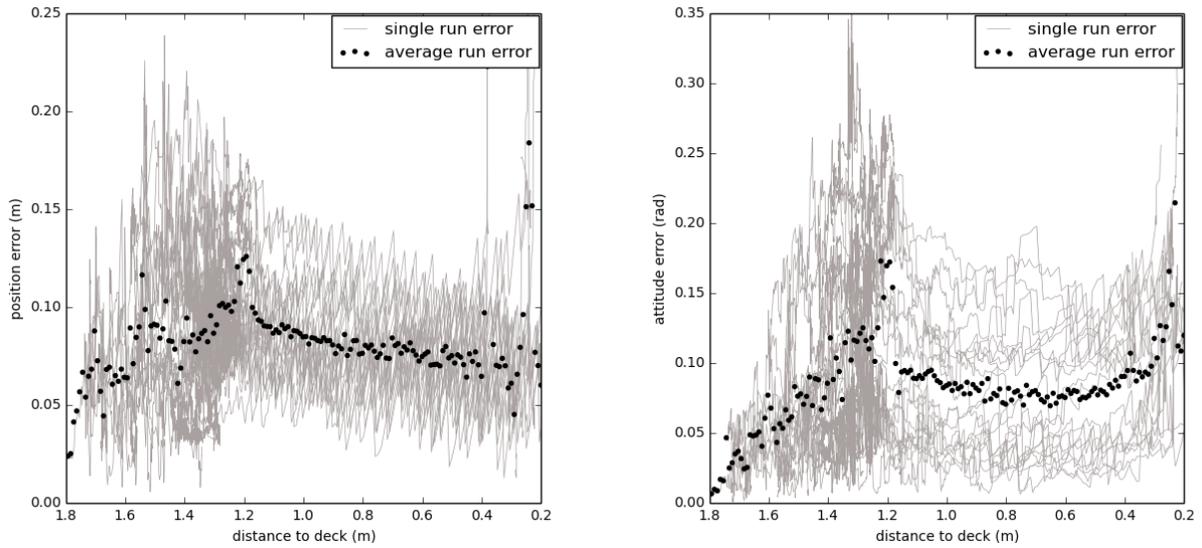


Figure 5.20: The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black.

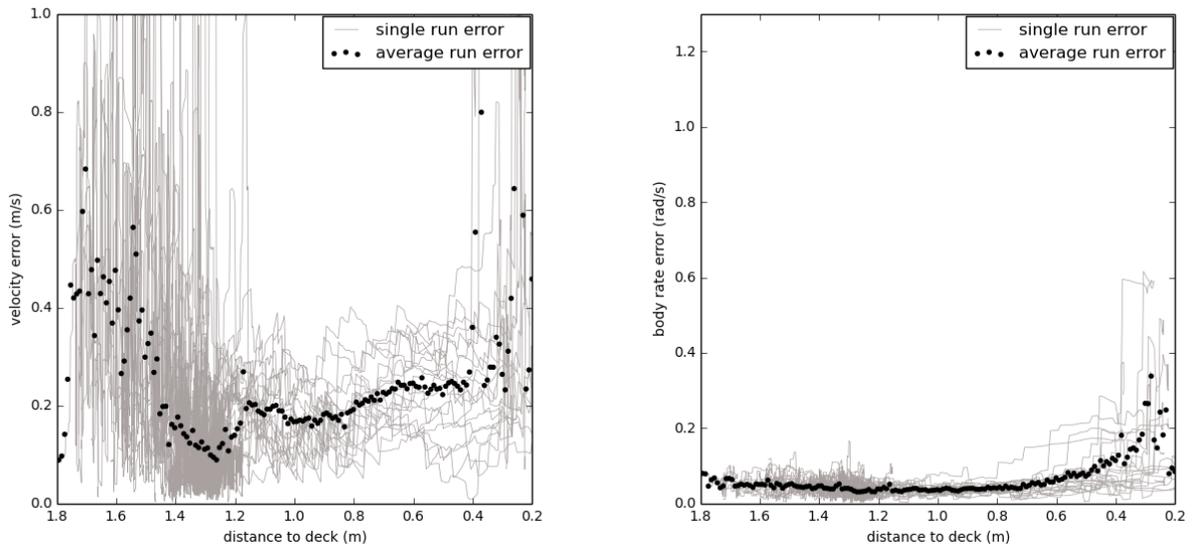


Figure 5.21: The error in the velocity estimates for 64 flights to a static deck. Individual errors are shown in grey while the average error in 1cm bins is shown in black.

of the flight, the average error in the attitude estimate was .1 radian, or about 5 degrees. This error decreases to .08 radians at half the distance to the deck. This decrease in error is almost certainly caused by better observability of the system while still measuring the features of the

platform at locations near the optical center of the camera. As the vehicle gets closer to the deck the bearing measurements of the features, which ostensibly would contain more accurate system measurements of the deck, become corrupted by constant errors due to calibration. The norm of the error in attitude estimates increases to .12 radians as the vehicle finishes the second half of the flight.

The behavior of the velocity measurements can be seen in figure 5.21. Velocity estimates are noisy while the vehicle is at distances greater than the starting distance. This is likely because there were still transients in the vehicle dynamics leftover from the transfer from autonomous flight. As the vehicle gets closer, errors in the system drops to  $.2m/s$ . While the vehicle is approaching the deck there is significant deviation in estimate accuracy with some approaches having velocity estimates less than  $.1m/s$  with other flights having norms of error greater than  $.3m/s$ . At the end of the flight there was a large increase estimate error. This is likely caused by errors in camera calibration and biases in the accelerometer.

The behavior of the deck body rate estimates can be seen in the right plot in figure 5.21 where the norm of the attitude error is plotted against distance to the deck. Deck body rate measurements follow a trend similar to the previous sets of states described; the estimate is far more accurate while the vehicle is hovering and approaching than when the vehicle is directly above the deck. Norms of error start at  $.05rad/s$  while the vehicle is hovering and grow to  $.3rad/s$  as the vehicle is landing on the platform.

### 5.3.6 Sources of Error

The error discussed in previous sections likely had two sources. The first and dominant source would be the error caused by IMU biases. Even with a perfectly calibrated camera, the a bias in the IMU would pull the estimates from the true state by a substantial amount when the vehicle is near the beginning of the flight path. The component of an acceleration bias in line with the vector to the deck would have the greatest impact on the integrity of the estimates. This is due the the difficulty of measuring distance to the deck using the set of measurements of the lights. Biases orthogonal to the vector between the vehicle and deck would have a near negligible effect as the direction to the deck is highly observable using camera measurements. Biases in the gyroscope would result in noisy estimates of velocity though the calibration of the gyroscope was found to be robust against the temperature changes found on the vehicle during the course of flying. Biases in the accelerometer were found to drift much more, upwards of  $.2m/s^2$  over the course of five minutes of flight. To counteract this, the system was left running for several minutes with all software running to allow the system to reach a steady state temperature before sensors were calibrated.

The second likely source of error in the system is the camera calibration. Errors in the camera calibration are assumed to become greater with distance from the optical center of the camera. This is largely because during there is more information of the parameters of the camera model taken in during the calibration process. Errors in the camera calibration model would cause grater errors in attitude and body rates than position and velocity as it is assumed the these errors would not significantly change the direction in which the deck lies which is the measurements

used to compute the relative position and velocity. Errors relative to camera calibration would decrease with distance to the deck assuming the vehicle is commanded to a heading that places the deck at the center of the image.

## **5.4 conclusion**

This chapter covered the results of the estimator presented in chapter 3 as well as the hardware the system was tested on. The system behaves well with little setup time between runs and keeps the error of the to a manageable level. Errors in the estimates were seen to decay as the vehicle approached the deck. These errors were caused by biases in the accelerometer. Individual runs to a static and moving deck were covered as well as composite runs where the total number of runs were aggregated. The system has proven to be a useful solution to the problem deck state estimation for autonomous shipboard landings.

# Chapter 6 |

## Conclusions

This thesis presented a method of estimating the relative state of ships for the autonomous landing of UAVs using a monocular vision system. The work presented here was motivated by several factors. Giving UAVs the ability to land autonomously on ships during high sea states would greatly expand their capabilities and responsibilities during shipboard operations. These drones becoming ubiquitous and are being looked to for many new commercial applications such as delivery and site inspection. The need for a relative state estimation system is great for these potential new fields. The loss of GPS is not an acceptable cause for failure if relative pose information is available from sensors such as cameras.

### 6.1 Summary of Contributions

- **Estimator Design:** An unscented Kalman filter was designed to estimate the relative state of a ship in rough seas. The system needs only a range of expected accelerations experienced by the ship and a set of known features in the ship frame. The assumption of zero mean acceleration makes the system attractive because it allows the system to be deployed to a ship-sea system with limited knowledge of the dynamics and system.
- **Tau Based Trajectories:** The use of tau based trajectories was explored as a means of generating closed form trajectories to a moving ship flight deck. Several methods of generating these trajectories were presented including tau-dot, tau guidance, and high-order tau-guidance. The use of high-order tau guidance was found to be the most practical of the three methods as there was no assumption that the vehicle could track trajectories with instantaneous changes in acceleration.
- **Hardware Implementation and Testing:** The system was deployed on a Y6 hexacopter which carried an Odroid XU4 computer and a Bluefox Machine Vision camera. These were used to autonomously land the vehicle on a static and moving platform using the estimator described in this thesis. Using Vicon data, it was found that errors in the estimates decreased with distance to the platform with touchdown errors on the order of a centimeter and a few degrees for position and attitudes estimates respectively.

## 6.2 Lessons Learned

- **IMU biases:** Throughout the flight testing process the accelerometer on the vehicle was very difficult to calibrate. Significant biases could develop while the vehicle was sitting on the ground or during flight. Thermally calibrating the IMU carried by the Pixhawk could have reduced the impact of IMU biases significantly.
- **Testing Process:** During the static flights to the deck a flight crew of two people was needed. This made testing much slower because the helper needed to be trained how to operate the ground station computer. During flights to the moving deck there was no additional helper. This proved to be far desirable. While it was easier to plan for single person flight tests, the real advantage was the reduction in time between flights. During flights to the moving deck, the transmitter passed commands to the Odroid computer resetting the internal state of the estimator and controller. This allowed for many flights without landing the vehicle.
- **Simulation:** A large amount of time was spent hunting errors in the code while flight testing. A significant amount of effort spent hunting for erroneous or missing transformations and sign errors could have been saved with minimal effort spent on a simulation.
- **System Parameters:** While examining ROS logged data, it was often hard to determine which parameters or gains were causing problematic behaviors. This is because early in the process, many gains were written as part of the code. This made it difficult to look at old data, after all parameters could have changed, and make sense of how the system was responding. Later in the project, gains and parameters were moved to the ROS ecosystem. This made it much easier to record parameters and much more likely that they would be recorded.

## 6.3 Future Work

This work has contributed to the estimation and tracking of a ship in GPS denied environments. There are many areas for future work in estimation, trajectory planning, and control.

### 6.3.1 Future Estimation Work

The assumption that there is a known set of features on the deck is reasonable, ship flight deck are painted according to regulation and can be known before the process of tracking the ship has started. However, the knowledge that will likely be required for full autonomous landing will probably not be found in this set. The knowledge of large prominent features will allow for tracking, with a great degree of accuracy, when the vehicle is hovering above the deck while all features are within view. When the vehicle attempts to land it would be prudent to use the information of previously unknown features while landing, especially if the observable set of original set of features is largely outside the field of view. There is room for work on identifying

robust features capable of being tracked on the superstructure of the ship that will be known to be observable within the field of view. There is also room to fuse measurements between LIDAR and vision. LIDAR, while consuming more power and adding significant cost to applications, would add a high degree of certainty to measurements. In this work it was assumed that the knowledge of the features on the ship flight deck were known with perfect certainty. If LIDAR was introduced, uncertainty in the relative positions between these features could be better handled. There is also room for tracking ship motion using stereo vision. Using a stereo system with asynchronous timings could double the measurement rate and using different offsets of the vehicle's body frame would likely improve the observability of the system. Additionally, there would be some redundancy in the system. There would be no need to redesign the estimator, during the measurement updates for each camera could call different relative poses.

### **6.3.2 Ship Motion**

Throughout this thesis it was assumed that an application built to land on a platform following sinusoidal motion would be applicable to real ship motion. While ship motion estimation would likely be unaffected by a more realistic model, because the estimator here only needs the range of accelerations seen and is not reliant on ship phase, the control and trajectory planning may be. Since ship design is always an interest, new and more accurate ship motions should be considered.

### **6.3.3 Motion Planning**

Tau based trajectory planning will likely have a future in trajectory planning for small drones. Investigating their suitability for docking where there a small force may be a requisite for a safe and secure landing. The wind effects that were a bi-product of the greater wind field interacting with the superstructure of the ship and other aerodynamic effects were largely outside the focus of this thesis. While the assumption that these effects may not greatly affect the assumption of first order tracking, they should be investigated further.

# Bibliography

- [1] FERRIER, B. and T. MANNING (1998) “Simulation and testing of the landing period designator (LPD) helicopter recovery aid,” *Naval engineers journal*, **110**(1), pp. 189–205.
- [2] PERVAN, B., F.-C. CHAN, D. GEBRE-EGZIABHER, S. PULLEN, P. ENGE, and G. COLBY (2003) “Performance Analysis of Carrier-Phase DGPS Navigation for Shipboard Landing of Aircraft,” *Navigation*, **50**(3), pp. 181–191.
- [3] GOLD, K. L. and A. K. BROWN (2004) “A hybrid integrity solution for precision landing and guidance,” in *Position Location and Navigation Symposium, 2004. PLANS 2004*, IEEE, pp. 165–174.
- [4] YAKIMENKO, O. A., I. I. KAMINER, W. J. LENTZ, and P. GHYZEL (2002) *Unmanned aircraft navigation for shipboard landing using infrared vision*, Tech. rep., DTIC Document.
- [5] OH, S.-R., K. PATHAK, S. K. AGRAWAL, H. R. POTA, and M. GARRATT (2006) “Approaches for a tether-guided landing of an autonomous helicopter,” *IEEE Transactions on Robotics*, **22**(3), pp. 536–544.
- [6] ARORA, S., S. JAIN, S. SCHERER, S. NUSKE, L. CHAMBERLAIN, and S. SINGH (2013) “Infrastructure-free shipdeck tracking for autonomous landing,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, pp. 323–330.
- [7] SARIPALLI, S., J. F. MONTGOMERY, and G. S. SUKHATME (2002) “Vision-based Autonomous Landing of an Unmanned Aerial Vehicle,” in *International Conference on Robotics and Automation*, IEEE, Washington DC, pp. 2799–2804.
- [8] PROCTOR, A. and E. N. JOHNSON (2005) “Vision-only Approach and Landing,” in *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2005-5871, American Institute of Aeronautics and Astronautics, San Francisco, California.
- [9] SARIPALLI, S. and G. S. SUKHATME (2007) “Landing a Helicopter on a Moving Target,” in *IEEE International Conference on Robotics and Automation*, IEEE, pp. 2030–2035.
- [10] WU, A. D., E. N. JOHNSON, and A. A. PROCTOR (2005) “Vision-aided inertial navigation for flight control,” *Journal of Aerospace Computing, Information, and Communication*, **2**(9), pp. 348–360.
- [11] WATANABE, Y., E. N. JOHNSON, and A. J. CALISE (2006) “Vision-based guidance design from sensor trajectory optimization,” in *AIAA Guidance, Navigation, and Control Conference, Keystone, Colorado (August 2006) AIAA-2006-6607*.
- [12] DIEHL, D. D., P. DEBITETTO, and S. TELLER (2005) “Epipolar constraints for vision-aided inertial navigation,” in *Application of Computer Vision, 2005. WACV/MOTIONS’05 Volume 1. Seventh IEEE Workshops on*, vol. 2, IEEE, pp. 221–228.

- [13] JOHNSON, E. N., A. J. CALISE, Y. WATANABE, J. HA, and J. C. NEIDHOEFER (2006) “Real-time vision-based relative navigation,” in *AIAA Guidance, Navigation, and Control Conference*, pp. 4411–4456.
- [14] LANGELAAN, J. W. (2007) “State estimation for autonomous flight in cluttered environments,” *Journal of guidance, control, and dynamics*, **30**(5), pp. 1414–1426.
- [15] YANG, J., D. RAO, S. CHUNG, and S. HUTCHINSON (2011) “Monocular vision based navigation in GPS-denied riverine environments,” in *Proceedings of the AIAA Infotech@ Aerospace Conference, St. Louis, MO*.
- [16] WU, A. D., E. N. JOHNSON, M. KAESS, F. DELLAERT, and G. CHOWDHARY (2013) “Autonomous flight in GPS-denied environments using monocular vision and inertial sensors,” *Journal of Aerospace Information Systems*.
- [17] WATANABE, Y., E. N. JOHNSON, and A. J. CALISE (2007) “Stochastically optimized monocular vision-based guidance design,” in *Proc. of AIAA atmospheric flight mechanics conference*.
- [18] KOCH, A., H. WITTICH, and F. THIELECKE (2006) “A vision-based navigation algorithm for a VTOL-UAV,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, vol. 6, p. 11.
- [19] FREW, E. W., T. GERRITSEN, S. PLEDGIE, C. BRINTON, S. PATEL, and B. SCHWARTZ (2008) “Vision-Based Navigation for Airfield Surface Operation,” in *AIAA Guidance, Navigation, and Control Conference*, pp. 4189–94.
- [20] HASHIMOTO, H., A. D. WU, G. CHOWDHARY, and E. N. JOHNSON (2011) “Avionics integration for a fully autonomous self-contained indoor miniature unmanned aerial system,” *AIAA Infotech@ Aerospace*, pp. 2011–1611.
- [21] SARIPALLI, S., J. F. MONTGOMERY, and G. S. SUKHATME (2002) “Vision-based autonomous landing of an unmanned aerial vehicle,” in *Robotics and automation, 2002. Proceedings. ICRA’02. IEEE international conference on*, vol. 3, IEEE, pp. 2799–2804.
- [22] SARIPALLI, S. and G. S. SUKHATME (2007) “Landing a helicopter on a moving target,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, IEEE, pp. 2030–2035.
- [23] SANCHEZ-LOPEZ, J. L., J. PESTANA, S. SARIPALLI, and P. CAMPOY (2014) “An approach toward visual autonomous ship board landing of a vtol uav,” *Journal of Intelligent & Robotic Systems*, **74**(1-2), pp. 113–127.
- [24] TRUSKIN, B. and J. W. LANGELAAN (2013) “Vision-based Deck State Estimation for Autonomous Ship-board Landing,” in *Proceedings of the AHS Forum*.
- [25] KUHN, H. W. (1955) “The Hungarian method for the assignment problem,” *Naval research logistics quarterly*, **2**(1-2), pp. 83–97.
- [26] CHEN, R. H., A. GEVORKIAN, A. FUNG, W.-Z. CHEN, and V. RASKA (2011) “Multi-sensor data integration for autonomous sense and avoid,” in *AIAA Infotech at Aerospace Technical Conference*.
- [27] RINK, K. A. (1998) “Use of the auction algorithm for target-object map,” *AIAA*, **12**, pp. 709–719.

- [28] HWANG, I., H. BALAKRISHNAN, K. ROY, and C. TOMLIN (2007) “Multiple-target tracking and identity management with application to aircraft tracking,” *Journal of guidance, control, and dynamics*, **30**(3), pp. 641–653.
- [29] LEE, D. N., P. E. REDDISH, and D. RAND (1991) “Aerial docking by hummingbirds,” *Naturwissenschaften*, **78**(11), pp. 526–527.
- [30] LEE, D. N., J. A. SIMMONS, P. A. SAILLANT, and F. BOUFFARD (1995) “Steering by echolocation: a paradigm of ecological acoustics,” *Journal of Comparative Physiology A*, **176**(3), pp. 347–354.
- [31] WANG, Y. and B. J. FROST (1992) “Time to collision is signalled by neurons in the nucleus rotundus of pigeons,” .
- [32] LEE, D. N., M. N. DAVIES, P. R. GREEN, ET AL. (1993) “Visual control of velocity of approach by pigeons when landing,” *Journal of Experimental Biology*, **180**(1), pp. 85–104.
- [33] KENDOUL, F. (2014) “Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems,” *The International Journal of Robotics Research*, **33**(2), pp. 237–267.