# Nonlinear Relative Pose Estimation for Autonomous Shipboard Landing

Brian L. Beechinor

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Aeronautics and Astronautics

University of Washington

2020

Reading Committee:

Kristi Morgansen

Sam Burden

Program Authorized to Offer Degree:
Aeronautics and Astronautics

©Copyright 2020
Brian L. Beechinor

University of Washington

## Abstract

Nonlinear Relative Pose Estimation for Autonomous Shipboard Landing

Brian L. Beechinor

Chair of the Supervisory Committee:
Professor Kristi Morgansen
Department of Aeronautics and Astronautics

Shipboard aerial operations represent extreme and dynamic environments for landing aircraft. Rough seas and turbulent conditions lead to aircraft and vessel motions that can be large and seemingly unpredictable. To commence a safe landing, the relative position and attitude of the ship's deck must be known and monitored. These conditions present adverse and challenging situations which serve as the motivation for this thesis. In the work reported here, a method of estimating the relative pose necessary for autonomous tracking and landing, using dual quaternions, is presented. Two nonlinear estimators were developed in the dual quaternion framework to use a Plücker coordinate based lidar measurement model. Synthetic measurements were generated from independent point models representing the aircraft and the ship. The estimators were tested using the synthetic measurements, yielding insights into the methods while successfully tracking the relative pose.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

The author wishes to express sincere appreciation to his academic advisor, Professor Kristi Morgensen, for her guidance and patience. He also wishes to thank the current and past members of the Nonlinear Dynamics and Control Lab for the assistance they provided in this endeavour.

Unsik Lee, Sierra Adibi, John Berg, Ena Hodzic, Kimber Hinson, Trevor Avant, Jake Quenzer, Natalie Brace, Brian Katona, and Tyson Mulder

# DEDICATION

...to my amazing wife Lindsey, without whom this work would have never come to be.

# Chapter 1

# INTRODUCTION

Autonomous shipboard landing requires advances in relative position estimation to accurately identify the vehicle's current state during the demanding landing phase. This problem of relative pose estimation is difficult given large sea states, but it is tractable. The highly dynamic nature of two vehicles in close proximity under the harsh conditions at sea is what motivated this thesis. Focus is placed on computing an estimate of the landing platform's position and orientation in the landing aircraft's body frame. Two nonlinear relative pose estimators, making use of a lidar-based measurement model are developed to solve this problem. The relative pose is itself captured through the use of dual quaternions. A limited sensor suite and communication between the ship and the aircraft means that only bearings-to-deck features are available. This thesis describes an estimation methodology allowing for autonomous shipboard landing operations. Characterization of a scanning lidar sensor measurement model that detects the deck pose through the use of ship based beacons is described. A deck-state estimation solution is then presented based on two successful nonlinear Kalman estimators.

## *1.1 Motivation*

The first landing of a fully operational helicopter occurred in 1943; since then, thousands of pilots around the world have performed the critical maneuver on ships of varying sizes in varying conditions. What was demonstrated on that day was the helicopter's ability to perform a task that is beyond the capability of traditional fixed-wing aircraft. Attributable to the success of this demon-

stration was the unique ability to vertically take off and land (VTOL): the first in a class of vehicles to do so. The operational versatility of such aircraft manifested great interest, and has seen steady growth.

The intersection of vertical flight with marine-based missions has seen great interest from commercial and government entities. Many roles have been undertaken or investigated, ranging from humanitarian, commercial [7], defense, and science [6],[3] missions.

Dangers do exist, principally concerning the synchronization of the ship's deck with the aircraft. Critical aspects include the angle of the deck plane relative to a plane of the rotor system and the lateral motion of the deck relative to the axis of the aircraft's landing point. Dangerous situations can develop and lead to catastrophic accidents such as dynamic rollover. Rollover occurs when some factor causes the helicopter to roll or pivot around its landing gear, until a critical angle is reached. Beyond this point, main rotor thrust continues to accelerate the roll, regardless of the cyclic input, and recovery is impossible. If the thrust controlling collective is not completely reduced, the situation is exacerbated, causing the aircraft to rollover.

Whether piloted or autonomous, the landing phase of an aerial vehicle, onto the deck of a ship, in rough seas is one of the most unique and challenging maneuvers for a vertical take off and landing (VTOL) vehicle. Currently humans must contend with the various complexities that evolve in the landing phase. Through training, they must develop situational awareness of the wind state with respect to the ship, the motion of their vehicle with respect to the wind, and the overall motion of the ship's deck on which they intend to land. Crucial and time-sensitive decisions are based on this large stream of assimilated information. Humans have a limit to how much they can take in [18] which can lead to pilot overload. Current operations procedures [2] take this into account by limiting when, and under what conditions, air operations may occur. This produces the inherent drawback that limits the availability of the aerial resources. Increasingly as technology has progressed, autonomy has played a larger role; finding new was to assist, and possibly replace the pilots abilities while enabling further developments, with the ultimate goal of becoming fully

autonomous. As costs and risks decrease for these platforms, they will continuously be asked to do more.

The short final to wheels-on-deck phase of the mission is the primary interest of this thesis. The ability to estimate the relative pose between the aircraft and the vessel will enable operations from moving vessels of all sizes. The benefits and the dangers are clear; however, the functionality of such systems are still in question. Given the very nature of the marine environment—with wind and sea conditions fluctuating rapidly with time and location—the ability to launch and retrieve such vehicles autonomously becomes crucial. These are in effect close-proximity operations performed in a highly dynamic environment, operating in time scales of a few minutes. Importantly, a means of quickly finding accurate vehicle attitude and position is necessary during mission-critical maneuvers. Estimation is a fundamental aspect of aerospace systems and can play a key role in "drone" autonomy and mobility.

One of the biggest challenges in autonomous maritime operations is the need to continuously and accurately track the time varying relative position and attitude with respect to a moving target (the ship) in order to avoid collisions. This thesis investigates the problem of autonomously estimating the position and attitude of a ships deck with respect to a moving aircraft. Estimators treating rotation and translation require the rotation estimator before initiating the translational estimator, leading to delays. Dual quaternion estimation allows for combined convergence.

### *1.2 Problem Formulation*

Landing an aircraft on a small, constantly moving, platform presents a series of challenges. Ships performing seakeeping maneuvers in rough seas experience the randomness of wave actions on a large scale. Wave energy is transferred to the ship's hull, driving its motion in unpredictable ways. Unpredictable motions naturally lead to uncertainty in the deck's state at any given time. Additionally, aircraft with the ability to land vertically must contend with similar disturbances. Perturbations in the form of wind gusts and turbulence affect the vehicle in unplanned ways. Avoiding

high landing forces and moments that can lead to damage or destruction are the highest priority. Consequently, knowledge of the relative attitude and position of the ship's deck with respect to the aircraft is critical. While this may sound simple, it can be increasingly difficult in harsh weather conditions.

## *1.3 Background*

Research, as well as development, of systems and methodologies for position and attitude estimation has been pursued throughout the aerospace community. Grown out of a long history of operational necessity, these technologies have lead to development and commercialization, demonstrating a need has been met. As we are dealing with landing operations in particular, it is helpful to realize that this is merely part of a broader class of operations that a VTOL can perform, known as proximity operations.

### *1.3.1 Landing Platform Estimation*

Varied approaches to estimating the state of a ship's deck have been explored with great interest in recent years. Various methods utilize a sensor or grouping of sensors, either located on the ship, the aircraft, or some combination of the two. Currently deployed by the US Navy, Sierra Nevada Corporations UCARS-V2 utilizes a ship-mounted radar system for autonomous launch and landing operations of the Navys MQ-8B/C autonomous helicopter. This clearly represents a successful approach to the problem, seeing that it is in current active operation/status. While deployable, such an approach is expensive and requires intensive installations that may not be practicable for all UAV operators, or the ships they will land on. Another approach to securely landing a ship in rough seas–currently in operation crewed missions–makes use of a hauldown device. With a cable securely connecting the aircraft to the deck of the ship, a winch slowly draws the aircraft down to the deck. Instrumenting the cable was an approach for estimating the relative pose of the VTOL and the ship investigated in [28] using computer simulations to verifying this method.

The largest segment of research into this topic goes to the use of a vision system in one form or another, as the primary means of measurement. Such models as presented in [2-7] use forms of distinct fiducial markers acting as points of reference or a means to measure by. Often these markers are designed to allow for detection with relatively low latency, enabling detection and use in real time. These markers are cheap and relatively easy to deploy, which is an attractive feature, garnering much focus. Initially, [23] explored this for a fixed platform landing; position only estimation proceeded using an extended Kalman filter (EKF). A fixed target was replaced with a moving one in [26], albeit, motion was restrained to translation along a single dimension, using a linear Kalman filter. Position was again the only state being estimated. Direct tracking using measurements alone was the method in [10] to determine position, and attitude, producing promising results. With the inclusion of non-deterministic aspects, [25] extends the vision-based system by utilizing an extended Kalman filter. Development of the EKF included taking in visual cues as well as rate gyro data, providing the ability to estimate the pose of the ship's deck with respect to the aircraft's coordinate frame. The inherent drawback in vision system comes from bright-light conditions, which [13] rectified with the addition of a Lidar sensor. The camera tracked a single point marker in the form of a bright light, while the lidar scanned in an arc. From this, a plane representing the measurements was calculated. Vision-based bearing information was then fused with lidar-based ship's deck orientation relative to the aircraft. Experimental results obtained with a large autonomous rotorcraft and a 3DOF motion simulator that emulated the roll, pitch, and heave characteristics demonstrated this method's feasibility. Unlike the previous vision + lidar system, [5] requires no infrastructure on the ship's deck. The two systems operate independently of one another, enabling redundant tracking. The lidar performs full scans of the deck, which is then run through a particle filter to determine deck pose, while a camera system uses features preexisting on the deck to back out its orientation. Lidar was the sole source of measurement in [4], where multiple lidar sensors were used in various locations on the aircraft. Doing so meant position was abandoned, and deck attitude was the only state to be tracked. None of the works mentioned

above made use of quaternions for attitude, let alone dual quaternions for pose. For papers that did estimate pose, it was handled as a separate state, and thus required individual updates for position and attitude.

### *1.3.2 Relative Pose Estimation*

Quaternions, as representations of attitude, have their roots in the world of spacecraft proximity research and operations dating back to the 1970s. An extended Kalman filter using the unit quaternion was presented in [17], which went on to be used in several NASA missions. Benefits came from the reduction in computational burden by composing the state in terms of the three element vector, portion of the error quaternion. This reduction in quaternion terms unfortunately entails the same singularity issues associated with other attitude parameterizations such as Euler angles. It is expected however that the error between the true attitude and the estimated attitude will be close to zero, and nowhere near the 180 degree singularity that will cause it to fail. The only time where this is a concern is during the initialization phase of the estimator. An approach to avoiding the initialization issues with the EKF, as well as overall lower expected error, led [8]-[30] to make use of the unscented Kalman filter (UKF). Results demonstrated the ability for the UKF to indeed reduce the overall error, but computation time was increased due to the nature of the scheme. Given the nature of space programs, rendezvous and docking utilizing quaternions for attitude parameterizations became a focus. In these types of proximity operations, the relative states between the two spacecraft are of interest. The relative pose was investigated through the use of a vision based system in [34], where the relative attitude in quaternions and the relative position of the two vehicles were treated as separate states in the EKF. The introduction and successful demonstration of dual quaternions to handle the pose of the spacecraft was first used in in [11]. The authors were able to show the capabilities of such a parameterization which necessitated the inclusion of dual velocity terms to handle the rate gyro measurement models used. Extending the use of dual quaternions with the UKF occurred in [22], with the use of a vision measurement model coupled

with a rate gyro model. In their work they compared the same reduced state models for both the multiplicative EKF and multiplicative UKF systems. Unsurprisingly, what they found confirmed previous outcomes, whereby the estimates from the UKF were more accurate, but took longer time to compute.

### *1.4 Contributions*

This thesis addresses relative pose estimation for ship based landing operations using dual quaternions. Two methods of estimating the relative pose between the ship and the VTOL during the landing phase of aerial shipboard operations are presented. Both methods utilize a Lidar based measurement model with dual vector Plücker Coordinates. The first estimation method utilizes an adaption for the extended Kalman filter. The second method utilizes an adaption of the unscented Kalman filter (UKF). In both methods, various conditions are used to simulate the VTOL motion as well as the ship motion. Following the development and application of the measurement model to the two estimators, we investigate the capabilities of each estimator through a number of simulations.

### *1.5 Thesis Organization*

In Chapter 2, we discuss the mathematical background as it relates to pose. In Chapter 3, a measurement model as well as the kinematic model for the system based on dual quaternions is developed. In Chapter 4, motion trajectories for both the ship and the VTOL aircraft are developed, which in turn provide to the synthetic measurements to be fed into the two estimators. Finally, Chapter 5 contains concluding remarks as well as future work encouraged by this effort.

# Chapter 2

# MATHEMATICAL PRELIMINARIES

This chapter serves as a means to introduce the attitude and eventual pose parameters to be used in this thesis. We review the quaternion for attitude representations, then make the natural progression to dual quaternions. Beginning with the general formulations, their attributes are expanded with the associated algebras, before discussing their transformation (rotational) properties. Through review of the main properties, a better understanding of the mathematical underpinnings of the work presented in later chapters will be apparent. The main properties required are presented, with further sources available regarding the benefits of quaternions and dual quaternions to be found in [27], [20], and [16].

*Attitude Parameterizaton*

It should be noted that several attitude parameterizations and their corresponding transformations exist for a rigid body in three dimensional space. These include the direction cosine matrix representation, the Euler angle representation, and the quaternion representations. Each method has advantages and disadvantages in how it represents attitude. Table 2.1 summarizes some of these, with the quaternion being the chosen method, as is discussed fully later on.

| Parameter | Advantages | Disadvantages | Dimension | Singularity | Uniqueness | Constraints |
|---|---|---|---|---|---|---|
| DCM ($C_{ij}$) | no trigonometric functions | 6 redundant parameters | 9 | none | yes | $C^TC$ = 1 |
| Euler Angles ($\theta_i$) | no redundant parameters | trigonometric functions of $\theta_i$ | 3 | $\theta_2 = \pm\frac{\pi}{2}$ | no | none |
| Quaternions ($q_i$) | algebraic functions | one redundant parameter | 4 | none | no | $q^Tq$ = 1 |

Table 2.1: Attitude representation alternatives and their properties

## *2.1 Quaternions*

The Irish mathematician William Rowan Hamilton, Ireland's Astronomer Royal, was the first to describe quaternions and develop their properties. Hamilton was fascinated with complex numbers, and was attempting to expand the complex numbers from its two dimensional description of the complex plane, to higher dimensions. On the well documented day of Monday the 16th of October, he arrived at a means to accomplish this task. He realized that rather than have three parameters of which two were complex, he would need three complex elements, resulting in a four parameter representation. The three complex elements would be grouped together and given the title of vector for the first time in history. The fourth term would remain as the real part. With his four parameters at hand, Hamilton quickly developed the algebra to support this new object, which is to say that formal rules of arithmetical operations are valid for the objects defined. Hamilton spent the rest of his life working on quaternions, which became the first non-commutative algebra to be studied.

### *2.1.1 Quaternion Definition*

Hamiltons fundamental accomplishment came from defining the following properties:

$$i^2 = j^2 = k^2 = ijk = -1. \tag{2.1}$$

From this axiom, the following products can be derived:

$$\begin{aligned} ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j. \end{aligned} \tag{2.2}$$

The quaternion is then the expression in the general form of

$$\mathbf{q} = q_1 i + q_2 j + q_3 k + q_4 \tag{2.3}$$

Where $q_1$, $q_2$, $q_3$, $q_4 \in \mathbb{R}$, and $i$, $j$, $k \in \mathbb{C}$. Hamilton described this arrangement as consisting of the vector, or imaginary, component and the scalar, or real, component.

*Representations*

There are several ways to denote the quaternion object. It is important to note that the following are equivalent representations of the four parameters.

- **Hyper Complex**

$$\mathbf{q} = q_1 i + q_2 j + q_3 k + q_4 \tag{2.4}$$

- **Vector + Scalar Sum**

$$\mathbf{q} = \mathbf{q}_v + q_s \tag{2.5}$$

- **Ordered Pair**

$$\mathbf{q} = (\mathbf{q}_v, q_s) \tag{2.6}$$

- **Euler Parameters**

$$\mathbf{q} = (q_1,\ q_2,\ q_3,\ q_4) \tag{2.7}$$

- **Four Vector**

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{q}_v \\ q_s \end{bmatrix} \tag{2.8}$$

where $\mathbf{q}_v = [q_1\ q_2\ q_3]^T$ represents the vector component of the quaternion and $q_s = q_4$, the scalar part.

Placing the quaternion in vector form allows the use for matrix algebra operations involving quaternions, and will be featured heavily. If $\mathbf{q}_v = 0$, then it is considered a *real* quaternion. Conversely, if the $q_s = 0$, then it is considered a *pure* quaternion.

$$\text{real, } q_s = \begin{bmatrix} \mathbf{0_v} \\ q_s \end{bmatrix} \qquad \text{pure, } \mathbf{q}_v = \begin{bmatrix} \mathbf{q}_v \\ 0 \end{bmatrix} \tag{2.9}$$

We can then define the set of quaternions as

$$\mathbb{H} = \{\mathbf{q} : q_1 i + q_2 j + q_3 k + q_4, q_1, q_2, q_3, q_4 \in \mathbb{R}\} \tag{2.10}$$

### *2.1.2 Quaternion Algebra*

In this section we will provide an introduction to the quaternion algebras, which provide a convenient mathematical framework for attitude. Let $\mathbf{q}, \mathbf{p}, \mathbf{r} \in \mathbb{H}$ be three quaternions, let $\lambda \in \mathbb{R}$ represent a scalar value.

#### *Quaternion Addition*

Vector and scalar components are added individually,

$$\begin{aligned} \mathbf{q} + \mathbf{p} &= (\mathbf{q}_v + q_s) + (\mathbf{p}_v + p_s) \\ &= (\mathbf{q}_v + \mathbf{p}_v) + (q_s + p_s) \end{aligned} \tag{2.11}$$

By construction, the sum is commutative and associative,

$$\begin{aligned} \mathbf{q} + \mathbf{p} &= \mathbf{p} + \mathbf{q} \\ \mathbf{q} + (\mathbf{p} + \mathbf{r}) &= (\mathbf{q} + \mathbf{p}) + \mathbf{r} \end{aligned} \tag{2.12}$$

#### *Scalar Multiplication*

For a given scalar $\lambda \in \mathbb{R}$,

$$\begin{aligned} \lambda\mathbf{q} &= \lambda(\mathbf{q}_v + q_s) \\ &= \lambda\mathbf{q}_v + \lambda q_s. \end{aligned} \tag{2.13}$$

*Quaternion Multiplication*

Under the definition in (2.1)-(2.2), and the introduction of the quaternion product symbol $\otimes$, we are able to perform quaternion multiplication:

$$\mathbf{q} \otimes \mathbf{p} = \begin{bmatrix} q_s \mathbf{p}_v + p_s \mathbf{q}_v + \mathbf{q}_v \times \mathbf{p}_v \\ q_s p_s - \mathbf{q}_v \cdot \mathbf{p}_v \end{bmatrix}. \tag{2.14}$$

Represented in vector and scalar components, we see the non-commutativity aspect due to the cross product in the vector portion. However if one of the quaternions is *real* or if the vector parts are parallel, they may commute. The product is both associative and distributive over summation with another quaternion:

$$\mathbf{q} \otimes \mathbf{p} \neq \mathbf{p} \otimes \mathbf{q}, \tag{2.15}$$

$$\mathbf{q} \otimes (\mathbf{p} \otimes \mathbf{r}) = (\mathbf{q} \otimes \mathbf{p}) \otimes \mathbf{r}, \tag{2.16}$$

$$\mathbf{q} \otimes (\mathbf{p} + \mathbf{r}) = \mathbf{q} \otimes \mathbf{p} + \mathbf{q} \otimes \mathbf{r} \qquad \text{and} \qquad (\mathbf{q} + \mathbf{p}) \otimes \mathbf{r} = \mathbf{q} \otimes \mathbf{r} + \mathbf{p} \otimes \mathbf{r}. \tag{2.17}$$

Lastly, a bi-linear relation exists that can be expressed as two matrix products. By expanding (2.14) out into the full array, it is possible to express one quaternion as a matrix, and the other as its four element vector,

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} &= [\mathbf{q}]_L \mathbf{p} \\ &= [\mathbf{p}]_R \mathbf{q}. \end{aligned} \tag{2.18}$$

The bracketed terms are matrices, with the subscript $[\bullet]_L$ representing the left-quaternion product and $[\bullet]_R$ representing the right-quaternion product. Thus the product of two quaternions can be

expressed in two different forms,

$$[\mathbf{q}]_L = \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} = \begin{bmatrix} [\mathbf{q}_v^{\times}] + q_s I_{3x3} & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_s \end{bmatrix} \in \mathbb{R}^{4\times 4}$$
$$[\mathbf{q}]_R = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} = \begin{bmatrix} [\mathbf{q}_v^{\times}]^T + q_s I_{3x3} & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_s \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{2.19}$$

where a cross product matrix is represented by the skew operator such that for $\alpha = [\alpha_1 \; \alpha_2 \; \alpha_3]^T$,

$$[\alpha^{\times}] = \begin{bmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{bmatrix}. \tag{2.20}$$

This turns out to be a skew-symmetric matrix, meaning that its transpose is equal to its negative, and can be used interchangeably with the cross product operator such that,

$$[\alpha^{\times}]\beta = \alpha \times \beta, \qquad \forall\, \alpha, \beta \in \mathbb{R}^3. \tag{2.21}$$

Then, returning to the associative property for quaternion multiplication, we can re-write it in terms of left- and right-product matrices,

$$\begin{aligned} \mathbf{q} \otimes (\mathbf{p} \otimes \mathbf{r}) &= [\mathbf{q}]_L [\mathbf{r}]_R \mathbf{p} \\ (\mathbf{q} \otimes \mathbf{p}) \otimes \mathbf{r} &= [\mathbf{r}]_{\mathbf{R}} [\mathbf{q}]_{\mathbf{L}} \mathbf{p}. \end{aligned} \tag{2.22}$$

This in turn means that the left and right product matrices commute

$$[\mathbf{q}]_L [\mathbf{r}]_R = [\mathbf{r}]_{\mathbf{R}} [\mathbf{q}]_{\mathbf{L}}. \tag{2.23}$$

*Quaternion Identity*

The identity quaternion is defined as,

$$I_{\mathbf{q}} = [0\ 0\ 0\ 1]^T \text{ such that, } \mathbf{q} \otimes I_{\mathbf{q}} = I_{\mathbf{q}} \otimes \mathbf{q} = \mathbf{q}. \tag{2.24}$$

*Quaternion Conjugate*

A complex number with an equal real part and an imaginary part equal in magnitude but opposite in sign is known as a complex conjugate. Under a similar definition, we can extend this concept to quaternions; the imaginary, or vector, portion is of equal magnitude, but opposite in sign,

$$\mathbf{q}^* = \begin{bmatrix} -\mathbf{q}_v \\ q_s \end{bmatrix}. \tag{2.25}$$

This leads to the property through the quaternion product that leads to a *real* quaternion result,

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = \begin{bmatrix} \mathbf{0}_v \\ q_1^2 + q_2^2 + q_3^2 + q_4^2 \end{bmatrix}. \tag{2.26}$$

The conjugate of a product of two quaternions is the products of the conjugates, with the order reversed,

$$(\mathbf{q} \otimes \mathbf{p})^* = \mathbf{p}^* \otimes \mathbf{q}^*. \tag{2.27}$$

*Quaternion Norm*

A quaternion norm returns a scalar, defined as,

$$||\mathbf{q}|| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\mathbf{q}^* \otimes \mathbf{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}. \tag{2.28}$$

Therefore the norm of the quaternion product is the product of the two quaternion norms,

$$||\mathbf{q} \otimes \mathbf{p}|| = ||\mathbf{q}|| \otimes ||\mathbf{p}||. \tag{2.29}$$

*Quaternion Inverse*

The inverse of a quaternion is obtained through the normalization of the quaternion conjugate,

$$\mathbf{q} = \frac{\mathbf{q}^*}{||\mathbf{q}||}. \tag{2.30}$$

*Unit Quaternion*

A unit quaternion is one that has been divided by its norm, thus $||\mathbf{q}|| = 1$. Then for all unit quaternions, the inverse is equal to the conjugate as they have norm one,

$$\mathbf{q}^{-1} = \mathbf{q}^*. \tag{2.31}$$

It is with the unit quaternion that transformations and rotations are possible. Through the use of the conjugate quaternion, the inverse rotation and/or transformation can be achieved. As any rotation can be described by three parameters, the unit norm constraint is imposed on quaternions for attitude representation, where it is closed under multiplication, but not addition.

### *2.1.3 Quaternion Rotations*

Euler's principle rotation theorem [19] states that a rigid body, or coordinate reference frame, can be expressed as a pure rotation or sequence of rotations, through some angle about a fixed axis. The vector portion of the quaternion contains three elements, the normalized vector acts as the axis about which rotation is performed. The scalar element $q_4$ determines the amount of rotation to be executed about the vector part. Unit quaternions parameterize the pure rotational amount and rotation axis when we define the unit quaternion as

$$\mathbf{q} = \begin{bmatrix} \mathbf{n}\sin(\frac{\phi}{2}) \\ \cos(\frac{\phi}{2}) \end{bmatrix}, \tag{2.32}$$

where the Euler axis of rotation is expressed by the unit vector $\mathbf{n} = [n_x \; n_y \; n_z]^T$, and $\phi$ denotes the rotation angle about that axis. Note the definition's use of half angles; the reason arises from

rotations occurring in a two dimensional plane. With quaternions, their four dimensions make it possible to rotate in two independent planes simultaneously by the same angle. For example, we can create the parameterization of $q' = \cos\theta + \mathbf{n}\sin\theta$ with $\mathbf{n} = [i\ 0\ 0]^T$, and $\theta = \frac{\pi}{2}$ meaning we wish to rotate about the x-axis by 90 degrees. Then, the transformation is $q' = i$, and applying it to our quaternion gets,

$$\begin{aligned} q'' = q'q &= i(w + xi + yj + zk) \\ &= -x + wi - zj + yk. \end{aligned} \tag{2.33}$$

From this initial rotation, the (w,x) components rotated 90 degrees to become (-x,w). Likewise, the (y,z) components also rotated 90 degrees to become (-z,y). For rotations in three dimensions, we need to be able to rotate in a single plane that it doesn't contain the scalar component w (the axis of real numbers). By moving the transformation around to the right side, a different result is found, with results summarized in Fig. 2.1. From the figure we see that both left and right multiplication yields the selected plane about which we wish to rotate. Unfortunately, the rotation was twice the desired value. To reduce the rotation angle, the full angle is halved

$$\theta = \frac{\phi}{2}. \tag{2.34}$$

Therefore to rotate a vector $\mathbf{v} = [v_x\ v_y\ v_z\ 0] \in \mathbb{R}^4$, it is multiplied on the left by (2.32), and the right by the conjugate of (2.32),

$$\mathbf{v}' = \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^*. \tag{2.35}$$

By using the left and right hand matrices of (2.19), we can restate (2.35),

$$\begin{aligned} \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^* &= [\mathbf{q}^*]_R[\mathbf{q}]_L\mathbf{v} \\ &= A(\mathbf{q})\mathbf{v} \end{aligned} \tag{2.36}$$

where $A(\mathbf{q})$ is the quaternion attitude matrix that is formally,

$$A(\mathbf{q}) = (q_s^2 - \mathbf{q}_v^T\mathbf{q})\mathbf{I} + 2\mathbf{q}_v\mathbf{q}_v^T + 2q_s[\mathbf{q}_v^\times]. \tag{2.37}$$

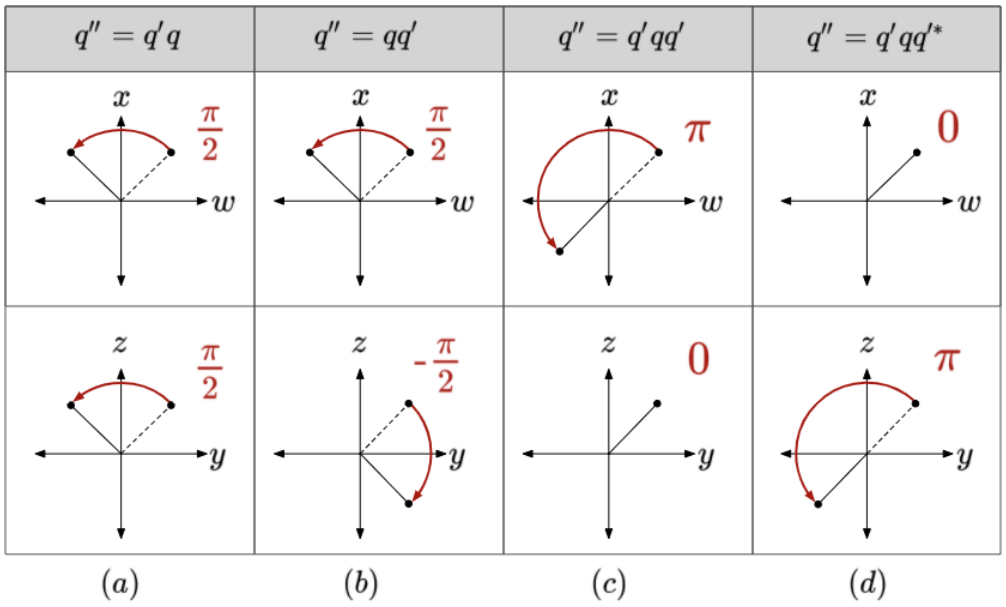


Figure 2.1: Rotation of quaternion by $\frac{\pi}{2}$, about i. (a) Left side multiplication: 90 degrees in both planes (b) Right side multiplication: 90 degrees in both planes, but opposite directions (c) Left and right side multiplication: 180 degrees, in wrong plane (d) Left and right-conjugate multiplication: 180 degrees, correct plane.

The attitude matrix has the following properties:

$$A([0\ 0\ 0\ 1]^T) = \mathbf{I} \tag{2.38}$$

$$A(-\mathbf{q}) = A(\mathbf{q}) \tag{2.39}$$

$$A(\mathbf{q}^*) = A(\mathbf{q})^T \tag{2.40}$$

$$A(\mathbf{q} \otimes \mathbf{p}) = A(\mathbf{q})A(\mathbf{p}). \tag{2.41}$$

### *2.1.4 Rotational Operator Direction*

As coordinates and vectors are transformed and rotated, the notion of direction becomes important. In the most general case, there exist local coordinate frames, and their global counterpart. These

are relative, given the specific task or mission,

$$v_G = \mathbf{q}_{G/L} \otimes v_L \otimes \mathbf{q}^*_{G/L} = \mathbf{q}^*_{L/G} \otimes v_L \otimes \mathbf{q}_{L/G} \qquad (\textit{Global} \leftarrow \textit{Local}) \tag{2.42}$$

$$v_L = \mathbf{q}_{L/G} \otimes v_G \otimes \mathbf{q}^*_{L/G} = \mathbf{q}^*_{G/L} \otimes v_L \otimes \mathbf{q}_{G/L}. \qquad (\textit{Local} \leftarrow \textit{Global}) \tag{2.43}$$

It can then be observed that,

$$\mathbf{q}_{L/G} = \mathbf{q}^*_{G/L}. \tag{2.44}$$

### *2.1.5 Rotational Compositions*

In addition to direction, it is important to think of how multiple rotations or transformations can by strung together. Composing quaternions is similar to the composition of rotational matrices. With the direction chosen, and using the associative property of the quaternion product, it is possible to chain successive relations together. From Fig. 2.2, we see that there are three reference frames given: A, B, and C.

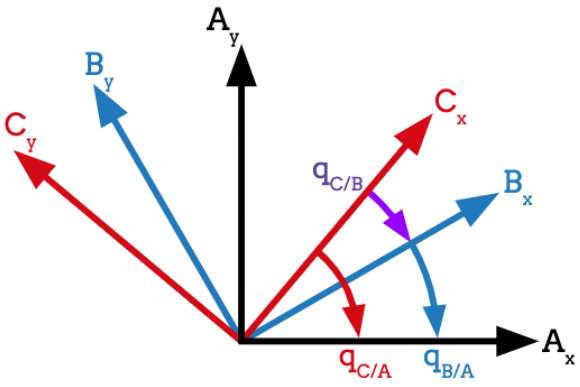


Figure 2.2: Composition of rotations from the C and B frames to the A frame.

The direction the attitude arrow is pointing, represents the frame with which the object is expressed in. Thus, $\mathbf{q}_{C/A}$ should be read as the quaternion that goes from the C-frame to the A-frame,

expressed in the A-frame. Utilizing (2.35), with the quaternions depicted above,

$$\mathbf{q}_{C/A} = \mathbf{q}_{B/A} \otimes \mathbf{q}_{C/B} \tag{2.45}$$

Then, the quaternion that takes you from the C-frame to the A-frame, expressed in the A-frame is equal to the the quaternions that goes from the B-frame to the A-frame, expressed in the B-frame, times the quaternion going from the C-frame to the B-frame, expressed in the B-frame. When dealing with vector transformations, the composition is written as

$$x_A = \mathbf{q}_{A/B} \otimes x_B \otimes \mathbf{q}^*_{A/B} \tag{2.46}$$

However, the vector in the B-frame is in itself a composition that originated in the C-frame,

$$x_B = \mathbf{q}_{C/B} \otimes x_C \otimes \mathbf{q}^*_{C/B}. \tag{2.47}$$

Making use of (2.17) and (2.27), we arrive at

$$\begin{aligned} x_A &= \mathbf{q}_{A/B} \otimes x_B \otimes \mathbf{q}^*_{A/B} \\ &= \mathbf{q}_{A/B} \otimes (\mathbf{q}_{C/B} \otimes x_C \otimes \mathbf{q}^*_{C/B}) \otimes \mathbf{q}^*_{A/B} \\ &= \mathbf{q}_{A/B} \otimes \mathbf{q}_{C/B} \otimes x_C \otimes (\mathbf{q}_{A/B} \otimes \mathbf{q}_{C/B})^* \\ &= \mathbf{q}_{A/C} \otimes x_C \otimes \mathbf{q}^*_{A/C}. \end{aligned} \tag{2.48}$$

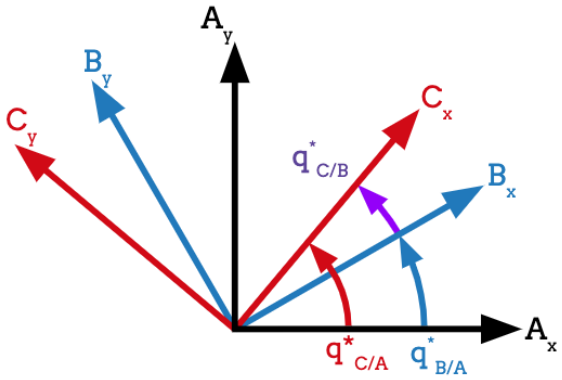


Figure 2.3: Conjugate composition of rotations from the A and B frames to the C frame.

### *2.1.6 Quaternion Kinematics*

The quaternions derivative can be written as either

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}^I \tag{2.49}$$

or

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\omega}^B \otimes \mathbf{q} \tag{2.50}$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_v{}^T 0]^T = [\omega_x\ \omega_y\ \omega_z\ 0]^T$ is the four element angular velocity vector, and specifically $\boldsymbol{\omega}^I$ is the angular velocity, with respect to the I-frame. Likewise, $\boldsymbol{\omega}^B$ represents the angular velocity of the B-frame.

### *2.1.7 Conventions*

The definition, and resulting algebra, for the quaternion is not singular. Decisions made regarding four areas will impact formulas for rotation, composition, and direction. Unfortunately specific conventions are not always directly stated, and it is up to the reader to discern what conventions are chosen. The primary factors responsible for the outcomes are listed below.

1. **Handedness** - Choice of left, or the more standard right hand system. This is decided with the multiplication formula for the imaginary components $i,\ j,\ k \in \mathbb{C}$. The subtlety lies in the order of the terms that define $k$

$$ij = \text{-}ji = k \qquad \text{vs} \qquad ji = \text{-}ij = k. \tag{2.51}$$

2. **Order of the Elements** - The two variations are scalar-real part first, or scalar-real part last

$$\mathbf{q} = \begin{bmatrix} q_s \\ \mathbf{q}_v \end{bmatrix} \qquad \text{vs} \qquad \mathbf{q} = \begin{bmatrix} \mathbf{q}_v \\ q_s \end{bmatrix}. \tag{2.52}$$

3. **Rotation Operator** - The decision regarding this factor is less severe, as it is often referenced as the difference between an attitude matrix and a rotation matrix

$$\text{Vector Transformation} \qquad \text{vs} \qquad \text{Frame Transformation.} \tag{2.53}$$

4. **Direction** - This factor is only for when dealing with the frame transformation from before. This is important as seen with the kinematic equation where the angular velocity term will switch position depending on what frame it is from

$$\text{Global} \leftarrow \text{Local} \qquad \text{vs} \qquad \text{Local} \leftarrow \text{Global.} \tag{2.54}$$

While this may seem trivial, it has caused difficulty in the past as reviewed in [32] and [33], which discuss how NASA has used multiple conventions concurrently in its history. Additionally as multiple disciplines have increased their use of quaternions, they have different requirements, and use different conventions. A fact that has created some ire amongst researchers as discussed in [29] who point out that software packages with built in functions may be using a convention different from your own.

### *2.2 Dual Quaternions*

Dual numbers were introduced by W. K. Clifford in 1873, 30 years after Hamilton's quaternion was brought into the world. He defined his creation as,

$$z = r + \epsilon d. \tag{2.55}$$

It is, once again, easy to draw similarities between this parameterization and complex numbers. Where complex numbers have a real and a complex part, dual numbers have real and dual parts. This ordered pair of real numbers is joined through the dual operator, $\epsilon$, that has the following properties.

$$\begin{aligned} \epsilon^2 &= 0 \\ \epsilon &\neq 0 \end{aligned} \tag{2.56}$$

Dual numbers were quickly expanded to include dual vectors, and then combined with unit quaternions to form dual quaternions. With mathematical properties retained, the dual quaternion grew to consists of eight elements. Hamilton's creation was able to handle rotations, while the dual quaternion is capable of handling rotations, with the incorporation of translations in a unified state variable. This elegant form has led dual quaternions to be the most compact and efficient form of rigid body transformation representation. This makes explicit transformation of position, rotations, and translations unnecessary under the single representation.

### *2.2.1 Dual Quaternion Definition*

The dual quaternion is formally defined as,

$$\widetilde{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d. \tag{2.57}$$

The set of dual quaternions may then be stated as

$$\mathbb{Q} = \{\widetilde{\mathbf{q}} : \mathbf{q}_r + \epsilon \mathbf{q}_d, \mathbf{q}_r, \mathbf{q}_d \in \mathbb{H}\}. \tag{2.58}$$

## *2.3 Dual Quaternion Algebra*

Let $\widetilde{\mathbf{q}}$, $\widetilde{\mathbf{p}} \in \mathbb{Q}$ be two dual quaternions, and $\lambda \in \mathbb{R}$ represent a scalar value, we then have the following mathematical operations for dual quaternions.

*Dual Quaternion Addition*

$$\widetilde{\mathbf{q}} + \widetilde{\mathbf{p}} = (\mathbf{q}_r + \mathbf{p}_r) + \epsilon(\mathbf{q}_d + \mathbf{p}_d) \tag{2.59}$$

*Scalar-Dual Quaternion Multiplication*

$$\begin{aligned} \lambda\widetilde{\mathbf{q}} &= \lambda(\mathbf{q}_r + \epsilon \mathbf{q}_d) \\ &= \lambda\mathbf{q}_r + \epsilon\lambda\mathbf{q}_d \end{aligned} \tag{2.60}$$

*Dual Quaternion Multiplication*

$$\widetilde{\mathbf{q}} \otimes \widetilde{\mathbf{p}} = (\mathbf{q}_r \otimes \mathbf{p}_r) + \epsilon(\mathbf{q}_r \otimes \mathbf{p}_d + \mathbf{q}_d \otimes \mathbf{p}_r) \tag{2.61}$$

In vector form, this becomes,

$$\widetilde{\mathbf{q}} \otimes \widetilde{\mathbf{p}} = \begin{bmatrix} \mathbf{q}_r \otimes \mathbf{p}_r \\ \mathbf{q}_r \otimes \mathbf{p}_d + \mathbf{q}_d \otimes \mathbf{p}_r \end{bmatrix} \tag{2.62}$$

Similar to what was seen with quaternions, dual quaternions multiplication turn out to be linear, thus we can represent it as a matrix-vector multiplication process as well,

$$\widetilde{\mathbf{q}} \otimes \widetilde{\mathbf{p}} = \begin{bmatrix} [\mathbf{q}_r]_L & 0_{4\times4} \\ [\mathbf{q}_d]_L & [\mathbf{q}_r]_L \end{bmatrix} \begin{bmatrix} \mathbf{p}_r \\ \mathbf{p}_d \end{bmatrix}. \tag{2.63}$$

We then have the similar left and right side multiplication matrices,

$$[\widetilde{\mathbf{q}}]_L = \begin{bmatrix} [\mathbf{q}_r]_L & 0_{4\times4} \\ [\mathbf{q}_d]_L & [\mathbf{q}_r]_L \end{bmatrix} \tag{2.64}$$

$$[\widetilde{\mathbf{q}}]_R = \begin{bmatrix} [\mathbf{q}_r]_R & 0_{4\times4} \\ [\mathbf{q}_d]_R & [\mathbf{q}_r]_R \end{bmatrix} \tag{2.65}$$

*Dual Quaternion Identity*

$$\mathbf{I}_{\widetilde{\mathbf{q}}} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T \tag{2.66}$$

$$\widetilde{\mathbf{q}} \otimes \mathbf{I}_{\widetilde{\mathbf{q}}} = \mathbf{I}_{\widetilde{\mathbf{q}}} \otimes \widetilde{\mathbf{q}} = \widetilde{\mathbf{q}} \tag{2.67}$$

*Dual Quaternion Conjugate*

The conjugate of the dual quaternion is essentially and extension of the quaternion conjugate,

$$\widetilde{\mathbf{q}}^* = \mathbf{q}_r^* + \epsilon\mathbf{q}_d^*. \tag{2.68}$$

Then, when written in vector form, it can be seen that the quaternion conjugate is applied ot the real and dual parts of the dual quaternion,

$$\widetilde{\mathbf{q}}^* = \begin{bmatrix} \mathbf{q}_r^* \\ \mathbf{q}_d^* \end{bmatrix}. \tag{2.69}$$

*Dual Quaternion Norm*

$$\begin{aligned} ||\widetilde{\mathbf{q}}|| &= \widetilde{\mathbf{q}}^* \otimes \widetilde{\mathbf{q}} = \widetilde{\mathbf{q}} \otimes \widetilde{\mathbf{q}}^* \\ &= \begin{bmatrix} \mathbf{I}_{\mathbf{q}} \\ 0_{4\times 1} \end{bmatrix} \end{aligned} \tag{2.70}$$

where the identity quaternion vector represents the norm of the unit dual quaternion.

*Dual Quaternion Inverse*

$$\widetilde{\mathbf{q}}^{-1} = ||\widetilde{\mathbf{q}}||^{-1}\widetilde{\mathbf{q}}^* \tag{2.71}$$

*Unit Dual Quaternion*

The unit dual-quaternion is key to the representation of any rigid rotational and translational transformations. Starting with the general arrangement,

$$\widetilde{\mathbf{q}} = \mathbf{q}_r + \epsilon\mathbf{q}_d \tag{2.72}$$

The rigid rotational and translational information for the unit dual-quaternion comes from,

$$\begin{aligned} \mathbf{q}_r &= \mathbf{q} \\ \mathbf{q}_d &= \frac{1}{2}\mathbf{r}^I \otimes \mathbf{q} = \frac{1}{2}\mathbf{q} \otimes \mathbf{r}^B, \end{aligned} \tag{2.73}$$

where $\mathbf{q}$ is a unit quaternion representing rotation and $\mathbf{r}$ is the quaternion describing the translation represented by the vector

$$\mathbf{r} = [r_x\ r_y\ r_z\ 0]^T = [\mathbf{r}_v^T\ 0]^T. \tag{2.74}$$

This means that to recover information once it has been placed into a dual quaternion you must perform the following,

$$\begin{aligned} \mathbf{r}^I &= 2\mathbf{q}_d \otimes \mathbf{q}_r \\ \mathbf{r}^B &= 2\mathbf{q}_r \otimes \mathbf{q}_d \end{aligned} \tag{2.75}$$

However, when extracting position information from the unit dual quaternion, it should be noted that $\mathbf{r}_B$ and $\mathbf{r}_I$ are represented in two different frames, yet they are still with respect to some common inertial frame. Therefore they will just be the opposites of one another.

### *2.3.1 Dual Quaternion Rotations and Vector Transformations*

Given the very nature of the dual quaternions, we are able to provide both translations and rotations depending on what is used. If we wish to construct a dual quaternion that only has a rotation, then it is of the form,

$$\widetilde{\mathbf{q}} = \begin{bmatrix} \mathbf{q}_r \\ 0_{4\times 1} \end{bmatrix} \tag{2.76}$$

and a dual quaternion that is pure translation will have,

$$\widetilde{\mathbf{q}} = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{q}_d \end{bmatrix} \tag{2.77}$$

It should come as no surprise that to achieve pure version of one motion over the other, all that is needed is to have either the real part or the dual part removed. Given a unit dual quaternion, and regular transforms are done in a the same fashion as quaternions.

### *2.3.2 Dual Quaternion Kinematics*

The unit dual quaternion kinematic equation is then given by

$$\dot{\widetilde{\mathbf{q}}} = \frac{1}{2}\widetilde{\mathbf{q}} \otimes \tilde{\boldsymbol{\omega}} \tag{2.78}$$

where we must introduce the dual velocity term that is composed of the angular and the translations velocity,

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix} \tag{2.79}$$

and

$$\omega = \begin{bmatrix} \omega_v \\ 0 \end{bmatrix} \in \mathbb{R}^4 \qquad \text{and,} \qquad \mathbf{v} = \begin{bmatrix} \mathbf{v}_v \\ 0 \end{bmatrix} \in \mathbb{R}^4. \tag{2.80}$$

For this section, quaternions were represented as a boldface $\mathbf{q}$, while the dual quaternion was represented by the boldface $\widetilde{\mathbf{q}}$ with the tilde above it. From this point on, the tilde will be removed, and it should be assumed that any $\mathbf{q}$ moving forward will be a dual quaternion, unless otherwise noted.

# Chapter 3

# ESTIMATION

Control laws require a closed loop feedback pathway to correct for errors that may cause instability, deviations from desired trajectories, or deviations from other desired response characteristics. Direct measurement of output values are not always possible, as the required sensors may not acquire the exact information needed. If a sensor does exist to capture the desired information, the physical elements and imperfections of a device means that noise will certainly be present. Estimation is then a successive and dynamic process that allows for the extraction of a richer set of information than the observed signals provide.

The most common estimation technique, used for a wide variety of dynamic systems, is the Kalman filter. Introduced in the seminal paper published in 1960 [15], Rudolph Kalman presented his solution to the linear-quadratic problem; an effective filter that recursively estimates a linear dynamic system's state from a series of noisy measurements. His solution required five steps:

1. State Estimate Propagation

2. Covariance Estimate Propagation

3. Filter Gain Computation

4. State Estimate Update

5. Covariance Estimate Update

The Kalman estimator makes an optimum mean square estimate of the system state by taking into account known system dynamics and their effects on Gaussian white noise. Knowledge of

the state-observation relationship in conjunction with the observation error characteristics are also necessary, as an estimate of the estimation-error is also provided.

For pose estimation, several Kalman estimator derivatives have been developed over the years [17],[14]. This chapter describes two types of Kalman filters, the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). We will develop the system model, including the selection of state variables, the relative kinematics involved, as well as the measurement model to be used.

### *3.1 Relative Dual Quaternion Kinematics*

We begin our process by developing the equations needed to describe the system model. Conventional kinematic equations can be used to independently describe the relative translational and rotation motions of objects in proximity to one another.

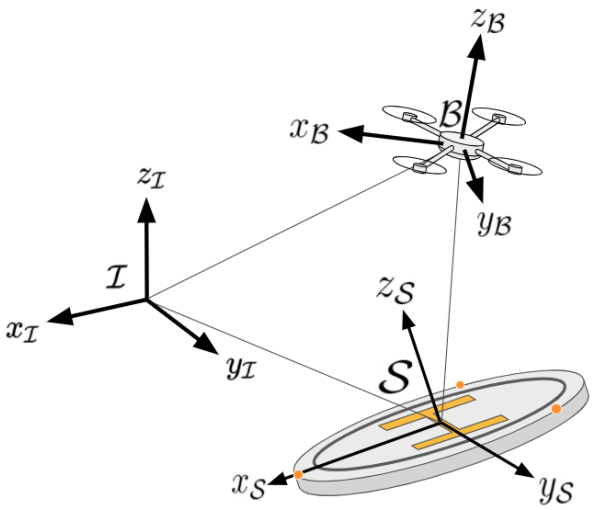


Figure 3.1: Depiction of the general arrangement of vehicles and their respective coordinate frames.

As we are interested in how the ship is moving relative to the VTOL/UAV, we can determine the correct quaternion to be used to achieve this. The relative attitude dynamics between the two rigid

bodies are defined using quaternion multiplication. From the section on quaternion rotations, and general compositions of attitude, the following figure helps to discern the required relative attitude.

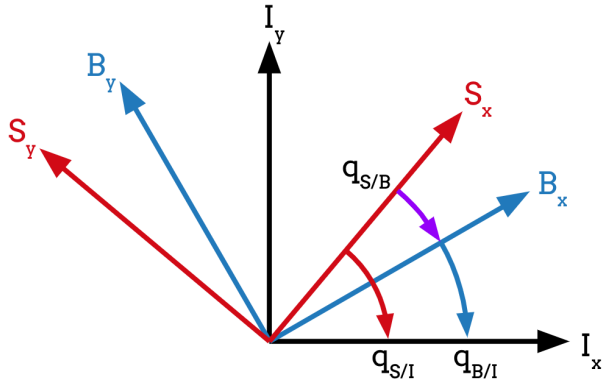


Figure 3.2: Composition of rotations from the ship frame (S-frame), to the body frame (B-frame), to the inertial frame (I-frame).

From the previous chapter, we can compose the following rotation sequence

$$\mathbf{q}^S_{S/I} = \mathbf{q}^B_{B/I} \otimes \mathbf{q}^B_{S/B} \tag{3.1}$$

where $\mathbf{q}^S_{S/I}$ is the dual quaternion of the ship, $\mathbf{q}^B_{B/I}$ is the dual quaternion of the VTOL/UAV, and $\mathbf{q}^B_{S/B}$ is dual quaternion of the ship with respect to the VTOL/UAV. Then, with some rearranging, we arrive at the equation for the relative dual quaternion. For the rest of this section, the superscript representing the expressed frame will be dropped unless otherwise noted,

$$\mathbf{q}_{S/B} = \mathbf{q}^*_{B/I} \otimes \mathbf{q}_{S/I}. \tag{3.2}$$

The relative dual quaternion kinematics are derived as follows,

$$\dot{\mathbf{q}}_{S/B} = \frac{d}{dt}\left(\mathbf{q}_{S/B}\right) = \frac{d}{dt}\left(\mathbf{q}^*_{B/I} \otimes \mathbf{q}_{S/I}\right) \tag{3.3}$$

Then, by way of the product rule, we have

$$\dot{\mathbf{q}}_{S/B} = \frac{1}{2}\left(\mathbf{q}_{S/B} \otimes \omega_{S/I} - \omega_{B/I} \otimes \mathbf{q}_{S/B}\right) \tag{3.4}$$

At this point it is useful to define the relative dual-velocity vector between the two rigid bodies,

$$\begin{aligned} \omega_{S/B} &= \boldsymbol{\omega}_{S/I} - R^T(\mathbf{q}_{S/B})\boldsymbol{\omega}_{B/I} \\ &= \boldsymbol{\omega}_{S/I} - \mathbf{q}^*_{S/B} \otimes \boldsymbol{\omega}_{B/I} \otimes \mathbf{q}_{S/B}. \end{aligned} \tag{3.5}$$

This definition can then be rearranged in terms of $\omega_s$, and inserted into the equation to yield

$$\begin{aligned} \dot{\mathbf{q}}_{S/B} &= \frac{1}{2}\mathbf{q}_{S/B} \otimes \boldsymbol{\omega}_{S/B} \\ &= \frac{1}{2}R(\mathbf{q}_{S/B})\boldsymbol{\omega}_{s/b}. \end{aligned} \tag{3.6}$$

### *3.2 Error Kinematics*

Now that we have the relative dual quaternion kinematics model we wish to deal with the dual relative error quaternion as it represents the difference between our desired pose and our estimated pose. We run into the issue that the additive approach destroys the norm constraint [17].

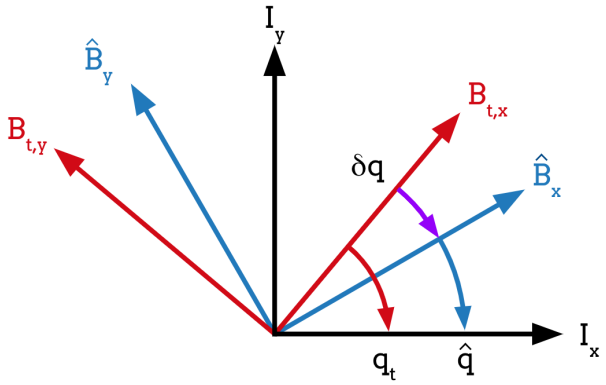


Figure 3.3: Composition of rotations from the nominal quaternion to the estimated quaternion

The solution is, instead, to use the multiplicative error quaternion, which can be seen easily from 3.3, and is written as

$$\delta\mathbf{q} = \hat{\mathbf{q}}^* \otimes \mathbf{q} \tag{3.7}$$

where $\delta\mathbf{q}$ is the dual error quaternion vector, $\hat{q}$ is the estimated quaternion vector, and $\mathbf{q}$ is the "true", or nominal dual quaternion. The dual error quaternion kinematics are derived in the same manner as the relative dual quaternion,

$$\dot{\delta\mathbf{q}} = \frac{d}{dt}\left(\delta\mathbf{q}\right) = \frac{d}{dt}\left(\hat{\mathbf{q}}^{*} \otimes \mathbf{q}_t\right). \tag{3.8}$$

Then, by way of the product rule, we have

$$\dot{\delta\mathbf{q}} = \frac{1}{2}\left(\delta\mathbf{q} \otimes \boldsymbol{\omega}_t - \hat{\boldsymbol{\omega}} \otimes \delta\mathbf{q}\right). \tag{3.9}$$

We now define the error dual-velocity vector

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega}_t - \hat{\boldsymbol{\omega}}. \tag{3.10}$$

By rearranging the dual velocity terms, we can perform the following simplification,

$$\begin{aligned}
\dot{\delta\mathbf{q}} &= \frac{1}{2}\left(\delta\mathbf{q} \otimes \left(\hat{\boldsymbol{\omega}} - \delta\boldsymbol{\omega}\right) - \hat{\boldsymbol{\omega}} \otimes \delta\mathbf{q}\right) \\
&= \frac{1}{2}\left(\delta\mathbf{q} \otimes \hat{\boldsymbol{\omega}} - \hat{\boldsymbol{\omega}} \otimes \delta\mathbf{q}\right) - \frac{1}{2}\delta\mathbf{q} \otimes \delta\boldsymbol{\omega} \\
&= \begin{bmatrix} [\hat{\boldsymbol{\omega}}^{\times}]\delta\mathbf{q}_v \\ 0 \end{bmatrix} + \frac{1}{2}\delta\mathbf{q} \otimes \delta\boldsymbol{\omega}.
\end{aligned} \tag{3.11}$$

This expression represents the exact kinematic relationship for the error rate. In fact the expression has required zero linearizations up until this point. The term on the right is nonlinear

$$\frac{1}{2}\delta\mathbf{q} \otimes \delta\boldsymbol{\omega} \approx \frac{1}{2}\delta\boldsymbol{\omega} + \mathcal{O}(|\delta\boldsymbol{\omega}||\delta\mathbf{q}|). \tag{3.12}$$

By ignoring the higher order term, we are left with the linear error state equation

$$\dot{\delta\mathbf{q}} = \begin{bmatrix} [\hat{\boldsymbol{\omega}}^{\times}]\delta\mathbf{q}_v \\ 0 \end{bmatrix} + \frac{1}{2}\delta\boldsymbol{\omega} \tag{3.13}$$

$$\begin{aligned}
\dot{\delta\mathbf{q}}_v &= [\hat{\boldsymbol{\omega}}^{\times}]\delta\mathbf{q}_v + \frac{1}{2}\delta\boldsymbol{\omega} \\
\delta q_s &= 0.
\end{aligned} \tag{3.14}$$

With our state equation ascertained, we need to be able to calculate the Jacobian for the EKF. Luckily enough, the form we currently have makes it very easy to calculate the partial derivatives,

$$\frac{\partial f}{\partial \delta \mathbf{q}} = \frac{\partial}{\partial \delta \mathbf{q}} \left( \begin{bmatrix} [\hat{\boldsymbol{\omega}}^{\times}] \delta \mathbf{q}_v \\ 0 \end{bmatrix} + \frac{1}{2} \cancelto{0}{\delta \boldsymbol{\omega}} \right) \tag{3.15}$$

$$F = [\hat{\boldsymbol{\omega}}^{\times}]_R = \mathbb{R}^{6 \times 6}.$$

## *3.3 Measurement Model*

We must now develop a measurement model that will allow us to inform the estimator of the position and attitude of the ships deck. An ideal sensor would weigh nothing, cost nothing, be extremely accurate, precise, and work in all conditions. Unfortunately, such a device does not exist, and other methods must be selected to perform the duties required.

While vision systems appear attractive, they tend to have a significant amount of noise associated with their measurements. Additional drawbacks have been noted when it comes to extreme lighting conditions. Given that they are, in the general case, passive sensing devices, they must work with the light levels that are available to them. Instances where this may pose a problem include bright sunny days, or near to total dark conditions, where alternative lighting means are required.

*Lidar Sensor*

A lidar sensor, short for light-detection-and-ranging, delivers where vision can not. With their active light source, they are not limited by over exposure, or dark conditions; providing an ability to detect the position of objects under a multitude of environmental conditions.


Figure 3.4: Velodyne Puck LITE, Range: 100m, Range Accuracy: Up to ± 3cm (Typical), Horizontal FOV: 360°, Vertical FOV: 30°, Refresh Rate: 5-20Hz, Credit: Velodyne Lidar, Inc., www.velodynelidar.com.

A sensor like the one shown in Fig.3.4, was what we had in mind when developing this model. It is for all practical purposes a highly accurate time of flight sensor. It operates by sending out thousands of pulses of light, results in millions points with the possibility of being reflected back, every second.

As the lidar sensor would be located on VTOL aircraft, position information is recorded in the VTOL reference frame. Rather than scan the entirety of the ship, fixed beacons would be located on the ship deck at known positions. Varying geometry and reflectively of the beacons would enable them to be fiducial markers [9] that can be easily identified by an algorithm would be used to perform feature detection. With the individual beacons singled out, their specific positions can be determined. We are then left with how to incorporate the position measurements of the beacons into our pre-existing dual quaternion framework. To do this we will make use of Plücker coordinates as a means of utilizing our 8-element dual quaternion.

*Plücker Coordinates*

If a line goes through the origin, it can be specified by the use of a single vector, but when the line does not traverse the origin, additional means are required. A unit dual vector represented in 3-space known as a Plücker coordinates, or Plücker line are a line with a unit direction vector that passes through a point. Given a to vector valued points,

$$P_1 = [x_1\ y_1\ z_1]^T \qquad \text{and} \qquad P_2 = [x_2\ y_2\ z_2]^T \tag{3.16}$$

we can construct a direction line by subtracting one point from the other,

$$l_{\overrightarrow{12}} = P_1 - P_2 = \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ z_1 - z_2 \end{bmatrix} \tag{3.17}$$

Taking the cross product of one of the points and the direction vector yields the line moment. which is the last element needed to define the Plücker coordinate as,

$$\tilde{l} = \begin{bmatrix} l_{\overrightarrow{12}} \\ 0 \\ P_1 \times l_{\overrightarrow{12}} \\ 0 \end{bmatrix} \tag{3.18}$$

### *3.3.1 Lidar Based Pose Measurement Model*

The mathematical tools are now in place to be able to construct the lidar measurement model. A general layout of the beacons on the landing platform can be seen in Fig.3.5. Three beacons exist on the platform, with beacon A at $\mathcal{P}_A$, beacon B at $\mathcal{P}_B$, and beacon C at $\mathcal{P}_C$, with coordinates,

$$\mathcal{P}_A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix}, \qquad \mathcal{P}_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}, \qquad \mathcal{P}_C = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \tag{3.19}$$

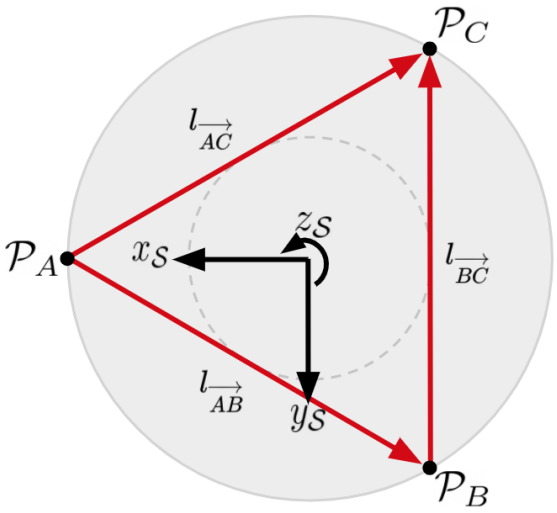


Figure 3.5: General layout of landing beacons located circumscribed circle of an equilateral triangle.

We can then construct the direction vectors,

$$l_{\overrightarrow{AB}} = \mathcal{P}_B - \mathcal{P}_A, \qquad l_{\overrightarrow{AC}} = \mathcal{P}_C - \mathcal{P}_A, \qquad l_{\overrightarrow{BC}} = \mathcal{P}_C - \mathcal{P}_B. \tag{3.20}$$

Then we can construct out dual vectors after generating the moment vectors,

$$L^S_{AB} = \begin{bmatrix} l_{\overrightarrow{AB}} \\ 0 \\ \mathcal{P}_A \times l_{\overrightarrow{AB}} \\ 0 \end{bmatrix}, \qquad L^S_{AC} = \begin{bmatrix} l_{\overrightarrow{AC}} \\ 0 \\ \mathcal{P}_A \times l_{\overrightarrow{AC}} \\ 0 \end{bmatrix}, \qquad L^S_{BC} = \begin{bmatrix} l_{\overrightarrow{BC}} \\ 0 \\ \mathcal{P}_C \times l_{\overrightarrow{BC}} \\ 0 \end{bmatrix}. \tag{3.21}$$

Lastly, we can concatenate the S-frame dual vectors

$$L^S = \begin{bmatrix} L^{S^T}_{AB} & L^{S^T}_{AC} & L^{S^T}_{BC} \end{bmatrix}^T \in \mathbb{R}^{8\times 3}. \tag{3.22}$$

For simplicity sake, I will remove the beacon identifiers, relabeling the measurement vector as

$$L^S = \begin{bmatrix} L^{S^T}_1 & L^{S^T}_2 & L^{S^T}_3 \end{bmatrix}^T. \tag{3.23}$$

With a vector of observations available, we need to determine the attitude matrix that transforms the measurements into the relative frame. The dual vector lidar measurement model is derived by assuming that three dual vector observations are available at time $t_k$. We can then concatenate them to form the following measurement vector

$$y_k = \begin{bmatrix} T(\mathbf{q}_{S/B})\mathbf{r}_1^S \\ T(\mathbf{q}_{S/B})\mathbf{r}_2^S \\ T(\mathbf{q}_{S/B})\mathbf{r}_3^S \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \equiv \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.24}$$

$$R = diag\left[\sigma_1^2\mathbf{I}_{3\times3}\ \sigma_2^2\mathbf{I}_{3\times3}\ \sigma_3^2\mathbf{I}_{3\times3}\right] \tag{3.25}$$

such that the diagonal elements are made up of the standard deviations of the measurement error. In order to be able to use this in the extended Kalman filter, we will need to be able to take a partial derivative of it. When perturbed

$$\begin{aligned} h(\hat{x}) &= T(\hat{\mathbf{q}}_{S/B} \otimes \delta\mathbf{q}_{S/B})\mathbf{r}^S \\ &= T(\hat{\mathbf{q}}_{S/B})T(\delta\mathbf{q}_{S/B})\mathbf{r}^S \end{aligned} \tag{3.26}$$

where

$$T(\hat{\mathbf{q}}_{S/B}) = [\hat{\mathbf{q}}_{S/B}]_L[\hat{\mathbf{q}}^*_{S/B}]_R \tag{3.27}$$

$$T(\delta\mathbf{q}_{S/B}) = [\delta\mathbf{q}^*_{S/B}]_L[\delta\mathbf{q}^*_{S/B}]_R. \tag{3.28}$$

We approximate the second term as

$$T(\delta\mathbf{q}_{S/B}) \approx \mathbf{I}_{8\times8} + [\delta\mathbf{q}_{S/B}]_L \tag{3.29}$$

then,

$$\begin{aligned} \mathbf{r}^B = y &= \left((\mathbf{I}_{8\times8} + [\delta\mathbf{q}_{S/B}]_L)([\mathbf{q}_{S/B}]_L[\mathbf{q}^*_{S/B}]_R)\right)\mathbf{r}_S \\ &= \left([\mathbf{q}_{S/B}]_L[\mathbf{q}^*_{S/B}]_R\right)\mathbf{r}^S + \left([\delta\mathbf{q}_{S/B}]_L[\mathbf{q}_{S/B}]_L[\mathbf{q}^*_{S/B}]_R\right)\mathbf{r}_S. \end{aligned} \tag{3.30}$$

Now we can differentiate with respect to the error state, $\delta\mathbf{q}_{S/B}$

$$H = \frac{\partial h}{\partial \delta \mathbf{q}} \Rightarrow \frac{\partial}{\partial \delta \mathbf{q}} \left( \cancelto{0}{\left( [\mathbf{q}_{S/B}]_L [\mathbf{q}^*_{S/B}]_R \right)} \mathbf{r}^S \right) + \frac{\partial}{\partial \delta \mathbf{q}} \left( [\delta \mathbf{q}_{S/B}]_L [\mathbf{q}_{S/B}]_L [\mathbf{q}^*_{S/B}]_R \right) \mathbf{r}^S \tag{3.31}$$

We now are left with the right most term, however our state is in the form of a matrix. Since we are dealing with dual quaternions, there is a simple way around this by using the following;

$$\left( \underbrace{[\delta \mathbf{q}_{S/B}]_L}_{matrix} \underbrace{[\mathbf{q}_{S/B}]_L [\mathbf{q}^*_{S/B}]_R}_{vector} \right) \mathbf{r}^S \Rightarrow \delta \mathbf{q}_{S/B} \otimes \underbrace{\left( \mathbf{q}_{S/B} \otimes \mathbf{r}^S \otimes \mathbf{q}^*_{S/B} \right)}_{[^{\mathbf{q}}v]_R} \tag{3.32}$$

What we are left with is a dual quaternion right-side multiplication matrix,

$$[^{\mathbf{q}}v]_R \delta \mathbf{q}_{S/B} \xrightarrow{\frac{\partial}{\partial \delta \mathbf{q}}} H = [^{\mathbf{q}}v]_R \in \mathbb{R}^{8\times 8} \tag{3.33}$$

### *3.4 Extended Kalman Filter*

As the key equations were derived in previous sections, a brief review of the extended Kalman filter will be discussed. The extended Kalman filter (EKF) was developed shortly after dissemination of Kalman's paper. As is the case with most techniques developed for linear systems, they only work within the narrow range with which the function has been approximated and/or linearized. When researchers in NASA's Marshall Space Flight Center were attempting to tackle the mid-course navigation required for astronauts to make it to and from the moon, a fast and reliable means of estimation was needed in order to work with observations made by the astronauts. While the Kalman filter seemed like it would be a perfect fit for the mission, the engineers determined that their system was nonlinear enough that some other method would have to be adopted before being able to us the Kalman Filter. It was shortly thereafter that the extended Kalman filter was created. Able to handle nonlinear systems, the EKF take successive linearization steps. Unfortunately, while this has proven to be wildly successful, the guarantees that were proven in the linear case were no longer assured. This has not stopped it however from being one of the most popular estimators.

### *3.4.1 EKF General Equations*

The main equations of the EKF are reviewed in this section. State equations may be written as

$$\dot{x}(t) = f(x(t), t) + g(x(t), t)w(t) \tag{3.34}$$

where x(t) is the state vector and the process noise w(t), is a Gaussian white noise process that has a mean and covariance function described by

$$\begin{aligned} E[w(t)] &= 0 \\ E[w(t)w^T(t')] &= Q(t)\delta(t - t'). \end{aligned} \tag{3.35}$$

With the term E representing the expectation. Conversely, the initial state mean and covarinace are

$$\begin{aligned} E[x(t_0)] &= x_0 \ (\equiv \hat{x}_0) \\ E[[x(t_0) - x_0][x(t_0) - x_0]^T] &= P_0 \ (\equiv P(t_0)) \end{aligned} \tag{3.36}$$

**Step 1: Time Update (Propagate)**

Given the state vector's initial conditions, as well as the the state error covariance matrix initial conditions, the minimum variance estimate of the state vector at a future time, $t$, is given in the absence of measurements by the conditional expectation

$$\hat{x}(t) = E[x(t)|\hat{x}(t_0) = x_0]. \tag{3.37}$$

This predicted estimate satisfies the differential equation

$$\frac{d}{dt}\hat{x}(t) = E[f(x(t), t)] \equiv \hat{f}(x(t), t) \tag{3.38}$$

which we write approximately as

$$\frac{d}{dt}\hat{x}(t) = \Phi(t, \hat{x}(t_0), t_0). \tag{3.39}$$

After formal integration,

$$\hat{x}(t) = \Phi(t, \hat{x}(t_0), t_0). \tag{3.40}$$

The state error vector and covariance matrix are defined by

$$\Delta x(t) = x(t) - \hat{x}(t) \tag{3.41}$$

$$P(t) = E[\Delta x(t) \Delta x^T(t)]. \tag{3.42}$$

Then, retaining the first-order terms, while neglecting the higher order ones, for both the state error vector and the process noise, the state error vector satisfies the differential equation

$$\frac{d}{dt}\Delta x(t) = F(t)\Delta x(t) + G(t)w(t) \tag{3.43}$$

where

$$F(t) \equiv \frac{\partial}{\partial x} f(x,t)\bigg|_{\hat{x}(t)} \tag{3.44}$$

$$G(t) \equiv g(\hat{x}(t), t). \tag{3.45}$$

Formal integration gives

$$\Delta = \Phi(t, t_0)\Delta x(t_0) + \int_{t_0}^{t} \Phi(t, t')G(t')w(t')dt' \tag{3.46}$$

where $\Phi(t, t_0)$ is the transition matrix, which satisfies

$$\frac{\partial}{\partial t}\Phi(t, t_0) = F(t)\Phi(t, t_0) \tag{3.47}$$

$$\Phi(t_0, t_0) = I \tag{3.48}$$

Note that for nonlinear systems, $\Phi(t, t_0)$, also depends implicitly on $\hat{x}(t_0)$, which for notational convenience will be suppressed. The predicted covariance matrix satisfies the Riccati equation,

$$\frac{d}{dt}P(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) \tag{3.49}$$

which may be integrated to give

$$P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \int_{t_0}^{t} \Phi(t, t')G(t')Q(t')G^T(t')\Phi^T(t, t')dt' \tag{3.50}$$

In more compact notation, $\hat{x}_k^-$ and $P_k^-$ denote the predicted values of the state vector and the state covariance matrix at time $t_k$, and $\hat{x}_k^+$ and $P_k^-$ denote the same quantities immediately following a measurement at time $t_k$. Thus the obvious notation

$$\hat{x}_{k+1}^- = \Phi(t_{k+1}, \hat{x}_k^+, t_k) \tag{3.51}$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + N_k \tag{3.52}$$

**Step 2: Measurement Update (Correct)**

The measurement vector at time $t_k$ is related to the state vector by

$$z_k = h(x_k) + v_k \tag{3.53}$$

where $v_k$ is the measurement noise that follows a discrete Gaussian white-noise process

$$E[v_k] = 0 \tag{3.54}$$

$$E[v_k v_k^T] = R_k \delta_{kk'}. \tag{3.55}$$

The minimum variance estimate of $x_k$ immediately following the measurement is given by

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left( z_k - h(\hat{x}_k^-) \right) \tag{3.56}$$

where the Kalman gain matrix is given by

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{3.57}$$

and the measurement sensitivity matrix is given by

$$H_k = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_k(-)} \tag{3.58}$$

the covariance matrix immediately following the measurement is given by

$$\begin{aligned} P_k^+ &= (I - KkH_k)P_k^- \\ &= (I - KkH_k)P_k(-)(I - K_k H_k)^T + K_k R_k K_k^T. \end{aligned} \tag{3.59}$$

## *3.5 Unscented Kalman Filter*

We will now briefly review the concepts surrounding the unscented Kalman filter, or what is sometimes referred to as a sigma point filter. The creator of the filter, Jeffery Uhlmann, began experimenting with forms of sigma point filters that make use of a method whereby the functions were sampled, then propagated to determine the mean and covariance values. Where his creation was different, was in how the sampling was performed. Through what is called an unscented transform, the mean and covariance values are propagated. This represents a significant departure from the Kalman family of estimators, in that the Riccati equation, which is used to propagate the error is supplanted by the new method. Rather than approximate the nonlinear function through an inaccurate method like linearization it approximates the the distribution of the state random variable. This can happen through the use of the sigma points. He was able to ascertain the mean and covariance of the random variable such that once propagated through the nonlinear function they would be able to map the distribution to a standard Gaussian distribution. This process allowing those points to distributed according to the specifics of the function with which it is being passed through. This meant that rates of convergence should increase, thus driving down the error bounds. This performance does come at a cost however. Due to the generation and propagation of the sigma points, there is a higher computational burden.

### *3.5.1 UKF General Equations*

We starter by considering a state variable $x$, that follows a normal distribution with mean $x_m$ and covariance, $P_x$,

$$x \sim N\left(x_m, P_\chi\right). \tag{3.60}$$

We next define the sigma points $\chi_i$

$$\begin{aligned} \chi_1 &= x \\ \chi_{i+1} &= x + u_i \qquad \text{for } i = 1...n \\ \chi_{i+n+1} &= x - u_i \qquad \text{for } i = 1...n \end{aligned} \tag{3.61}$$

and the weights $W_i$

$$\begin{aligned} W_1 &= \frac{\kappa}{n+k} \\ W_{i+1} &= \frac{1}{2(n+k)} \qquad \text{for } i = 1...n \\ W_{i+n+1} &= \frac{1}{2(n+k)} \qquad \text{for } i = 1...n \end{aligned} \tag{3.62}$$

where $u_i$ is a row vector of matrix U, satisfying

$$U^T U = (n+\kappa)P_\chi \tag{3.63}$$

and $\kappa$ is an arbitrary constant.Then the mean and covariance of a function are given by

$$y = f(x) \tag{3.64}$$

that are constructed as the summations

$$\begin{aligned} y_m &= \Sigma_{i=1}^{2n+1} W_i f(\chi_i), \\ P_y &= \Sigma_{i=1}^{2n+1} W_i \{f(\chi_i - x_m\}\{f(\chi_i) - x_m\}^T. \end{aligned} \tag{3.65}$$

The procedure above is called the unscented transform. The sigma points of the unscented transform correspond to the the standard deviation about the mean of the estimate. The weights are constants determining the weighting of each point when computing the mean and covariance. Next, the error dual quaternion based UKF can be derived also in the following step.

**Step 1: Time Update (Propagate)**

$$\begin{aligned}
\left(\chi_i\left(\delta\hat{\mathbf{q}}_{k-1}\right), W_i\right) &\leftarrow \left(\delta\hat{\mathbf{q}}_{k-1}, P_{k-1}\right) \\
\chi_i\left(\hat{\mathbf{q}}_{k-1}\right) &\leftarrow \chi_i\left(\delta\hat{\mathbf{q}}_{k-1}\right) \\
\left(\hat{\mathbf{q}}_k^-, P_k^-\right) &= UT\left(f\left(\chi_i\left(\delta\hat{\mathbf{q}}_{k-1}\right)\right), W_i, Q\right) \\
\delta\hat{\mathbf{q}}_{k-1}^- &= \hat{\mathbf{q}}_{k-1}^- \otimes f\left(\chi_i\left(\hat{\mathbf{q}}_{k-1}\right)\right)
\end{aligned} \tag{3.66}$$

**Step 2: Measurement Update (Correct)**

$$\begin{aligned}
\left(\chi_{i+1}\left(\delta\hat{\mathbf{q}}_k^-\right), W_{i+1}\right) &\leftarrow \left(\delta\hat{\mathbf{q}}^-, P_k^-\right) \\
\left(\hat{\mathbf{z}}_k, P_z\right) &= UT\left(h\left(\chi_{i+1}\left(\hat{\mathbf{q}}_k^-\right)\right), W_{i+1}, R\right) \\
P_{xz} &= \Sigma_{i=1}^{2n+1} W_i\left\{f(\chi_i) - \hat{x}_k^-\right\}\left\{h(\chi_{i+1}) - \hat{z}_k\right\} \\
K_k &= P_{xz}/P_z \\
\delta\hat{\mathbf{q}}_k &= K_k\left(z_k - \hat{z}_k\right) \\
\hat{\mathbf{q}}_k &= \hat{\mathbf{q}}_k^- \otimes \delta\hat{q}_k \\
P_k &= P_k^- - K_k P_z K_k^T
\end{aligned} \tag{3.67}$$

The DQ-UKF is able to be implement without the need for a Jacobian matrix. So, previous measurement formulation results presented in the next section.

# Chapter 4

# SIMULATION AND RESULTS

Validation of the two nonlinear estimators derived in the previous chapter are discussed in this section. We review the synthetic measurements used to test the estimators' abilities, and their development. Full system simulation incorporating measurements, measurement models, and the estimators that use them are performed with results from each method being reviewed.

## *4.1 Synthetic Measurements*

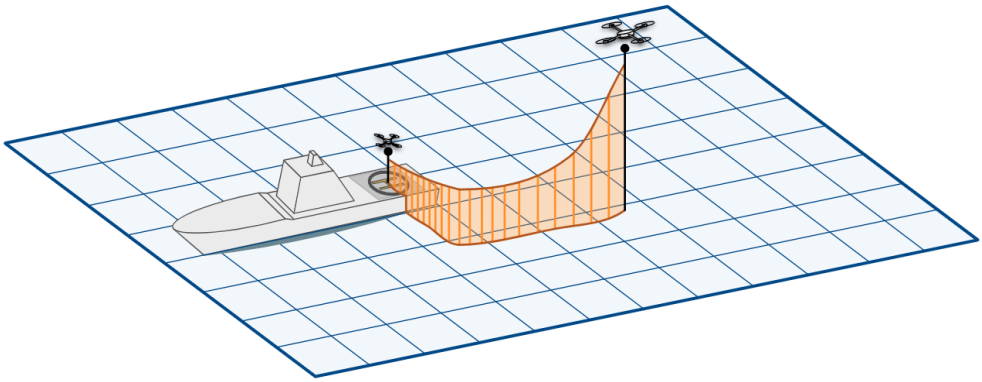

Figure 4.1: UAV short-final, and station keeping phases of landing on a ship

Two distinct phases exist, as shown in Fig. 4.1, when approaching to land on a ship. Short final is marked by the aircraft's approach to the vessel. Once maneuvered to its touchdown point, station keeping begins. The aircraft maintains position over the ship's deck, waiting for the ability to descend, such that crash risk is minimized. The ship is not underway in this scenario; it is actively

station keeping itself. The motion of the ship is then only influenced by environmental factors acting on the hull. Likewise, the same is true for the VTOL aircraft.

In an effort to reduce the scope, and demands, of the simulation, it was decided to not develop full 6DOF models of the two vehicles. Instead, a decoupled point mass motion approach was used to emulate, and not replicate, the motions. This method reduced simulation complexity, while retaining characteristics such as ocean wave height and frequency, and atmospheric wind gust and turbulence, that give reason for the difficulty associated with such a scenario.

*Ship Motion*

External, environmental, forces and moments that lead to ships' motions are generally created through wind, waves, and currents. As the ship is maintaining station, current interaction can be disregarded. As for wind, this is assumed for our problem to have minimal impact on a large marine vessel. Primary motion of a ship then comes from the interaction between the ship and waves. This ultimately results in bounded, oscillatory motion [24]; importantly, this can be easily described through the superposition principal

$$\sum_{i=1}^{n} A_i \sin(\omega_i t + \phi_i)(-1)^{c_i} \tag{4.1}$$

where A, $\omega$, $\phi$, and $(-1)^c$ represent the amplitude, frequency, phase, and a sign change term respectively. From the World Meteorological Organization's sea state scale for wave height [1], and Munk's classification of wind waves by period [21], ranges were generated, and values were randomly chosen from within them

$$[A_{min}, A_{max}] \Rightarrow A_i \in \mathbb{R} : A_{min} \leq A_i \leq A_{max} \tag{4.2}$$

$$[\omega_{min}, \omega_{max}] \Rightarrow \omega_i \in \mathbb{R} : \omega_{min} \leq \omega_i \leq \omega_{max} \tag{4.3}$$

$$[\phi_{min}, \phi_{max}] \Rightarrow \phi_i \in \mathbb{R} : \phi_{min} \leq \phi_i \leq \phi_{max} \tag{4.4}$$

$$(-1)^{c_i} = \begin{cases} 1, & \text{if } c_i = 2m \\ -1, & \text{if } c_i = 2m+1 \end{cases} \quad \forall\, m \in \mathbb{Z} \tag{4.5}$$

The phase parameter was randomly selected from a range of 0 to $\pi$, and an upper bound of summation was set at 50. Results showing a sub-sampling of the component wave forms as well as the final summation can be seen in Fig. 4.2. The pseudo random waveform represents the vertical motion of the ship deck that would be induced by the vessel encountering a series of wind waves.

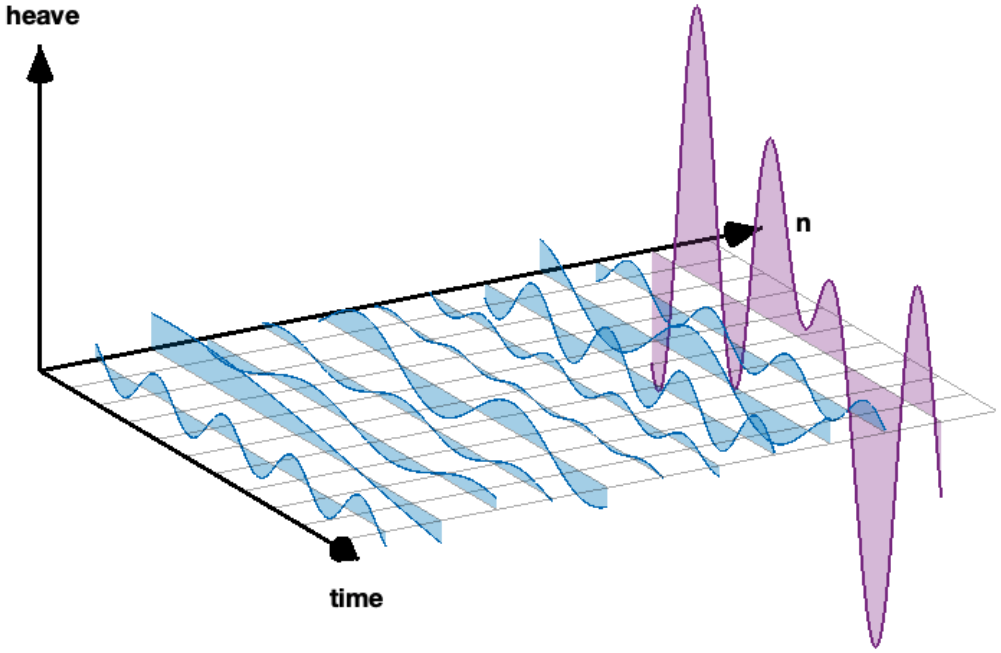


Figure 4.2: Superposition of randomized sinusoidal terms.

This method was applied to the remaining motions, with fluctuations in magnitude and frequency that come from the decoupled dynamics documented in marine vessels [12]. Specifically the heave and rolling components will see the largest variations, followed by pitch. The remaining degrees of freedom, while not motionless, will be subdued as they are not as easily excited by the wave interactions. By using the center of the landing platform as the point about which the motions

transpire, the translations and rotations were then generated, the results of which can be seen in Fig. 4.3 and Fig. 4.4.

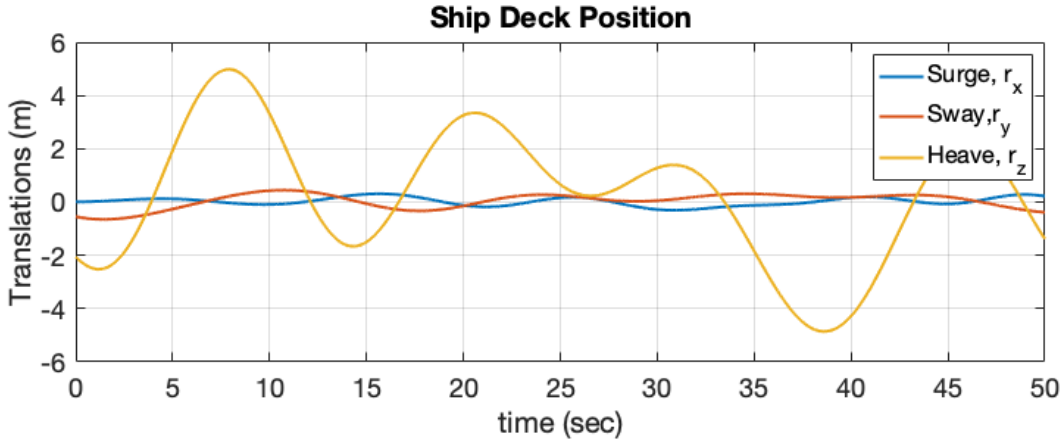


Figure 4.3: Ship Deck Position Trajectories

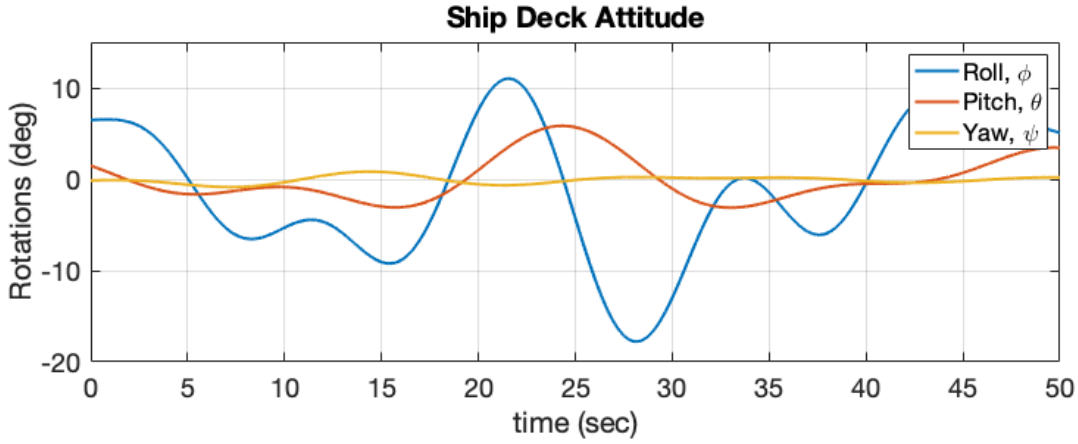


Figure 4.4: Ship Deck Attitude Trajectories (Euler angles as reference only)

The overall simplification comes from the idea that the platform will ‘ride’ the various motion curves.

### *4.1.1 VTOL Motion*

While not interacting with the ocean per se, the aircraft must contend with its own randomized disturbances. These documented interactions include complex wind-wake interaction that come from wind blowing over the ship's hull and superstructure [31], as well as the more general case of surface level wind gusts. For the VTOL motion simulation, the same superposition method was chosen. External disturbances such as wind and ship-VTOL interaction are the primary drivers in producing of the oscillations of the aircraft. Once again, the same superposition method was chosen to supplant a more complex model, as the ability to estimate is decoupled from the platform on which the system would be running.

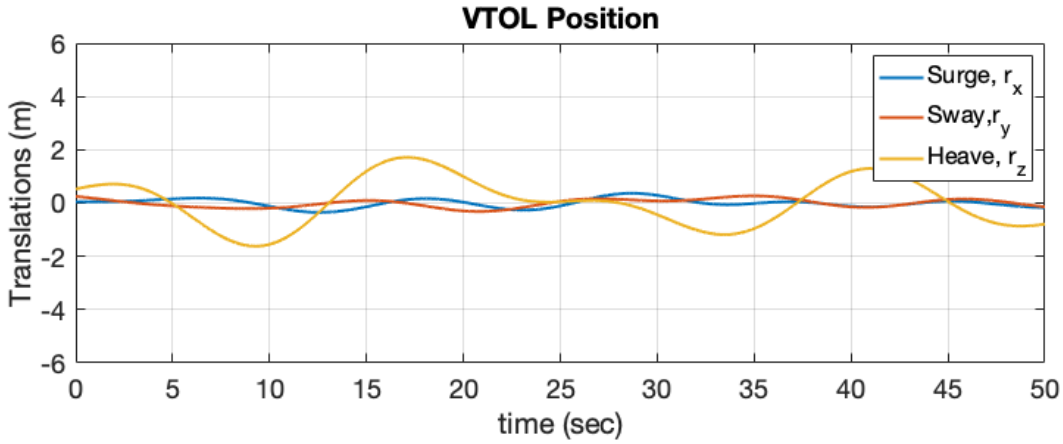


Figure 4.5: VTOL Position Trajectories

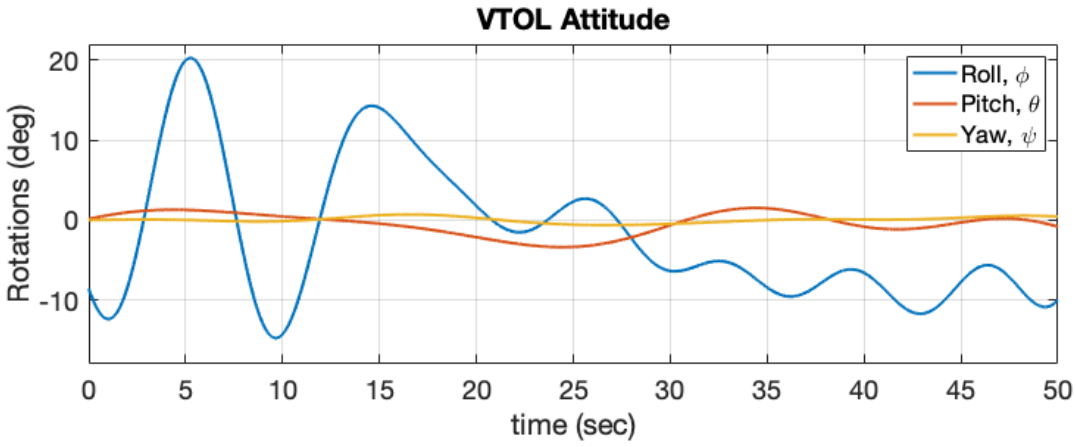


Figure 4.6: VTOL Attitude Trajectories (Euler angles as reference only)

While a shortcoming associated with this approach is the decoupling of rotational and translational positions, this was deemed acceptable, as the combination of movements are ultimately bounded by some regulating and tracking controller that would maintain position in the face of these disturbances. The generated trajectories in Fig. 4.5 and Fig. 4.6 were created under the assumption that disturbances from wind acting on the vehicle would lead to slight oscillatory motion of the vehicle.

## *4.2 Results*

To validate the methods considered in this thesis, the VTOL aircraft was set in a hovering position approximately 20 meters above the moving landing platform. Given the amount of vertical movement experienced by both vehicles, this value allowed for reasonable separation. This exercise's aim was to evaluate the steady state performance of the estimator while in the last phase of landing. The simulation was set to run for 50 seconds so as to collect ample data on the estimation of the relative deck motion while encountering disturbances.

*Initial Conditions*

The initial conditions for the full state of the estimator were determined by inspecting the first measurements from the system. These were then provided to the estimator as the initial state.

$$\hat{\mathbf{q}}_0 = [0.05 \;\; 0.01 \;\; 0 \;\; 0.998 \;\; -0.024 \;\; -0.397 \;\; -11.295 \;\; 0.0]^T \tag{4.6}$$

$$\delta\hat{\mathbf{q}}_0 = [0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0]^T \tag{4.7}$$

With the state error covariance matrix being initialized with,

$$P_0 = \mathbf{I}_{6\times 6} \tag{4.8}$$

The process and measurement noise covariance matrices were

$$Q = diag([0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.01]) \tag{4.9}$$

$$R = diag([0.001,\ 0.01,\ 0.01,\ 0.01,\ 0.01,\ 0.01]) \tag{4.10}$$

*EKF Performance*

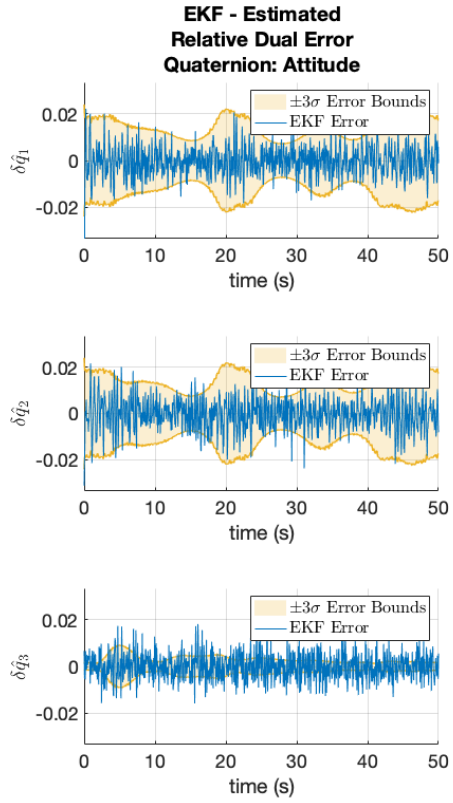


Figure 4.7: EKF Error State: Attitude

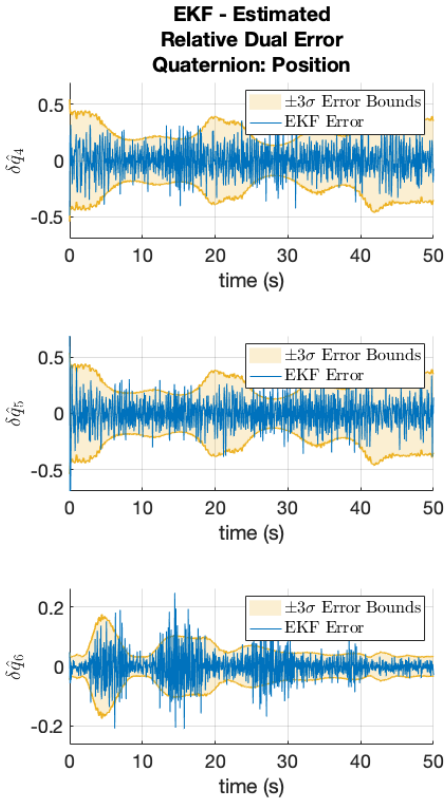


Figure 4.8: EKF Error State: Position

In Fig. 4.7 and Fig. 4.8, we see the reduced state output from the estimator. The $3\sigma$ bounds that represent three standard deviations have also been included as a measure of the estimators performance. Suffice to say the EKF performed very well with the exception of the third term in the attitude portion, the yaw motion. It can be see that the estimate error is well beyond the bounds provided by the covariance output.

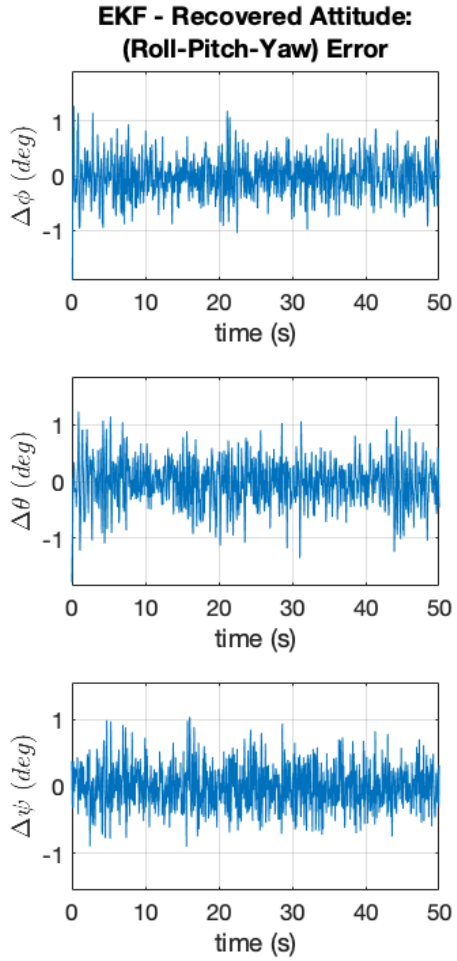


Figure 4.9: EKF Error State: Attitude Recovery

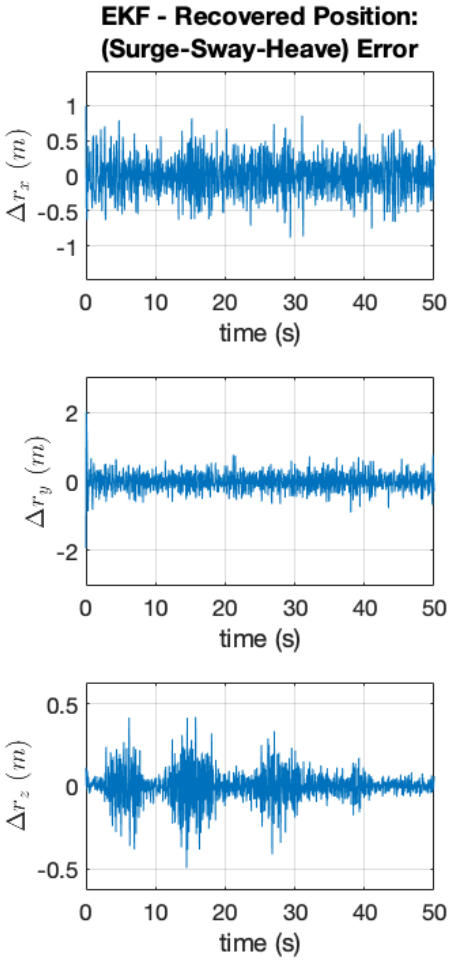


Figure 4.10: EKF Error State: Position Recovery

These nex figures represent the same error as was seen in Fig. 4.7 and Fig. 4.8. The difference however is that this time a transformation of the data has occurred. This was possible since when the state dropped from four states to three, we had made some additional simplifications in that the small angle rule was applied to the values, so rather than being a parameterization of the angles, it was just the angles. With units associated with the values on the plot we can see that the attitude

tracking of Fig. 4.9 was working very well since the values are between $\pm 1$ degree. While the position tracking is also good, within about a unit, we see in Fig. 4.10 the variance between values.

*UKF Performance*

Moving along to the results for the UKF, and we see a similar story. The attitude tracking for the third component of the reduced error state is well outside of its $3\sigma$ bounding. Which is supposed to provide a 99.7% cover, and yet the estimator had a hard time trying to track what would be yaw motion.

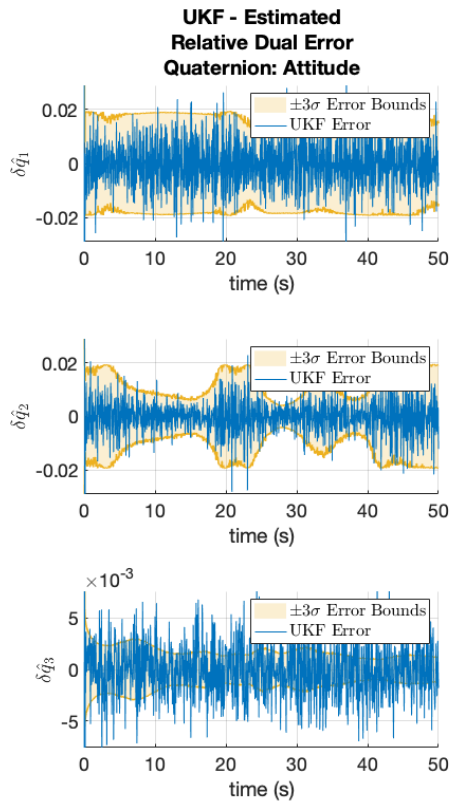


Figure 4.11: UKF Error State: Attitude

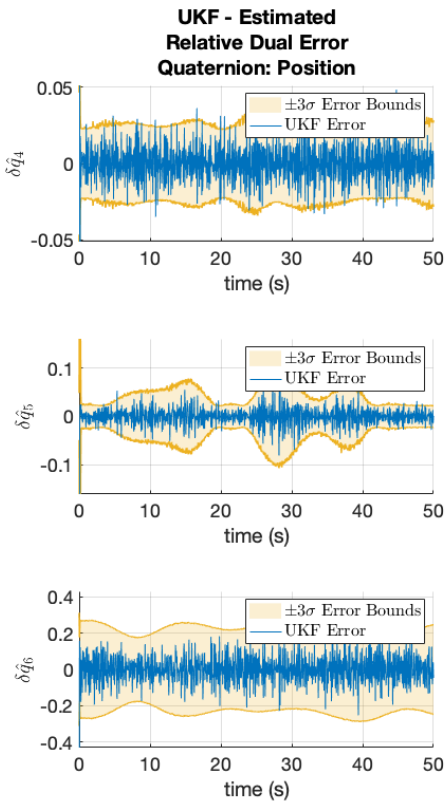


Figure 4.12: UKF Error State: Position

We see similar results for the recovered tracking error in Figs. 4.11 and 4.12. There is about 1 degree of variation on the attitude portion. It is interesting to see how the roll and yaw values appear to be larger than the pitching error.

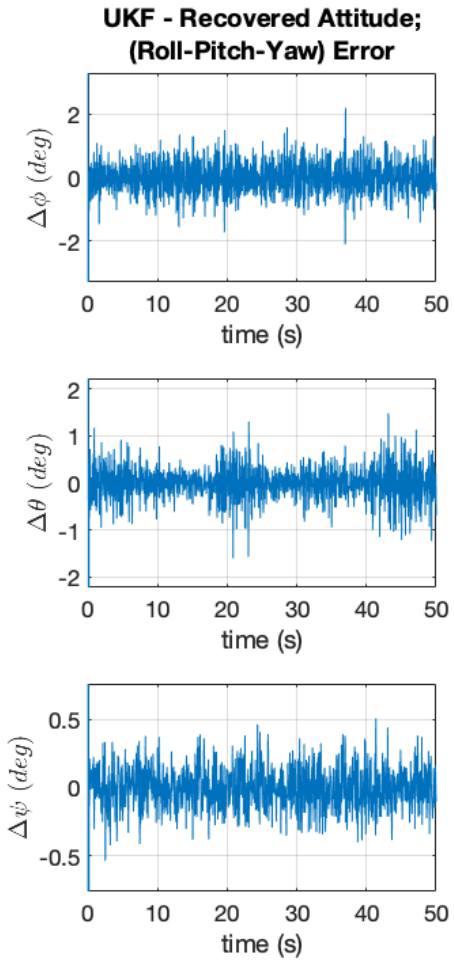


Figure 4.13: UKF Error State: Attitude Recovery

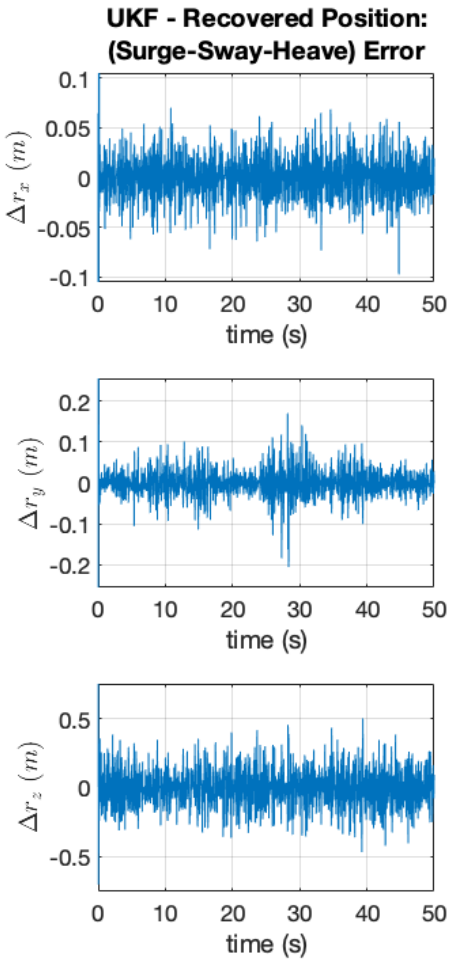


Figure 4.14: UKF Error State: Position Recovery

*Performance Comparison*

The estimators' tracking of the landing deck attitude quaternion relative to the aircraft can be seen in Fig. 4.15, while the estimated position of the deck can be seen in Fig.4.16.

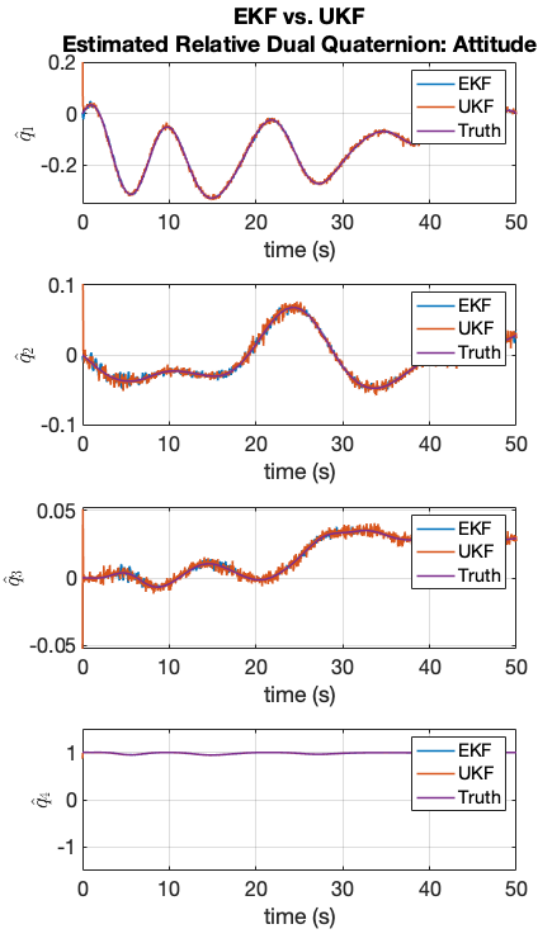


Figure 4.15: Full state attitude estimations for the EKF and the UKF

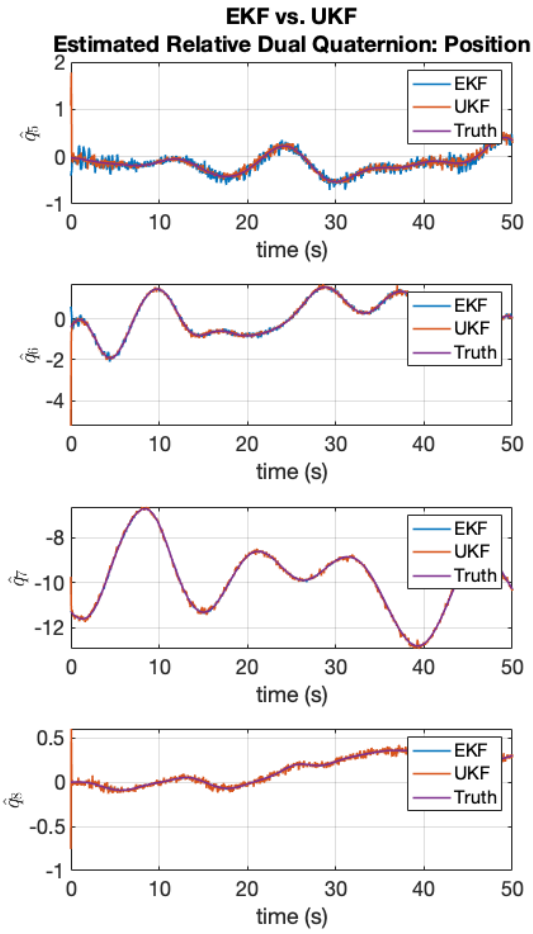


Figure 4.16: Full state position estimations for the EKF and the UKF

In both figures the EKF is shown as the blue line, the UKF as the orange line, and the ground truth value as the purple line. Both estimators are doing a good job at tracking the true value. The

position values in Fig. 4.16, appear to have a bit more noise associated with them, especially for the $\hat{\mathbf{q}}_5$ parameter, where the EKF seems to have a wider band with which it is oscillating between. Shifting focus to the second and third elements in the attitude portion of the dual quaternion in Fig. 4.15, there is a similar band occurring, albeit with the UKF. By reviewing the error quaternion, a better understanding of the differences between the two estimators will be displayed.

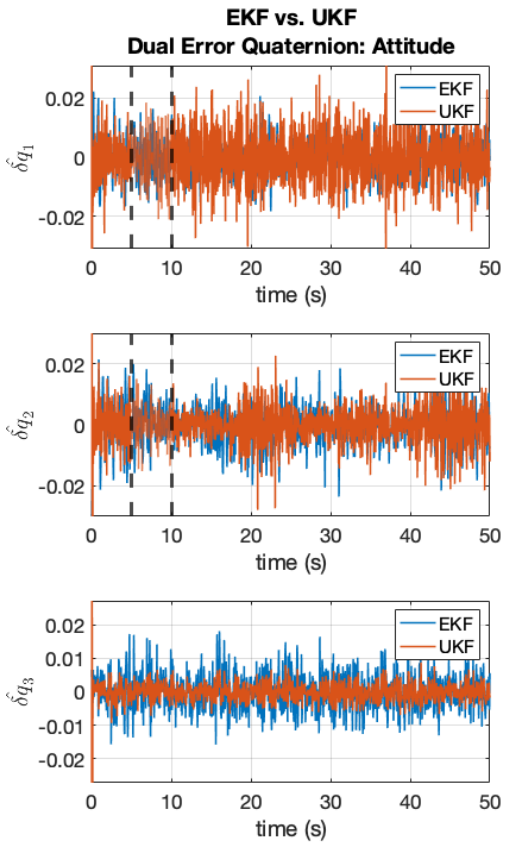


Figure 4.17: Attitude errors for the EKF and the UKF

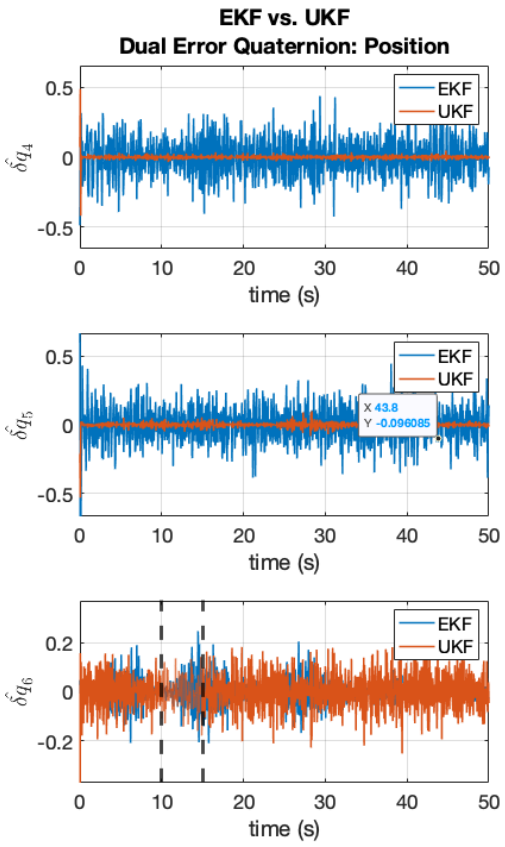


Figure 4.18: Position errors for the EKF and the UKF

From the error quaternion figures it becomes more apparent how well the estimators are performing. Two windows have been added to the first two sub-figures representing $\delta\hat{\mathbf{q}}_1$ and $\delta\hat{\mathbf{q}}_1$ in

Fig. 4.17. These allow us to better see the performance of the EKF, in blue, with respect to the UKF, in orange. From the attitude plot it is seen that there is greater fluctuation in the values for the first two values of the error quaternion for the UKF. These roughly correspond to roll and pitch motions, whereas the yaw term, associated with the $\delta\hat{\mathbf{q}}_3$ term is smaller than the same term for the EKF.

What is more interesting, is the seemingly opposite of this, with respect to the position error elements of Fig. 4.18. The error for the first two terms, representing the two lateral motions, shows the UKF as producing the choicer estimates by a large factor. The same windowing was applied to the third estimate, this allows us to see that the EKF value appears to oscillate, while the UKF value remains consistent.

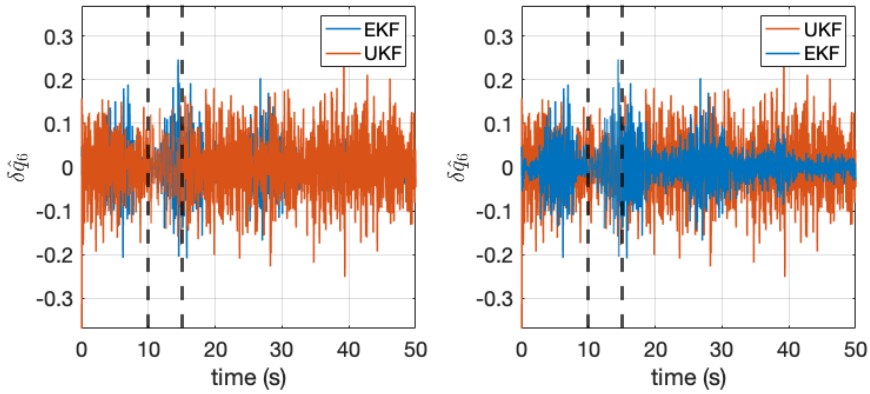


Figure 4.19: Sixth error quaternion element:

We can see from Fig. 4.19 that by flipping which plotting order, a better picture of the differences between the two outputs. It appears that the EKF error quaternion varying significantly, whereas the UKF value is more uniform over the simulation time span.

We get one of the best accounts of what is happening, when we recover the actually attitude and positional coordinate values. Fig. 4.20 and Fig. 4.21 give that to use and we can see associated units that help to relate the values.

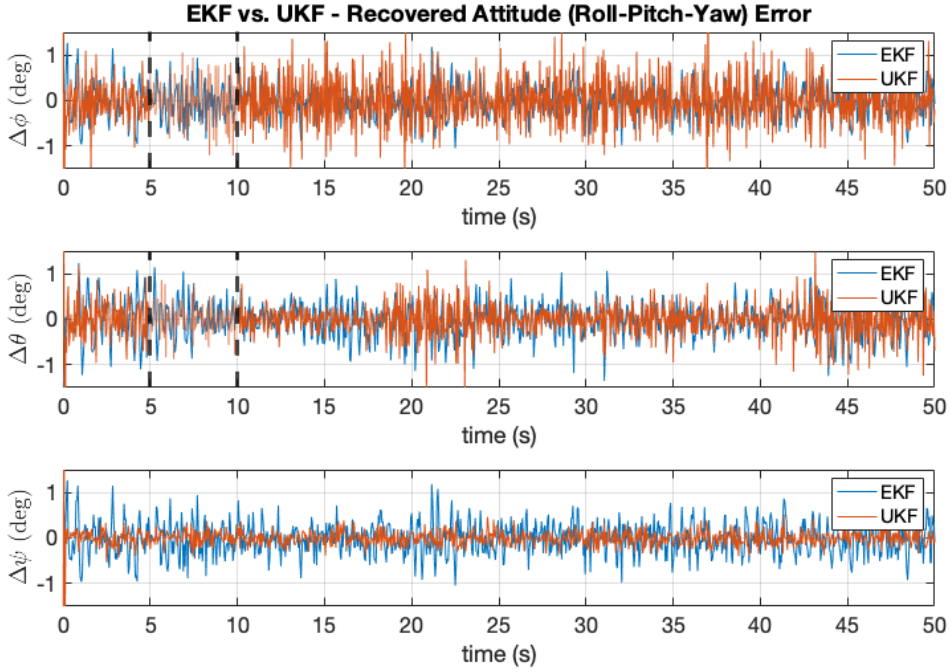


Figure 4.20: Recovered relative position values

It appears that the relative position is best tracked by the UKF. When directly overlaying the information as is being done now, it can be a fairly decisive choice.

For the last comparison between the two estimation processes, we looked at the computational time. Specifically for this project, that entailed measuring the time one estimation cycle was completed. As there was no discernible difference from the perspective of the student running the simulation. From Fig. 4.22 we see just how much faster the EKF is over the UKF. Clearly given that it was calculating 2n+1 sigma points, each cycle time took measureably longer.

The processing time is still respectable, however the simulations were run on a computer utilizing an Intel Core i7. The processing time may grow if an estimator is to run on microprocessor, or perhaps even a micro-controller. I am also intrigued as to what is causing the spikes in run time near the beginning, and end of the simulation run.

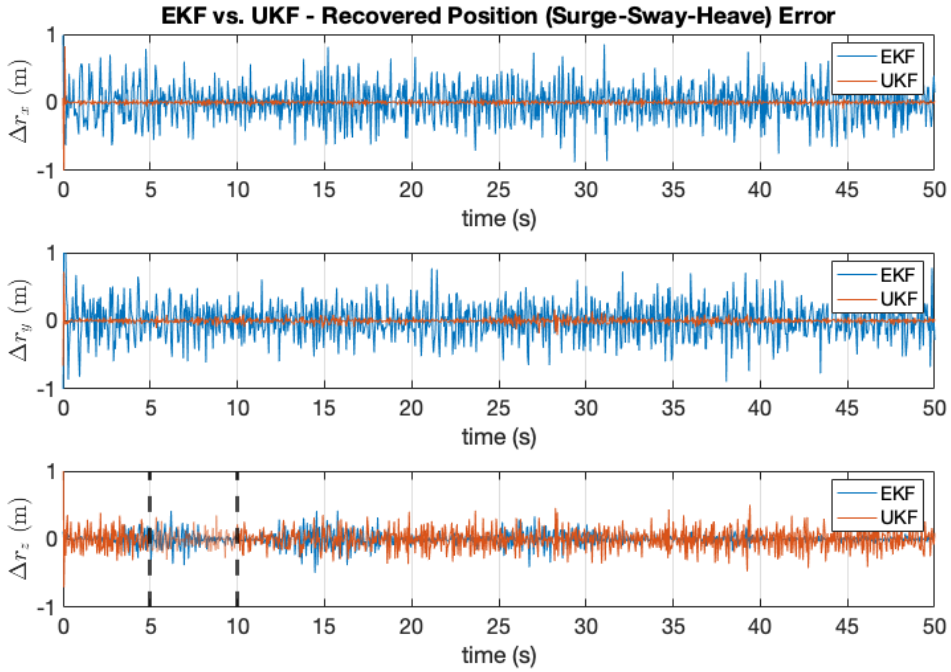


Figure 4.21: Recovered relative attitude values

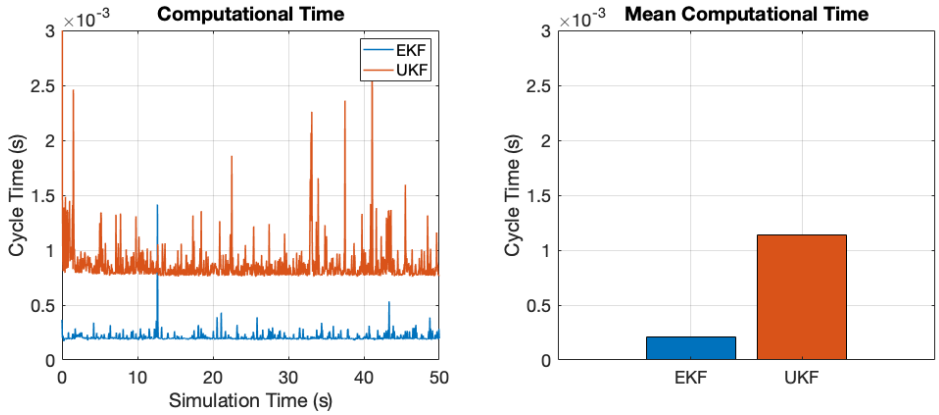


Figure 4.22: Estimator cycle time

# Chapter 5
# CONCLUSION

Inspired by the demanding conditions associated with the landing of a VTOL aircraft on the deck of a ship in rough seas, we have presented two methods to accurately estimate the position and attitude of the ships deck relative to the aircraft. Accurate knowledge of the landing platform can enable autonomous aircraft to expand their capabilities. That same knowledge can also serve to offset some of the workload that current pilots must overcome to safely complete their mission. Regardless of the mechanism by which the aircraft lands, awareness of what is happening is of vital importance. The need for relative pose estimation will grow; whether through expanding current capabilities, or taking on new roles; after all, once the aircraft takes off from the ship, it will eventually need to land whether the seas are calm or not.

## *5.1 Summary of Work*

This thesis successfully addressed three primary areas of interest: adopting dual quaternions for autonomous landing operations, developing and comparing two nonlinear estimators, and developing a simplified means to generate pseudo random motions to model both the ship and the VTOL motion. Through a wealth of successful work developed in the field of spacecraft operations, there was a significant framework of information to pull from in applying dual quaternions to this problem. Their application proved valid, yielding estimators that could hand both position and attitude as a unified state variable.

The nonlinear pose estimators that were created demonstrate feasibility of the method. Given the inability to predict the motion of the ocean, and by default the ship, the system could be deployed with limited knowledge of the dynamics driving the system. Both the EKF and the UKF

ended up using the dual error quaternion as the state, validating that the reduced state showed no drawbacks for this application.

The synthetic measurements that drove the simulation were created using a simplistic approach. Given the unpredictability of the ocean, there are still several characteristics that bound its motion. Elements such as known wave heights and frequencies for given conditions helped lay the ground work for the linear superposition modeling that was done. With the effort and complexity required to model not only a ship's motion but that of a VTOL in turbulent conditions, the uncoupled motion scheme that was created proved to be able to test the estimators appropriately with high levels of confidence with levels of accuracy that were demonstrated, and were found to be very acceptable.

### *5.1.1 Estimation*

While the estimators in this project performed well, further exploration of their limitations would prove beneficial. One limitation that should be overcome, is to incorporate relative velocity states in the estimation process. Modifications to the measurement models may need to occur, but the relative velocities will play an important part with autonomous landing, in that high relative closing speeds will need to be known so as to avoid accidents. Moving away from the algorithm aspect of the project, it would be worthwhile to investigate the ability of small microprocessors to microcomputers to run the EKF and the UKF. As the final plot showed, there is a very real and measurable difference in cycle time for the for the UKF.

An additional area of interest is with regards to the initialization of the estimators. It was assumed that some initialization scheme has taken place and would provide for a reasonable initial condition. This initialization phase if crucial for estimators of this type as they only work well if their initialization parameters are not too far off from reality. While this worked for these purposes, further development should see this change.

## *5.2 Future Work*

The work presented here represents contributions in nonlinear estimation as it applies to autonomous ship board landing. It is not the final solution for any of the problems mentioned in this body of work. This project provided valuable insight, and was a stepping off point for a myriad of topics that are associated with autonomous shipboard landing. From further improvements to the estimation process, to better models on which to test with, there are several areas that could stand to benefit if focused on.

We demonstrated the ability of the measurement model to accurately track the pose of the ships deck during the short final and station keeping phases of landing. Due to the control element required to actually land the vehicle, we did not test the capabilities of the method all the way to touchdown. It is conceivable that as the distance between the sensor and the deck decreases, it would reach a critical point where it no longer provides acceptable information.

### *5.2.1 Simulation and Testing*

Throughout my research I came across other students and researchers that were making use of full motion platforms. Have this piece of hardware would go a long way in demonstrating the efficacy of estimators or controllers. I briefly looked into the cost associated with the construction of one and found the overall expenditure to be rather small considering the amount of work that could be gotten out of it. The scoping, design, and construction of such a device would be an ideal project for undergraduate students, or masters students.

### *5.2.2 Control*

While we have demonstrated the capability and necessity of relative pose estimation, we have not discussed what we should do with that state estimate. The next logical step would be to develop a guidance and control process that would make use of the state estimate to properly time when it

is safe to land, and perform the actual maneuver. Current operations have a human determine the beginning/approaching quiescent period, then instructs the pilot to land. Future work could include the development of an autonomous system that would perform the same task. Machine learning offers exciting prospects for just such a problem. Periods of calm can be learned through large training sets, composed of wave data collected from observation buoys throughout the world.

# BIBLIOGRAPHY

[1] *Guide to wave analysis and forecasting*. WMO (Series) ; no. 702. Secretariat of the World Meteorological Organization, Geneva, Switzerland, 1988.

[2] *Joint tactics, techniques, and procedures for shipboard helicopter operations.* Joint pub ; 3-04.1. Joint Chiefs of Staff, Washington, D.C.], 1997.

[3] *Development and use of UASs by the National Marine Fisheries Service for surveying marine mammals*. Marine Mammal Commission, Bethesda, MD, 2016.

[4] Shadi Abujoub, Johanna Mcphee, Cassidy Westin, and Rishad Irani. Unmanned aerial vehicle landing on maritime vessels using signal prediction of the ship motion. pages 1–9, 10 2018.

[5] S. Arora, S. Jain, S. Scherer, S. Nuske, L. Chamberlain, and S. Singh. Infrastructure-free shipdeck tracking for autonomous landing. In *2013 IEEE International Conference on Robotics and Automation*, pages 323–330, 2013.

[6] A. Bennett, D. Barrett, V. Preston, J. Woo, S. Chandra, D. Diggins, R. Chapman, A. Wee, Z. Wang, M. Rush, and I. Kerr. Autonomous vehicles for remote sample collection: Enabling marine research. In *OCEANS 2015 - Genova*, pages 1–8, May 2015.

[7] Gaemus Collins, David Twining, and Joshua Wells. Using vessel-based drones to aid commercial fishing operations. In *OCEANS 2017 - Aberdeen*, volume 2017-, pages 1–5. IEEE, 2017.

[8] John L. Crassidis and F. Landis Markley. Unscented filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, 2003.

[9] S. Davis, K. G. Ricks, and R. A. Taylor. Reflective fiducials for localization with 3d light detection and ranging scanners. *IEEE Access*, 7:45291–45300, 2019.

[10] S Dou, J J Zhang, X Y Wang, and Q Zhang. Relative pose estimation for vision-based UAV vertically landing on the ship. *IOP Conference Series: Materials Science and Engineering*, 439:032101, nov 2018.

[11] N. Filipe, M. Kontitsis, and P. Tsiotras. Extended kalman filter for spacecraft pose estimation using dual quaternions. In *2015 American Control Conference (ACC)*, pages 3187–3192, 2015.

[12] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. Wiley, Chichester, West Sussex, 2011.

[13] Matt Garratt, Hemanshu Pota, ANDREW LAMBERT, Sebastien Eckersley-Maslin, and Clement Farabet. Visual tracking and lidar relative positioning for automated launch and recovery of an unmanned rotorcraft from ships at sea. *Naval Engineers Journal*, 121:99 – 110, 06 2009.

[14] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[15] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[16] Ben Kenwright. A beginners guide to dual-quaternions: What they are, how they work, and how to use them for 3d. In *Character Hierarchies, The 20th International Conference on Computer Graphics, Visualization and Computer Vision, WSCG 2012 Communication Proceedings,pp.1-13*.

[17] E.J. Leffens, F.L. Markley, and M.D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.

[18] P.H. Lehmann, M. Jones, and M. Hfinger. Impact of turbulence and degraded visual environment on pilot workload. *CEAS Aeronautical Journal*, 8(3):413–428, 2017.

[19] F Landis Markley and John L Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*, volume 33 of *Space Technology Library*. Springer New York, New York, NY, 2014.

[20] J. M McCarthy. An introduction to theoretical kinematics, 1990.

[21] Walter H. Munk. Origin and generation of waves. *Coastal Engineering Proceedings*, 1(1):1, Jan. 1950.

[22] Yunju Na, Hyochoong Bang, and Sung-Hoon Mok. Vision-based relative navigation using dual quaternion for spacecraft proximity operations. *International Journal of Aeronautical and Space Sciences*, 06 2019.

[23] Miguel A. Olivares-Méndez, Iván Fernando Mondragón, and Pascual Campoy. Autonomous landing of an unmanned aerial vehicle using image-based fuzzy control. 2013.

[24] A. S. Peters and J. J. Stoker. The motion of a ship, as a floating rigid body, in a seaway. *Communications on Pure and Applied Mathematics*, 10(3):399–490, 1957.

[25] Jose Sanchez-Lopez, Jesus Pestana, Srikanth Saripalli, and Pascual Campoy. Toward visual autonomous ship board landing of a vtol uav. *Journal of Intelligent Robotic Systems*, 74, 04 2014.

[26] Srikanth Saripalli. *Vision-Based Autonomous Landing of an Helicopter on a Moving Target*.

[27] Malcolm David Shuster. The nature of the quaternion. *The Journal of the Astronautical Sciences*, 56:359–373, 2008.

[28] So-Ryeok Oh, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garrett. Autonomous helicopter landing on a moving platform using a tether. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3960–3965, 2005.

[29] Hannes Sommer, Igor Gilitschenski, Michael Bloesch, Stephan Weiss, Roland Siegwart, and Juan Nieto. Why and how to avoid the flipped quaternion multiplication. *Aerospace*, 5(3):72, Jul 2018.

[30] Matthew C. Vandyke, Jana L. Schwartz, and Christopher D. Hall. Unscented kalman filtering for spacecraft attitude state and parameter estimation. In *in Proceedings of the AAS/AIAA Space Flight Mechanics Conference, no. AAS 04-115, (Maui*, 2004.

[31] Neale Watson, Michael Kelly, Ieuan Owen, and Mark White. The aerodynamic effect of an oblique wind on helicopter recovery to the queen elizabeth class aircraft carrier. 05 2019.

[32] Douglas Yazell. *Origins of the Unusual Space Shuttle Quaternion Definition*.

[33] Renato Zanetti. Rotations, transformations, left quaternions, right quaternions? *The Journal of the Astronautical Sciences*, 66(3):361–381, 2019.

[34] Lijun Zhang, Huabo Yang, Shifeng Zhang, Hong Cai, and Shan Qian. Kalman filtering for relative spacecraft attitude and position estimation: A revisit. *Journal of Guidance, Control, and Dynamics*, 37:1706–1711, 09 2014.