Tightly Integrating Optical And Inertial Sensors For Navigation Using The UKF

THESIS

Sedat Ebcin, First Lieutenant, TUAF

AFIT/GE/ENG/08-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/08-09

Tightly Integrating Optical And Inertial Sensors For Navigation Using The UKF

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Sedat Ebcin, B.S.E.E.

First Lieutenant, TUAF

March 2008

AFIT/GE/ENG/08-09

Tightly Integrating Optical And Inertial Sensors For Navigation Using The UKF

Sedat Ebcin, B.S.E.E.
First Lieutenant, TUAF

Approved:

/signed/	27 Mar 2008
Maj Michael Veth, PhD (Chairman)	Date
/signed/	27 Mar 2008
Dr. John Raquet (Member)	Date
/signed/	27 Mar 2008
Dr. Guna Seetharaman (Member)	Date

Abstract

Recently, many scientists clearly proved on their work that aircraft navigation information (position, velocity, and attitude) can be determined using optical measurements from imaging sensors combined with an inertial navigation system. This can be accomplished by tracking the locations of stationary optical features in multiple images and using the resulting geometry to estimate and remove inertial errors.

The effectiveness of fusing imaging and inertial sensors using an Extended Kalman Filter (EKF) algorithm has been shown in previous research efforts. In this approach, the idea was to increase the robustness of the feature tracking algorithm. Thus, image feature correspondence search was aided using the inertial sensor measurements, resulting in more robust feature tracking. The resulting image-aided inertial algorithm was tested using both simulation and experimental data. Although the feature correspondence search is stabilized, the overall problem remained unstable due to the well-known deleterious effects of the nonlinear measurement model. These effects caused a divergence in the EKF implementation seen during our long-duration Monte-Carlo simulations. In other words, the measurement model is highly sensitive to the current parameter estimate, which invalidates the linearized measurement model assumed by the EKF.

In order to cope with divergence problem, the Unscented (Sigma-Point) Kalman Filter (UKF) has been proposed in the literature in order to address the large class of recursive estimation problems. In this research, a variation of the UKF is applied to the image-aided inertial navigation problem, with the goal of improving upon the established limitations of our previous EKF implementation. Tightly integrating optical and inertial sensors for navigation using UKF is rigorously designed from first principles, yielding a novel hybrid UKF algorithm which increases the sigma-point density along the axes of highest uncertainty. The UKF is evaluated using a

combination of simulated and experimental data. The performance of the image-aided navigation system is analyzed and compared to the baseline EKF from our previous work.

A combination of simulation and experimental analysis indicates that the UKF algorithm is superior to the EKF, namely the divergence problem is removed and overall errors are reduced. The covariance of the UKF algorithm represents well while the processing time increases such that it requires 410 seconds to process 60 seconds of simulation where EKF algorithm needs 178 seconds only. Since the processing speed is a very important design constraint for the application, an efficient modification using quaternion is applied. Consequently, UKF algorithm is optimized such that it requires 198 seconds of processing time for 60 seconds of simulation.

Table of Contents

Page

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
GPS	Global Positioning System	1
ATR	Automatic Target Recognition	1
ICBM	Inter-Continental Ballistic Missile	1
UKF	The Unscented Kalman Filter	5
PF	Particle Filter	5
NED	North, East and Down	7
INS	Inertial Navigation Systems	11
IMU	Inertial Measurement Unit	11
WGN	White Gaussian Noise	14
EKF	Extended Kalman Filter	25
GRV	Gaussian Random Variable	25
UKF	Unscented Kalman Filter	25
UT	Unscented Transform	26
SIS	Sequential Importance Sampling	30
SIR	Sampling Importance Resampling	32
DCM	Direction Cosine Matrix	36

Tightly Integrating Optical And Inertial Sensors For Navigation Using The UKF

I. Introduction

1.1 Background

The advent of Global Positioning System (GPS) has changed precision navigation capability for navigators who have utilized mechanical instruments such as astrolabes, sextants and driftmeters to determine their position, velocity and angle precisely. The fact that GPS cannot be used in all environments forces people to find new methods. Obviously, it can be seen that there is a synergy between imaging and inertial sensors which is already being used by human or other animals. This synergy is a motivation for using optical measurements to provide perfect navigation information.

The interpretation of the image has always been the challenging problem for autonomous navigation. This is also a difficulty shared with Automatic Target Recognition (ATR). Indeed, the ATR problem in this structured environment is tractable for celestial tracking, and automatic star trackers are widely used for space navigation and ICBM guidance (see [17], [16], [9]). When ground images are to be used, the difficulties associated with image interpretation are paramount. At the same time, the problems associated with the use of optical measurements for navigation are somewhat simpler than ATR. Moreover, there are improvements motivating the use of inertial measurements to aid the image interpretation such as recent developments in feature tracking algorithms, miniaturization, and reduction in cost of inertial sensors and optical images aided by the continuing improvement in microprocessor technology.

Typically, there are two image-aiding methods depending on how the image correspondence problem is addressed. These are optic flow methods and feature-based methods. Optic flow methods are generally used for elementary motion detection, fo-

cusing on determining relative velocity, angular rates, or obstacle avoidance (see [7]). Also, these methods determine correspondence for a whole portion of the image between frames. On the contrary, feature-based methods determine correspondence for "landmarks" in the scene over multiple frames.

A rigorous, stochastic projection algorithm is presented in [29], which incorporates inertial measurements into a predictive feature transformation, effectively constraining the resulting correspondence search space. The algorithm was incorporated into an extended Kalman filter and tested experimentally in [27] using both tactical and consumer grade inertial sensors. The integrated system demonstrated at least two orders of magnitude improvement over the inertial-only navigation solution.

One nonlinear filtering approach is investigated in this thesis. In order to improve the sub-optimal performance of the extended Kalman filter, the unscented Kalman filter will be used. In the EKF, the state distribution is approximated by a jointly Gaussian random vector and propagated through a linearized approximation of the nonlinear dynamics and measurement model. Our analysis indicated this is the reason for sub-optimal performance and divergence of previous work. The UKF addresses this issue by representing the state distribution as a collection of sigma points, which are directly transformed using the nonlinear dynamics and measurement models. This has been shown in the literature to preserve additional moments of the state distribution and, as such, is more resilient to the deleterious effects of linearization errors.

1.2 Problem Definition

The fact that GPS signals are not available in all locations causes a weakness in navigation and requires the development of non-GPS based navigation reference which can aid an inertial navigation system. Thus, one of the motivations of this research is to address the benefits of tightly integrating navigation sensors, such as inertial measurement units (IMU) and global positioning system measurements. The complimentary characteristics of the two sensors allow the integrated system to perform

at levels which are much better than the levels attained by using either sensor alone (see [3]). Consequently, integrated systems have become more popular, especially in military-grade navigation systems.

The weakness in GPS-based navigation can be handled by a non-GPS navigation approach which is coupling of the imaging and inertial sensors at a deep level (see [22], [18], [2]). This technique has some important advantages. The sensors can operate in all environments while GPS signal can not be received everywhere (e.g., indoors, under trees, underwater, etc.). Secondly, passive signals are used, so they do not require the transmission (or reception) of radio signals. As a result, optical and inertial sensors are immune to disruptions in the radio spectrum.

Beside all the reasons, the most valuable motivation of this work is to improve the efficiency of previous work. In previous work, a method using extended Kalman filter is developed to integrate optical and inertial sensors at a deep level. The linearization errors of the extended Kalman filter remains uncorrected, especially at the presence of large attitude errors. This thesis describes an estimator which doesn't suffer from the linearization errors of the extended Kalman filter. The estimator should overcome the divergence problem of EKF during long-duration Monte Carlo simulations. Hopefully, this research gives better results for long-term autonomous navigation.

1.3 Assumptions

This research is made under a number of assumptions listed below.

- A strapdown inertial measurement unit (IMU) is rigidly attached to one or more cameras. Synchronized raw measurements are available from both sensors.
- The camera images areas in the environment which contain some stationary objects.
- Binocular measurements are available which provide an indication of range to objects in the environment.

- The inertial and optical sensors' relative position and orientation is known (see [26]) for a discussion of boresight calibration procedures).

1.4 Thesis Overview

The thesis is divided into five chapters. Currently, a brief background and motivation are presented in Chapter One. Chapter Two provides required background information for the optical and inertial integration problem. It also prepares the reader for the following chapter by explaining reference frames, error analysis etc. The navigation algorithm is presented in Chapter Three in details. This is followed by a description of simulation and experimental results in Chapter Four. Finally, Chapter Five is reserved for conclusion and future work.

II. Background

This chapter describes some issues required to understand the fusion of imaging and inertial sensors. This chapter begins by providing an overview of reference frames and inertial navigation. Next, the image-aided navigation techniques are defined. In previous work section, the reason why there is a need for a better approach is explained. Depending on the need for a better estimation algorithm, the unscented Kalman filter (UKF) and particle filter (PF) are presented at the end of chapter.

2.1 Reference Frames

The process of inertial navigation is defined according to a number of Cartesian co-ordinate reference frames. Since they are Cartesian, it simplifies computations in navigation. These frames are right-handed coordinate frames consisting of mutually perpendicular **x**, **y** and **z** axes. Position, velocity and orientation of a body are expressed in reference frames. For this research, the following reference frames are defined based on that presented in [4] and [24]:

- True inertial frame (*I-frame*)
- Earth-fixed inertial frame (*i-frame*)
- Earth-centered Earth-fixed frame (*e-frame*)
- Navigation frame (*n'-frame*)
- Earth-fixed navigation frame (*n-frame*)
- Body frame (*b-frame*)
- Camera frame (*c-frame*)

True inertial frame (*I-frame*) is the reference frame in which Newton's laws of motion apply. This frame is determined by fixed stars in $\Re^{\mathbf{3}}$. Due to the relative nature of universe, it doesn't have a predefined origin.

The Earth-centered inertial frame (*i-frame*) has its origin at the center of Earth and aligned with respect to the fixed stars. This non-rotating frame's **z** axis is aligned

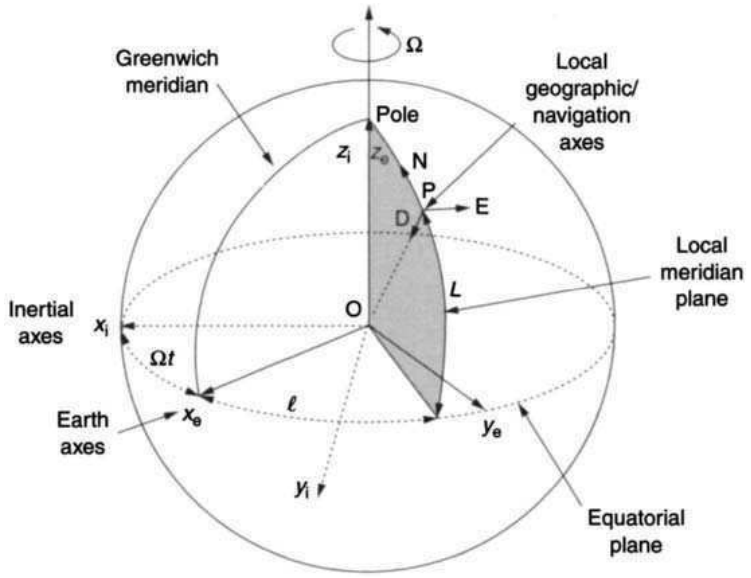

Figure 2.1: Frames of Reference. In this figure inertial, Earth and vehicle-fixed frames are illustrated as given in [24]. Vehicle-fixed frame originate at local meridian while inertial and Earth frames originate at Earth's center of mass.

with the Earth spin axis that is assumed to be invariant. The **x** and **y** axes are located on the equatorial plane. Since the frame moves with Earth, although it is not rotating, it does accelerate with respect to true inertial frame. However, Newton's laws are approximately correct in this frame and it can be considered as an inertial frame for navigational purposes.

Like the Earth-centered frame, the Earth-centered Earth-fixed frame (ECEF) (*e-frame*) has its origin at the Earth's center and it is an orthonormal basis in $\Re^{\mathbf{3}}$. Its **z** axis is aligned with the Earth's spin axis while **x** axis is on the equatorial plane pointing toward the Greenwich meridian. The **y** axis is also on the equatorial plane pointing toward 90 degrees east longitude. This frame is fixed to the Earth and rotates with Earth. Hence, the Earth-centered and ECEF frames coincide once each 24 hours.

The vehicle-fixed navigation frame (*n'*-frame) is an orthonormal basis in $\Re^{\mathbf{3}}$, with origin located at a predefined point on a vehicle. The vehicle-fixed navigation

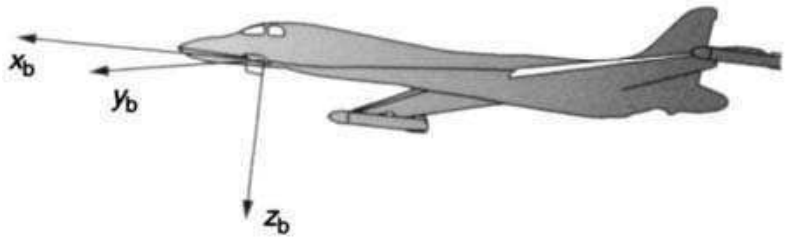

Figure 2.2: Body Reference Frame is located at a fixed point on the aircraft [24].

frame's **x**, **y** and **z** axes point in the north, east and down directions, respectively. This is called (NED) convention. For the purposes of this research, down is defined using the gravity vector. The *n'*-frame rotates with respect to the *e*-frame due to translational motion of the vehicle. The *i*, *e* and *n'*-frames are illustrated in Figure 2.1.

The Earth-fixed navigation frame (**n**-frame) is an orthonormal basis in $\Re^{\mathbf{3}}$, with origin located at a predefined location on the Earth, typically on the surface. The NED convention is current relative to the origin. As in the previous case, down is defined as the direction of the gravity vector. This frame remains fixed to the Earth. Thus, the turn rate of the navigation frame is governed by the motion of the frame's origin with respect to Earth. This frame is not useful for very-long distance navigation, but it can simplify equations for local navigation routes.

The body frame (*b*-frame) is rigidly fixed to the vehicle. The origin is located at a fixed point on the aircraft. The axes of the frame are aligned with the roll, pitch and yaw axes of the vehicle. The positive **x** axis points out the nose, the positive **y** axis points out right wing and the **z** axis points out the bottom of vehicle. The body frame is illustrated in Figure 2.2.

The camera frame (*c*-frame) is an orthonormal basis in $\Re^{\mathbf{3}}$, rigidly attached to a camera and the origin is at the camera's optical center. The **x** and **y** axes point up and to the right, respectively, and are parallel to the image plane of the camera. The **z** axis points out of the camera perpendicular to the image plane. The *c*-frame is shown in Figure 2.3.

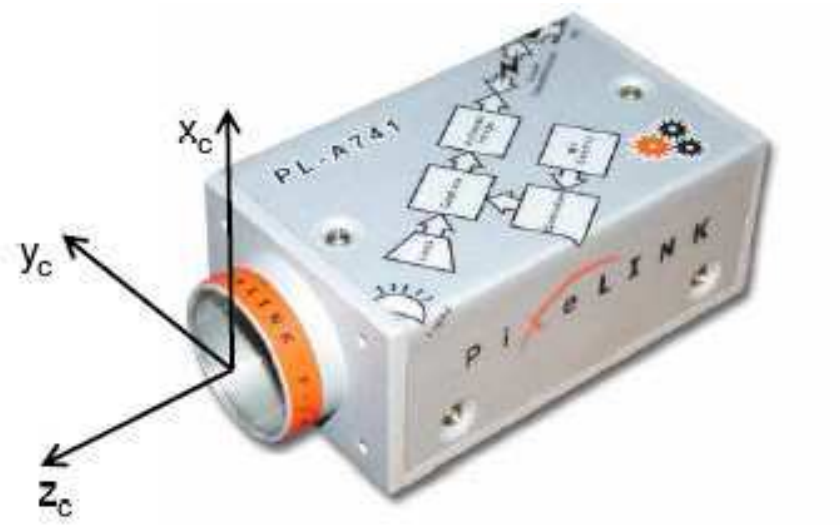

Figure 2.3: Camera frame illustration. The camera reference frame originates at the optical center of the lens [29].

2.2 Quaternions

In three dimensional space, rotations can be applied to a vector using a DCM matrix that is also used in previous work. Another possible way to represent three-dimensional rotation is quaternion algebra (see [6], [11], [24]). Actually, quaternion concept is related to rotation vector. Quaternion is a four parameter coordinate transformation (one real dimension and 3 imaginary dimensions):

$$\mathbf{Q} = \phi + \hat{i}x + \hat{j}y + \hat{k}z \tag{2.1}$$

There is a physical explanation for quaternion when quaternion is used to represent the rotation between two coordinate frames. This is called axis angle representation which is the closest physical explanation:

- ϕ = angle of rotation
- $\hat{\mathbf{v}}(x, y, z)$ = unit vector representing axis of rotation

These four components are also called "Euler's Symmetric Parameters". The formula to convert axis representation to quaternion form is:

$$\mathbf{Q} = cos(\frac{\phi}{2}) + \hat{i}(xsin(\frac{\phi}{2})) + \hat{j}(ysin(\frac{\phi}{2})) + \hat{k}(zsin(\frac{\phi}{2})) \tag{2.2}$$

In this case, quaternion can be represented as:

$$\mathbf{Q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} cos(\frac{\phi}{2}) \\ xsin(\frac{\phi}{2}) \\ ysin(\frac{\phi}{2}) \\ zsin(\frac{\phi}{2}) \end{bmatrix} \tag{2.3}$$

where the relationship between parameters is based on orthonormal vector as shown in the following section.

2.2.1 Quaternion Inverse. The inverse of a normalized quaternion is simply the conjugate:

$$\|\mathbf{Q}\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{2.4}$$

$$\mathbf{Q^{-1}} = \frac{q_1 - \hat{i}q_2 - \hat{j}q_3 - \hat{k}q_4}{\|\mathbf{Q}\|^2} \tag{2.5}$$

Simply $\mathbf{Q^{-1}}$ is:

$$\mathbf{Q^{-1}} = \begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} = q_1 - \hat{i}q_2 - \hat{j}q_3 - \hat{k}q_4 \tag{2.6}$$

Using Equations 2.1 and 2.6 orthogonality of quaternion can be shown as:

$$\begin{aligned} \mathbf{QQ^{-1}} &= (q_1 + \hat{i}q_2 + \hat{j}q_3 + \hat{k}q_4)\frac{(q_1 - \hat{i}q_2 - \hat{j}q_3 - \hat{k}q_4)}{\|\mathbf{Q}\|^2} \\ &= q_1^2 + q_2^2 + q_3^2 + q_4^2 \\ &= \|\mathbf{Q}\|^2 = 1 \end{aligned} \tag{2.7}$$

2.2.2 Quaternion Product. Quaternion multiplication is not commutative, but associative. Quaternion product is given in matrix notation below:

$$\mathbf{\tilde{Q}_1} \bigotimes \mathbf{Q_2} = \begin{bmatrix} \tilde{q}_1 & -\tilde{q}_2 & -\tilde{q}_3 & -\tilde{q}_4 \\ \tilde{q}_2 & \tilde{q}_1 & -\tilde{q}_4 & \tilde{q}_3 \\ \tilde{q}_3 & \tilde{q}_4 & \tilde{q}_1 & -\tilde{q}_2 \\ \tilde{q}_4 & -\tilde{q}_3 & \tilde{q}_2 & \tilde{q}_1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.8}$$

Two successive DCM rotations can be operated as $D_3 = D_2D_1$ where initial rotation D_1 is followed by D_2. In quaternion, rotations are combined as $Q_3 = \tilde{Q}_1 \bigotimes Q_2$ where initial rotation $\tilde{Q}_1$ is followed by Q_2.

2.2.3 Conversions. Conversions between direction cosine matrices and quaternions can be easily made using the following conversions.

2.2.3.1 Quaternion to DCM. DCM matrix can be easily attained as shown below:

$$\mathbf{D} = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_1q_2 + q_3q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_3q_4 - q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix} \tag{2.9}$$

2.2.3.2 DCM to quaternion. One way of computing quaternion from a rotation matrix $\mathbf{D}$ is as follows:

$$\mathbf{Q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4q_4}(D_{23} - D_{32}) \\ \frac{1}{4q_4}(D_{31} - D_{13}) \\ \frac{1}{4q_4}(D_{12} - D_{21}) \\ \frac{1}{2}\sqrt{1 + D_{11} + D_{22} + D_{33}} \end{bmatrix} \tag{2.10}$$

2.2.4 Vector Rotation. A vector can be rotated using quaternion in three-dimensional space. While the form of quaternion is given in Equation (2.3), the form of vector is:

$$\vec{\boldsymbol{v}} = \hat{i}v_1 + \hat{j}v_2 + \hat{k}v_3 \tag{2.11}$$

using these parameters the rotated vector has the form of:

$$\vec{\boldsymbol{v'}} = \begin{bmatrix} v'_1 \\ v'_2 \\ v'_3 \end{bmatrix} = \begin{bmatrix} (q_1^2 + q_2^2 - q_3^2 - q_4^2) & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & (q_1^2 - q_2^2 + q_3^2 - q_4^2) & 2(q_1q_2 + q_3q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_3q_4 - q_1q_2) & (q_1^2 - q_2^2 - q_3^2 + q_4^2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

2.3 Inertial Navigation

The inertial navigation systems (INS) is a universal application which has been used for estimating the position and orientation of vehicles. The operation of inertial navigation highly depends on Newton's laws of classical mechanics. According to Newton's laws, the motion of a body should continue straightly unless disturbed by an external force. In this section, basic concepts of inertial navigation are presented. The following inertial navigation basics are explained based on those presented in [13] and [24].

2.3.1 Inertial Sensors. Most of the inertial measurement units (IMU) which are the core of INS are comprised of 3-axial accelerometers and gyroscopes. But the primary sensor is the accelerometer which produces the output that is proportional to

acceleration applied along the input axis. In fact, this output is not the acceleration of the vehicle. It is the measurement of *specific force* $\mathbf{f}$ which is the difference between inertial acceleration and gravity:

$$\mathbf{f} = \mathbf{G} - \ddot{\mathbf{R}} \tag{2.12}$$

where $\mathbf{G}$ stands for gravitational field vector and $\ddot{\mathbf{R}}$ is inertial acceleration. *Specific force* is the only measurement which contains information about the vehicle motion and can be measured inside a moving vehicle without using external signals.

The gyroscopes which are other sensors of IMU are sensitive to angular velocity relative to inertial space. They are used to accomplish the orientation control of the accelerometers since gyroscopes can measure rotation relative to inertial space. Either three single-degree-of-freedom gyros or two two-degree-of-freedom gyros can be used to obtain three-axis reference.

In general, the gyroscopes and accelerometers are mounted in a cluster arrangement which is gimballed or strapdown. In this research, strapdown system is used.

2.3.2 Accelerometer Errors. Besides their benefits, both accelerometers and gyroscopes have errors which decreases the accuracy of either applied specific force or angle of rotation. These corruptions that cause accelerometer errors are mainly listed below [24]:

- **Bias:** A bias is the quantity which accelerometer reads when the specific force is zero. It is either a constant or slowly-varying additive error. It is possible to measure some bias components and correct them through factory calibration techniques. Unfortunately, some bias components remain uncorrected.
- **Scale Factor:** The accelerometer scale factor error is a multiplicative error. It can be either constant or slowly-varying. As with bias errors, some scale factor effects can be corrected through calibration.

- **Sensor Misalignment:** These are the result of mechanical fabrication and installation errors.
- **Cross Coupling:** A cross coupling error will occur in a system which assumes a fixed accelerometer input axis. If input axis is assumed fixed then it causes accelerometer to sense one of the components of acceleration along the axis which is normal to input axis:

$$\mathbf{A} = \mathbf{a_x cos}\theta - \mathbf{a_y sin}\theta \tag{2.13}$$

  In this equation $\mathbf{a_y sin}\theta$ is the cross-coupling error.
- **Vibro-pendulosity:** This is a dynamic cross-coupling error. When the accelerometer is operated in a vibratory environment, the vibrational acceleration effects both the input axis and other axis which is perpendicular to the input axis. This event causes a torque that effects the output of accelerometer.
- **Measurement Noise:** When high-bandwidth power spectral density is present, measurement noise is observed as an additive error component with high-bandwidth power spectral density. This noise component is the theoretical result of many high-bandwidth sources.
- **Gravity Model Errors:** The accelerometer measures specific force. Thus, the acceleration due to local gravity must be added to the accelerometer output to produce an estimate of acceleration in the inertial frame. Errors of this local gravity model causes additive errors in accelerometer.

2.3.3 Gyroscope Errors. Strapdown navigation systems use gyroscopes to measure angular rate relative to inertial space, $\boldsymbol{\omega}_{ib}^{b}$.

The corruptions which cause the gyroscope errors are mainly listed below [24]:

- **Bias:** This bias is an either constant or slowly-varying additive error that is independent of acceleration.

- **Acceleration-dependent Bias:** This is the bias that is a function of applied acceleration.
- **Anisoelastic Bias:** This bias is proportional to the product of acceleration along pairs of axes that are normal to each other.
- **Scale Factor:** A scale factor error is a constant or slowly-varying multiplicative error which is the ratio of output signal change to the input rate change.
- **Sensor Misalignment:** As in accelerometer errors, sensor misalignment errors are also a result of mechanical fabrication and installation errors.
- **Measurement Noise:** This is an additive error component with high-bandwidth power spectral density which is the theoretical result of many high-bandwidth sources such as electrical noise, thermal noise, etc.

2.4 Inertial Navigation Error Model

Inertial navigation error model is developed based on the inertial navigation dynamics [24]. The error models, described in this section, are the same models presented in [25].

2.4.1 Inertial Sensor Error Model. Both the accelerometer and gyroscopic error models consist of a bias and a random noise where the random noises are modeled as an additive white Gaussian noise (WGN) process and the biases are modeled as first-order Gauss-Markov processes [10] based on the specification for the IMU:

$$\mathbf{f}_m^b = \mathbf{f}^b + \mathbf{a}^b + \mathbf{w}_a^b \tag{2.14}$$

$$\boldsymbol{\omega}_{ib_m}^b = \boldsymbol{\omega}_{ib}^b + \mathbf{b}^b + \mathbf{w}_b^b \tag{2.15}$$

where $\mathbf{a}^b$ and $\mathbf{b}^b$ are the accelerometer and gyroscopic biases, $\mathbf{w}_a^b$ and $\mathbf{w}_b^b$ are accelerometer and gyroscope additive white Gaussian noise processes, respectively.

The bias differential equations for a first-order Gauss-Markov process model for both accelerometer and gyroscope are expressed as:

$$\dot{\mathbf{a}}^b = -\frac{1}{\tau_a}\mathbf{a}^b + \mathbf{v}_a^b \tag{2.16}$$

$$\dot{\mathbf{b}}^b = -\frac{1}{\tau_b}\mathbf{b}^b + \mathbf{v}_b^b \tag{2.17}$$

where τ_a and τ_b are the time constants, $\mathbf{w}_a^b$ and $\mathbf{w}_b^b$ are the WGN terms for the accelerometer and gyroscopic biases, respectively.

2.4.2 Position And Velocity Error Development. The position and velocity errors are modeled as a stochastic process based on the Pinson navigation error model [24]. The position and velocity error models are the same models presented in [25]. These errors are:

$$\delta\mathbf{p}^\mathbf{n} = \bar{\mathbf{p}}^\mathbf{n} - \mathbf{p}^\mathbf{n} \tag{2.18}$$

$$\delta\mathbf{v}^\mathbf{n} = \bar{\mathbf{v}}^\mathbf{n} - \mathbf{v}^\mathbf{n} \tag{2.19}$$

The position error can be explained using the kinematic relationship between position and velocity:

$$\delta\dot{\mathbf{p}}^\mathbf{n} = \delta\mathbf{v}^\mathbf{n} \tag{2.20}$$

and the acceleration error vector is:

$$\delta\dot{\mathbf{v}}^\mathbf{n} = \dot{\bar{\mathbf{v}}}^\mathbf{n} - \dot{\mathbf{v}}^\mathbf{n} \tag{2.21}$$

where $\dot{\mathbf{v}}^\mathbf{n}$ is the acceleration dynamics equation and $\dot{\bar{\mathbf{v}}}^\mathbf{n}$ is the calculated velocity vector differential equation.

In order to derive the dynamics of velocity error, the acceleration dynamics equation, in which the gravity function is substituted, is used [25]:

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \mathbf{v}^n + \mathbf{C}_e^n \mathbf{g}^e \left(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n\right) \tag{2.22}$$

This equation consists of specific force, Coriolis effects and the gravity vector, $\mathbf{g}^{\mathbf{e}}(\mathbf{p}^{\mathbf{e}})$. The calculated velocity vector differential equation is corrupted by the accelerometer measurement errors and the attitude errors [25]. Combining the calculated velocity vector differential equation with the position, velocity, attitude and accelerometer measurement error equations results in:

$$\begin{aligned} \dot{\tilde{\mathbf{v}}}^n &= \left(\mathbf{I} - (\boldsymbol{\psi}\times)\right) \mathbf{C}_b^n \left(\mathbf{f}^b + \mathbf{a}^b + \mathbf{w}_a^b\right) - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \left(\mathbf{v}^n + \delta\mathbf{v}^n\right) \\ &\quad + \mathbf{C}_e^n \mathbf{g}^e \left(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n + \mathbf{C}_n^e \delta\mathbf{p}^n\right) \end{aligned} \tag{2.23}$$

Rewriting the Equation (2.21) using the Equations (2.22), (2.23) and eliminating second-order terms yields:

$$\delta\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{a}^b - (\boldsymbol{\psi}\times)\, \mathbf{C}_b^n \mathbf{f}^b - 2\mathbf{C}_e^n \boldsymbol{\Omega}_{ie}^e \mathbf{C}_n^e \delta\mathbf{v}^n + \mathbf{C}_e^n \mathbf{G} \mathbf{C}_n^e \delta\mathbf{p}^n + \mathbf{C}_b^n \mathbf{w}_a^b \tag{2.24}$$

where $\mathbf{G}$ is the gradient of the gravity vector, $\mathbf{g}^e(\mathbf{p}_0^e + \mathbf{C}_n^e \mathbf{p}^n)$ (see [25] for details).

2.4.3 Attitude Error Development. The attitude errors are modeled as a stochastic process based on the Pinson navigation error model [24]. According to the Pinson navigation error model, small angular errors about nominal orientation can be represented by a simple rotation error vector. However, this is not true for general orientation. The attitude error model is the same model presented in [25].

The attitude error vector is modeled as below:

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_n \\ \psi_e \\ \psi_d \end{bmatrix} \tag{2.25}$$

The skew-symmetric form of attitude error vector $(\boldsymbol{\psi}\times)$ is used to express the computed body-to-Earth-fixed navigation frame DCM [24]:

$$\bar{\mathbf{C}}_b^n \approx (\mathbf{I} - (\boldsymbol{\psi}\times))\,\mathbf{C}_b^n \tag{2.26}$$

Taking the derivative of Equation 2.26:

$$\dot{\bar{\mathbf{C}}}_b^n = -(\dot{\boldsymbol{\psi}}\times)\mathbf{C}_b^n + (\mathbf{I} - (\boldsymbol{\psi}\times))\,\dot{\mathbf{C}}_b^n \tag{2.27}$$

The derivative of a DCM is:

$$\dot{\mathbf{C}}_{\mathbf{b}}^{\mathbf{n}} = \mathbf{C}_{\mathbf{b}}^{\mathbf{n}}\boldsymbol{\Omega}_{\mathbf{nb}}^{\mathbf{b}} \tag{2.28}$$

Combining Equations 2.27, 2.28 and solving for $(\dot{\boldsymbol{\psi}}\times)$ results in:

$$(\dot{\boldsymbol{\psi}}\times) = [(\mathbf{I} - (\boldsymbol{\psi}\times))\,\mathbf{C}_b^n\boldsymbol{\Omega}_{nb}^b - \bar{\mathbf{C}}_b^n\bar{\boldsymbol{\Omega}}_{nb}^b]\mathbf{C}_n^b \tag{2.29}$$

Since perfect measurements are not available, the calculated body-to-earth rotation rate vector can be expressed as:

$$\bar{\boldsymbol{\omega}}_{nb}^b = \boldsymbol{\omega}_{ib_m}^b - \bar{\mathbf{C}}_n^b\mathbf{C}_e^n\boldsymbol{\omega}_{ie}^e \tag{2.30}$$

Combining Equations 2.15, 2.26, 2.29 and 2.30 results in:

$$\begin{aligned}(\dot{\boldsymbol{\psi}}\times) &= (\mathbf{I}-(\boldsymbol{\psi}\times))\,\mathbf{C}_b^n\boldsymbol{\Omega}_{nb}^b\mathbf{C}_n^b \\ &\quad -(\mathbf{I}-(\boldsymbol{\psi}\times))\,\mathbf{C}_b^n\left[\boldsymbol{\Omega}_{nb}^b+\left(\mathbf{b}^b\times\right)-\left[\left(\mathbf{C}_n^b\left[\boldsymbol{\psi}\times\right]\mathbf{C}_e^n\boldsymbol{\omega}_{ie}^e\right)\times\right]\right. \\ &\quad \left.+\left(\mathbf{w}_b^b\times\right)\right]\mathbf{C}_n^b \end{aligned} \tag{2.31}$$

After eliminating common terms, second-order terms and collapsing the skew-symmetric form, Equation 2.31 yields the linearized angular error equation:

$$\dot{\boldsymbol{\psi}} = -\left(\left(\mathbf{C}_e^n\boldsymbol{\omega}_{ie}^e\right)\times\right)\boldsymbol{\psi}-\mathbf{C}_b^n\mathbf{b}^b-\mathbf{C}_b^n\mathbf{w}_b^b \tag{2.32}$$

2.5 Measurement Model

The measurement model is the same model presented in [25]. The model is based on the landmark of opportunity which are modeled as stationary. In order to incorporate landmarks into navigation filter, initial landmark location is determined. Then, the measurement errors are calculated. Although there are different methods to determine the initial location of landmark, the method of binocular stereopsis with no terrain model is addressed and used in this research to estimate landmark location.

2.5.1 Estimating Location of Landmark Using Binocular Stereopsis. It is possible to determine the distance of an object within an image utilizing binocular measurements. In order to estimate the landmark location, a binocular disparity reference frame is located between the optical centers of two cameras (see Figure 2.4). This method is completely same as the one presented in [25].

Determining the landmark location requires three steps:

- Select the feature and match between image pairs,
- Calculate the relative line of sight vector ($s_0^{c_0}$),
- Estimate the location of landmark.

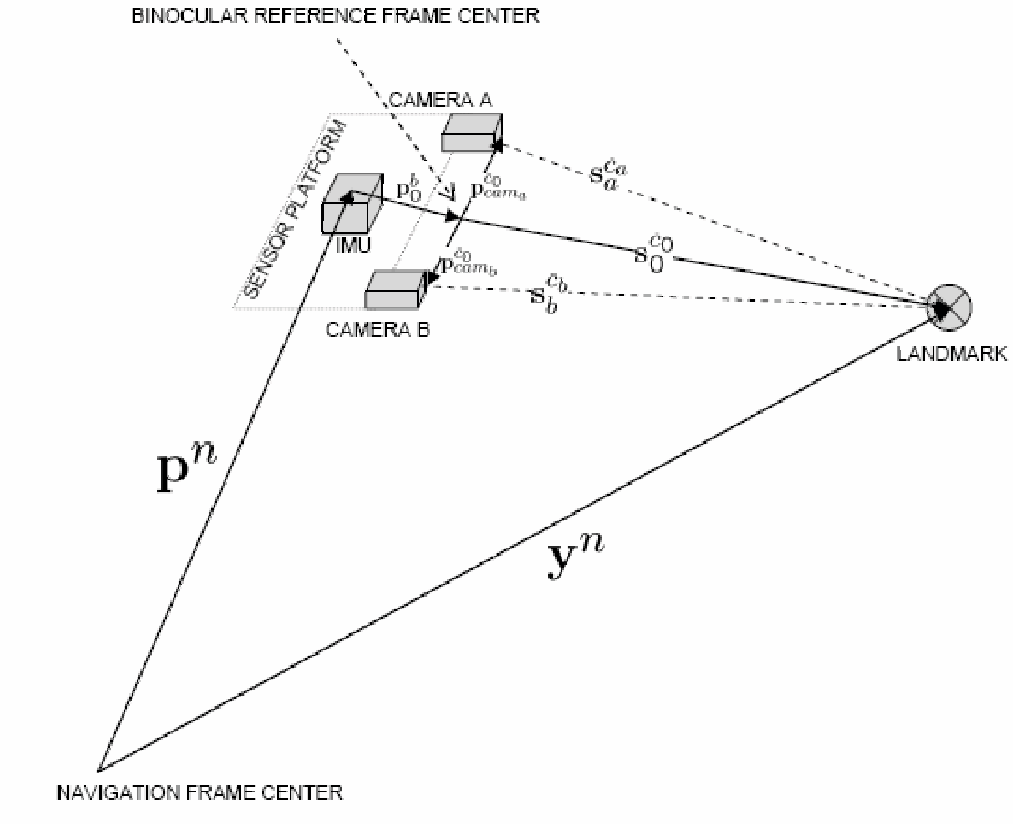

Figure 2.4: Binocular imaging geometry. The line of sight vector, $\mathbf{s_0^{co}}$, is a function of the landmark location, $\mathbf{y^n}$, sensor platform location, $\mathbf{p^n}$, the lever arm from the inertial sensor to the binocular reference point to *camera a* and *camera b*, $\mathbf{p_{c_a}^{co}}$ and $\mathbf{p_{c_b}^{co}}$, respectively [25].

Initially, a candidate feature is chosen from *image a* and statistically projected into the feature space of *image b* with corresponding mean and uncertainty. The description of the pixel location of the feature in the *b frame* is:

$$\mathbf{z^b} = \mathbf{T^p_{c_b}} \breve{\mathbf{s}}^{\mathbf{c_b}} \tag{2.33}$$

$$\mathbf{s}^{\mathbf{c_b}} = \mathbf{C}^{\mathbf{c_b}}_{\mathbf{c_0}} \left(\boldsymbol{\xi} \mathbf{C}^{\mathbf{c_0}}_{\mathbf{c_a}} \mathbf{T}^{\mathbf{c_a}}_{\mathbf{p}} \mathbf{z^a} + \mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_a}} - \mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_b}} \right) \tag{2.34}$$

where $\breve{\mathbf{s}}^{\mathbf{c_b}}$ is the normalized form of $\mathbf{s}^{\mathbf{c_b}}$ about *z axis*. $\mathbf{T}^{\mathbf{c_a}}_{\mathbf{p}}$, $\mathbf{T}^{\mathbf{p}}_{\mathbf{c_b}}$ are camera projection matrices and $\mathbf{C}^{\mathbf{c_0}}_{\mathbf{c_a}}$, $\mathbf{C}^{\mathbf{c_b}}_{\mathbf{c_0}}$ are the orientation direction cosine matrices for *camera a* and *camera b*, respectively. As seen on Figure 2.4, $\mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_a}}$ and $\mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_b}}$ are the location vectors from the $\mathbf{c_0}$ frame for *camera a* and *camera b*, respectively. $\boldsymbol{\xi}$ is the distance parameter that is modeled as a Gaussian distribution (see [25] for details).

After calculating the predicted pixel location mean ($\hat{\mathbf{z}}^{\mathbf{b}}$) and error covariance matrix ($\mathbf{P_{z_b z_b}}$) [25] on *image b*, the feature that minimizes the comparative feature description distance is chosen.

The second step of determining the landmark location is to estimate the relative line of sight vector ($\mathbf{s}^{\mathbf{c_0}}_{\mathbf{0}}$). It is possible to estimate the relative line of sight vector using nonlinear regression techniques, since the pixel locations of the feature in both *camera a* and *camera b* frame are function of $\mathbf{s}^{\mathbf{c_0}}_{\mathbf{0}}$:

$$\mathbf{z^a} = \mathbf{f}\left[\mathbf{s}^{\mathbf{c_0}}_{\mathbf{0}}, \mathbf{C}^{\mathbf{c_a}}_{\mathbf{c_0}}, \mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_a}}, \mathbf{T}^{\mathbf{p}}_{\mathbf{c_a}}\right] \tag{2.35}$$

$$\mathbf{z^b} = \mathbf{f}\left[\mathbf{s}^{\mathbf{c_0}}_{\mathbf{0}}, \mathbf{C}^{\mathbf{c_b}}_{\mathbf{c_0}}, \mathbf{p}^{\mathbf{c_0}}_{\mathbf{c_b}}, \mathbf{T}^{\mathbf{p}}_{\mathbf{c_b}}\right] \tag{2.36}$$

Expanding these equations yields:

$$\mathbf{z^a} = \mathbf{T^p_{c_a}} \breve{\mathbf{s}}^{\mathbf{c_a}}_{\mathbf{a}} \tag{2.37}$$

$$\mathbf{z^b} = \mathbf{T^p_{c_b}} \breve{\mathbf{s}}^{\mathbf{c_b}}_{\mathbf{b}} \tag{2.38}$$

where

$$\mathbf{s}_{\mathbf{a}}^{\mathbf{c_a}} = \mathbf{C}_{\mathbf{c_0}}^{\mathbf{c_a}} \left(\mathbf{s}_{\mathbf{0}}^{\mathbf{c_0}} - \mathbf{p}_{\mathbf{c_a}}^{\mathbf{c_0}} \right) \tag{2.39}$$

$$\mathbf{s}_{\mathbf{b}}^{\mathbf{c_b}} = \mathbf{C}_{\mathbf{c_0}}^{\mathbf{c_b}} \left(\mathbf{s}_{\mathbf{0}}^{\mathbf{c_0}} - \mathbf{p}_{\mathbf{c_b}}^{\mathbf{c_0}} \right) \tag{2.40}$$

The final step is the estimation of the landmark location. The landmark location can be calculated using the line of sight vector, $\mathbf{s}_{\mathbf{0}}^{\mathbf{c_0}}$, and the navigation state estimate (see Figure 2.4):

$$\mathbf{y}^{\mathbf{n}} = \mathbf{p}^{\mathbf{n}} + \mathbf{C}_{\mathbf{b}}^{\mathbf{n}} \left(\mathbf{p}_{\mathbf{0}}^{\mathbf{b}} + \mathbf{C}_{\mathbf{c_0}}^{\mathbf{b}} \mathbf{s}_{\mathbf{0}}^{\mathbf{c_0}} \right) \tag{2.41}$$

The landmark location estimation might result in negative distance estimate due to the pixel measurement noise if the landmark is a large distance away [25]. However, this condition can be easily detected before applying regression method and the binocular feature can be changed.

2.6 Image-Aided Navigation Techniques

In this section, some of the image-aided inertial navigation techniques are presented.

2.6.1 INS Aiding By Tracking An Unknown Ground Object. As presented in [15], this theory is developed to increase the accuracy of INS, which has a degradation in accuracy over time, by updating periodically. In this approach, a precision telescope is used in an aircraft. This telescope is remained pointed to the ground object with a gimbal where the telescope is mounted on.

The main idea is to find the aircraft's angular navigation variables. These variables consist of 3 positional variables which are ψ (heading), θ (pitch), ϕ (roll) and 2 angular variables which are H (the regressor) and $\boldsymbol{\gamma}$ (the angle between the line of sight and the velocity vector). In order to find these variables, a plane is formed by the aircraft's velocity vector. Local frame of reference and aircraft's body axes are

collocated initially. The direction to the ground object is called Line of Sight (LOS) which is measured relative to the aircraft's body axes. Then, it is possible to estimate the angles between the aircraft's inertial velocity vector and aircraft's body axes:

$$\boldsymbol{\alpha}' = arctan(\frac{w}{u}) \tag{2.42}$$

$$\boldsymbol{\beta}' = arctan(\frac{v}{u}) \tag{2.43}$$

where **u**, **v** and **w** are the North, East and Down components of the inertial velocity vector and $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ angles are related to the aircraft's attitude, heading and flight path. After finding $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$, these angles can be related to the aircraft's angular navigation variables.

In the first phase mentioned above, the necessary measurements are obtained. Next, these measurements are used to update INS-provided estimates. The results of Phase 1, which give information about the drift of aircraft, are used in update cycle in Phase 2.

This technique reveals that the LOS measurements are conducive to a stand alone estimate of the angles $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ included between the aircrafts inertial velocity vector and the aircraft body axes. The measured $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ angles are related to the aircrafts attitude, heading, and flight path angle angular navigation variables. As a result, accurate position estimation is possible using prior ground object position and altitude information. Since this method is based on a stationary ground object, it can be called as an INS aiding using a modern driftmeter.

2.6.2 Inertial Navigation Sensor Integrated Motion Analysis For Obstacle Detection. This technique, which is presented in [1], is developed to obtain a desired obstacle detection system that should work in all-weather conditions in day or night. Also, it should be minimizing the threat to the vehicle with a graceful degradation. This system is a passive approach and uses inertial measurements by calculating the

translation and rotation between images to estimate obstacle distance. The algorithm presented in the article consists of seven steps:

1. Reading of the input frames, which will be compared, along with their associated inertial data,
2. Selection of interest points from each input frame,
3. Computation of the location of FOE (focus of expansion) using the INS derived velocity vector in each frame,
4. Projection of FOE and interest points (in frame B) onto image plane parallel to image A,
5. Matching of interest points,
6. Computation of range to each interest points,
7. Creation of a dense range map using context.

The interest points are computed by passing an operator over each frame among a set of distinguishable points. Before matching process, these selected interest points are derotated into a plane that is parallel to frame A. A projection matrix P is used to derotate and the angles of the P matrix are obtained from inertial sensors.

Matching of interest points is performed as one-to-one match between frames after identifying three candidate matches for each interest point in frame B. Range can be calculated after matching process. This technique is a good example for INS integrated motion analysis. It also gives the idea of combining inertial sensors with image sensors since it is proved that image transformations are useful for navigation.

2.6.3 Augmenting Inertial Navigation With Image-Based Motion Estimation. This is another way of augmenting inertial navigation with image-based motion estimation. In this application as presented in [21], the idea is to help the problem of landing on hazardous terrain by producing estimates of spacecraft relative motion. The main contribution of current work is using Kalman filter for these estimates.

The algorithm used in this approach is two-frame feature-based motion estimation. Due to the fact that most INS require absolute position and orientation information in order to reduce tracking errors, especially in cases where the relative pose measurements are available at a lower rate than the IMU signals, a variant of Kalman filter (called 6DOF Kalman filter) is developed. This filter is capabled of optimally integrating the inertial measurements with displacement estimates provided by a vision-based feature tracking algorithm.

In the first step, two images and laser altimeter readings are recorded in a short period. The features are selected randomly. The motion between these pairs of measurements are estimated by selecting distinct features on first image and tracking them on second image. The motion state and the covariance are computed given the feature matches. Finally, altimetry is combined with the motion in order to compute the magnetude of translation. A variant of Kalman filter is used to estimate the errors in the estimated states which are derived from sensors. Relative position measurement, attitude measurement and pose measurement errors are estimated in the filter and applied to estimated states via update equations.

As a result, Kalman filter methodology has been applied successfully. Image-based motion estimation can be used as measurements in inertial navigation which is degrading over time and needs to be updated with measurements. This research provides advantages over using each method separately by increasing state estimation accuracy. It also reveals that the fusion of IMU measurements with motion estimates from the Image-based Motion Estimation increases the robustness of the Kalman filter estimator due to the fact that these two sensing modalities have complimentary operating conditions and noise characteristics.

2.6.4 Navigation Using Optical Measurements of Objects at Unknown Locations. In this approach, presented in [19], the idea is to navigate via optical measurements using some advantages of cameras such that they are passive and difficult

to jam which makes them useful for military operations. What is more, this approach does not require prior knowledge of the location of the objects being tracked.

While dealing with the error estimates to find true trajectory, this research is based on some assumptions as followed:

- An INS is available for navigation,
- There is a working target registration algorithm to identify a target in multiple images,
- These targets are stationary objects.

For the algorithm, an extended Kalman filter is used to integrate the inertial measurements with the optical measurements. It also makes use of GPS measurements to initialize the filter. While the measurement model is different, the Kalman filter state model is very similar to the approach presented in this research.

A flight test is generated to validate that the algorithm works properly and different test cases (GPS only, optical only, no update etc.) are evaluated. Results obviously proves that optical measurement update reduced the position error by 70 percent. This is a very encouraging test result to incorporate optical and inertial sensors for autonomous navigation.

2.7 Unscented Kalman Filter

In highly non-linear equations the extended Kalman filter (EKF) has a poor performance, since only the mean is propagated through the non-linear function. The state distribution is propagated through the first-order linearization of non-linear system. This might cause large errors in the mean and covariance of transformed Gaussian random variable (GRV). In this section, the unscented transformation and the filter algorithm is explained in details as presented in [5].

While having the same approximation issues of EKF (e.g., GRV), *the unscented Kalman filter (UKF)* uses a deterministic sampling approach (unscented transform

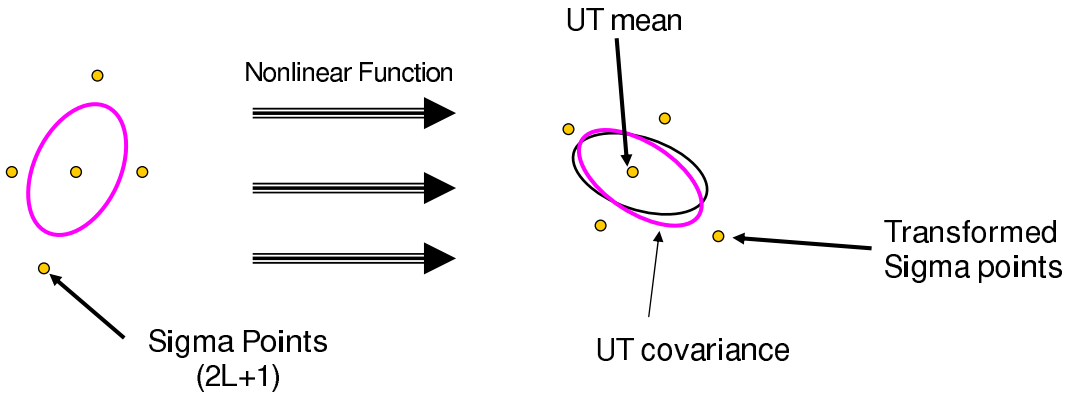

Figure 2.5: Unscented Transformation. UKF uses a deterministic sampling approach, unscented transform (UT) in which the state distribution is represented using a minimal set of carefully chosen sigma points [5].

(UT)) in which the state distribution is represented using a minimal set of carefully chosen sigma points around the mean. Besides the mean, the sigma points are also transformed using the non-linear function. As seen on Figure 2.5, transformed sigma points are used in calculating UT mean and UT covariance. Consequently, the non-linear function is applied to every single point to yield a set of transformed sigma points without losing the true mean and covariance.

2.7.1 Unscented Transformation. This method is used to calculate the statistics of a GRV through a nonlinear transformation, $\mathbf{y} = \mathbf{h}(\mathbf{x})$. The random variable $\mathbf{x}$ (dimension L) with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_x}$ is used to generate a matrix χ of $\mathbf{2L + 1}$ sigma points as follows:

$$\begin{aligned} \chi_{\mathbf{0}} &= \bar{\mathbf{x}} \\ \chi_{\mathbf{i}} &= \bar{\mathbf{x}} + (\sqrt{(L+\boldsymbol{\lambda})\mathbf{P_x}})_i \qquad i = 1, ..., L \\ \chi_{\mathbf{i}} &= \bar{\mathbf{x}} - (\sqrt{(L+\boldsymbol{\lambda})\mathbf{P_x}})_{i-L} \qquad i = L+1, ..., 2L \end{aligned} \tag{2.44}$$

where $\sqrt{(L+\boldsymbol{\lambda})\mathbf{P_x}}$ is the ith column of the matrix square root and $\boldsymbol{\lambda}$ is the scaling parameter defined by:

$$\boldsymbol{\lambda} = \boldsymbol{\alpha}^{\mathbf{2}}(L+\mathbf{K}) - L \tag{2.45}$$

where $\boldsymbol{\alpha}$ is the constant value usually set to $\mathbf{1} \leq \boldsymbol{\alpha} \leq \mathbf{10^{-4}}$ ($\boldsymbol{\alpha} = 10^{-2}$ is used for the algorithm) and used to determine the spread of sigma points and $\mathbf{K}$ is the secondary scaling parameter (usually $\mathbf{K = 3 - L}$ [5]). Once χ is computed, each column of χ is propagated to the next time step through the non-linear function to yield the set of transformed sigma points (see Figure 2.5). The mean and covariance of $\mathbf{y}$ is approximated using the weighted average and weighted outer product of transformed sigma points $\mathbf{Y}$, respectively.

$$\bar{\mathbf{y}} \approx \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W_i^{(m)} Y_i} \tag{2.46}$$

$$\mathbf{P_y} \approx \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W_i^{(c)}} (\mathbf{Y_i} - \bar{\mathbf{y}})(\mathbf{Y_i} - \bar{\mathbf{y}})^{\mathbf{T}} \tag{2.47}$$

where the weights are:

$$\begin{aligned} \mathbf{W_0^{(m)}} &= \frac{\boldsymbol{\lambda}}{\mathbf{L} + \boldsymbol{\lambda}} \\ \mathbf{W_0^{(c)}} &= \frac{\boldsymbol{\lambda}}{\mathbf{L} + \boldsymbol{\lambda}} + \mathbf{1} - \boldsymbol{\alpha^2} + \boldsymbol{\beta} \\ \mathbf{W_i^{(m)}} &= \mathbf{W_i^{(c)}} = \frac{\mathbf{1}}{\mathbf{2(L + \lambda)}}, \qquad \mathbf{i = 1, ..., 2L} \end{aligned} \tag{2.48}$$

where $\boldsymbol{\beta}$ is used to incorporate prior knowledge of distribution of $\mathbf{x}$ ($\boldsymbol{\beta} = \mathbf{2}$ is optimal for Gaussian [5]).

2.7.2 Unscented Filter. The unscented Kalman filter is the straight forward extension of the UT. Ultimately, we seek stochastic difference equations of the following form [5]:

$$\mathbf{x_{k+1}} = \mathbf{F}(\mathbf{x_k}, \mathbf{u_k}, \mathbf{v_k}) \tag{2.49}$$

$$\mathbf{y_k} = \mathbf{H}(\mathbf{x_k}, \mathbf{n_k}) \tag{2.50}$$

where $\mathbf{x_k}$ represents error state vector, $\mathbf{u_k}$ is a known input, $\mathbf{v_k}$ is the process noise that drives the dynamic system, $\mathbf{y_k}$ is the observed measurement and the observation

noise is given by $\mathbf{n_k}$. For an additive (zero mean) noise case, the UKF consists of the following steps:

1. Initialization in UKF is in the same way as EKF,

$$\hat{\mathbf{x}}_\mathbf{0} = \mathbf{E}[\mathbf{x_0}] \tag{2.51}$$

$$\mathbf{P_0} = \mathbf{E}[(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})(\mathbf{x_0} - \hat{\mathbf{x}}_\mathbf{0})^\mathbf{T}] \tag{2.52}$$

2. Generate matrix χ of sigma points,

$$\chi_{\mathbf{k-1}} = [\hat{\mathbf{x}}_{\mathbf{k-1}} \quad \hat{\mathbf{X}}_{\mathbf{k-1}} + \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k-1}}} \quad \hat{\mathbf{X}}_{\mathbf{k-1}} - \boldsymbol{\gamma}\sqrt{\mathbf{P}_{\mathbf{k-1}}}] \tag{2.53}$$

where $\hat{\mathbf{X}}_{\mathbf{k-1}}$ is:

$$\hat{\mathbf{X}}_{\mathbf{k-1}} = [\hat{\mathbf{x}}_{\mathbf{k-1}} \quad \hat{\mathbf{x}}_{\mathbf{k-1}} \quad ...]_{LxL} \tag{2.54}$$

3. Transform each point through the process model for time-update,

$$\chi^*_{\mathbf{k|k-1}} = \mathbf{F}(\chi_{\mathbf{k-1}}, \mathbf{u_{k-1}}) \tag{2.55}$$

$$\hat{\mathbf{x}}^-_\mathbf{k} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{m})} \chi^*_{\mathbf{i,k|k-1}} \tag{2.56}$$

$$\mathbf{P}^-_\mathbf{k} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{c})} (\chi^*_{\mathbf{i,k|k-1}} - \hat{\mathbf{x}}^-_\mathbf{k})(\chi^*_{\mathbf{i,k|k-1}} - \hat{\mathbf{x}}^-_\mathbf{k})^\mathbf{T} + \mathbf{R}^\mathbf{v} \tag{2.57}$$

4. Instantiate each of the prediction points using augmented sigma points through the observation model,

$$\chi_{\mathbf{k|k-1}} = [\chi^*_{\mathbf{k|k-1}} \quad \chi^*_{\mathbf{0,k|k-1}} + \boldsymbol{\gamma}\sqrt{\mathbf{R}^\mathbf{v}} \quad \chi^*_{\mathbf{0,k|k-1}} - \boldsymbol{\gamma}\sqrt{\mathbf{R}^\mathbf{v}}] \tag{2.58}$$

$$\mathbf{Y_{k|k-1}} = \mathbf{H}(\chi_{\mathbf{k|k-1}}) \tag{2.59}$$

$$\hat{\mathbf{y}}^-_\mathbf{k} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{m})} \mathbf{Y_{i,k|k-1}} \tag{2.60}$$

5. Apply measurement-update using measurement covariance and cross correlation matrices.

$$\mathbf{P}_{\bar{\mathbf{y}}_\mathbf{k}\bar{\mathbf{y}}_\mathbf{k}} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{c})}(\mathbf{Y}_{\mathbf{i,k|k-1}} - \hat{\mathbf{y}}_\mathbf{k}^-)(\mathbf{Y}_{\mathbf{i,k|k-1}} - \hat{\mathbf{y}}_\mathbf{k}^-)^\mathbf{T} + \mathbf{R}^\mathbf{n} \tag{2.61}$$

$$\mathbf{P}_{\mathbf{x}_\mathbf{k}\mathbf{y}_\mathbf{k}} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{c})}(\chi_{\mathbf{i,k|k-1}} - \hat{\mathbf{x}}_\mathbf{k}^-)(\mathbf{Y}_{\mathbf{i,k|k-1}} - \hat{\mathbf{y}}_\mathbf{k}^-)^\mathbf{T} \tag{2.62}$$

$$\mathbf{K}_\mathbf{k} = \mathbf{P}_{\mathbf{x}_\mathbf{k}\mathbf{y}_\mathbf{k}}\mathbf{P}^{-\mathbf{1}}_{\bar{\mathbf{y}}_\mathbf{k}\bar{\mathbf{y}}_\mathbf{k}} \tag{2.63}$$

$$\hat{\mathbf{x}}_\mathbf{k}^+ = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K}_\mathbf{k}(\mathbf{y}_\mathbf{k} - \hat{\mathbf{y}}_\mathbf{k}^-) \tag{2.64}$$

$$\mathbf{P}_\mathbf{k} = \mathbf{P}_\mathbf{k}^- - \mathbf{K}_\mathbf{k}\mathbf{P}_{\bar{\mathbf{y}}_\mathbf{k}\bar{\mathbf{y}}_\mathbf{k}}\mathbf{K}_\mathbf{k}^\mathbf{T} \tag{2.65}$$

where $\boldsymbol{\gamma} = \sqrt{L + \boldsymbol{\lambda}}$, $\mathbf{R}^\mathbf{v}$ is the process-noise covariance and $\mathbf{R}^\mathbf{n}$ is the measurement-noise covariance. As shown in the algorithm, it is very easy to implement correlated noises. The parameters $\boldsymbol{\gamma}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\mathbf{K}$ are empirically determined. Hence, various modifications are also possible.

2.8 Particle Filter

The *particle filter* is a recursive, non-linear estimation algorithm based on a sequential Monte Carlo method. The particle filter can be considered as an extension to the Kalman filter, since they both can be used for the same kinds of engineering problems (e.g., recursive estimation). However, particle filters can be more accurate than EKF and UKF if the choice of particles represents the pdf well. The particle filter uses Bayesian approach and allows us to represent the posterior density by a set of randomly chosen particles. If we can find the posterior distribution with sufficient number of weighted particles, then the Bayesian optimal estimate can be computed easily. Such that:

$$\mathbf{E}(\mathbf{g}(\mathbf{x}_\mathbf{k})) = \int \mathbf{g}(\mathbf{x}_\mathbf{k})\hat{\mathbf{p}}(\mathbf{x}_\mathbf{k}|\mathbf{Y}_\mathbf{0}^\mathbf{k})\mathbf{d}\mathbf{x}_\mathbf{k} \tag{2.66}$$

where $\mathbf{p}(\mathbf{x_k}|\mathbf{Y_0^k})$ is the posterior distribution which is used in Equation 2.67. There are two cases where the particle filter can offer improved performance:

- Non-Gaussianity
- Non-Linearity

In contrast to the EKF and UKF, the particle filter does not require a Gaussian assumption. In this section, a brief view to the fundamentals of particle filters will be described as presented in [5] and [12].

While depending on Monte Carlo-based statistics to determine the location of a collection of particles, the particle filter utilize sequential importance sampling (SIS) to represent the desired p.d.f. accurately. The overall goal is to estimate unknown state $\mathbf{x_k}$ given a sequence of observations $\mathbf{Y_0^k}$:

$$\hat{\mathbf{p}}(\mathbf{x_k}|\mathbf{Y_0^k}) = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \delta(\mathbf{x_k} - \mathbf{x_k^{(i)}}) \tag{2.67}$$

where δ is the Dirac delta function, $\mathbf{x_k^{(i)}}$ is a random sample drawn from $\mathbf{p}(\mathbf{x_k}|\mathbf{Y_0^k})$ which is a marginal of the full posterior density $\mathbf{p}(\mathbf{X_0^k}|\mathbf{Y_0^k})$ where $\mathbf{X_0^k}$ is sets of states that can be attained given the observations $\mathbf{Y_0^k}$. $\mathbf{N}$ is the number of weighted particles which are set to $\frac{\mathbf{1}}{\mathbf{N}}$ initially and remains the same during state propagation. During an update, the weights are modified using Equation 2.68 under the assumption that the states correspond to Markov process and observations are independent.

$$\mathbf{w_k} = \mathbf{w_{k-1}} \frac{\mathbf{p}(\mathbf{y_k}|\mathbf{x_k})\mathbf{p}(\mathbf{x_k}|\mathbf{x_{k-1}})}{\mathbf{q}(\mathbf{x_k}|\mathbf{X_0^{k-1}}, \mathbf{Y_0^k})} \tag{2.68}$$

In Equation (2.68), $\mathbf{p}(\mathbf{x_k}|\mathbf{x_{k-1}})$ is the transition probability, $\mathbf{p}(\mathbf{y_k}|\mathbf{x_k})$ is the likelihood of making observations given the states and $\mathbf{q}(\mathbf{x_k}|\mathbf{X_0^{k-1}}, \mathbf{Y_0^k})$ is the proposal distribution where $\mathbf{X_0^{k-1}}$ and $\mathbf{Y_0^k}$ are the previous state history and all observations, respectively. This is the most critical design issue for successful particle filter. Taking samples from this density is impractical. Hence, the transition prior is the most

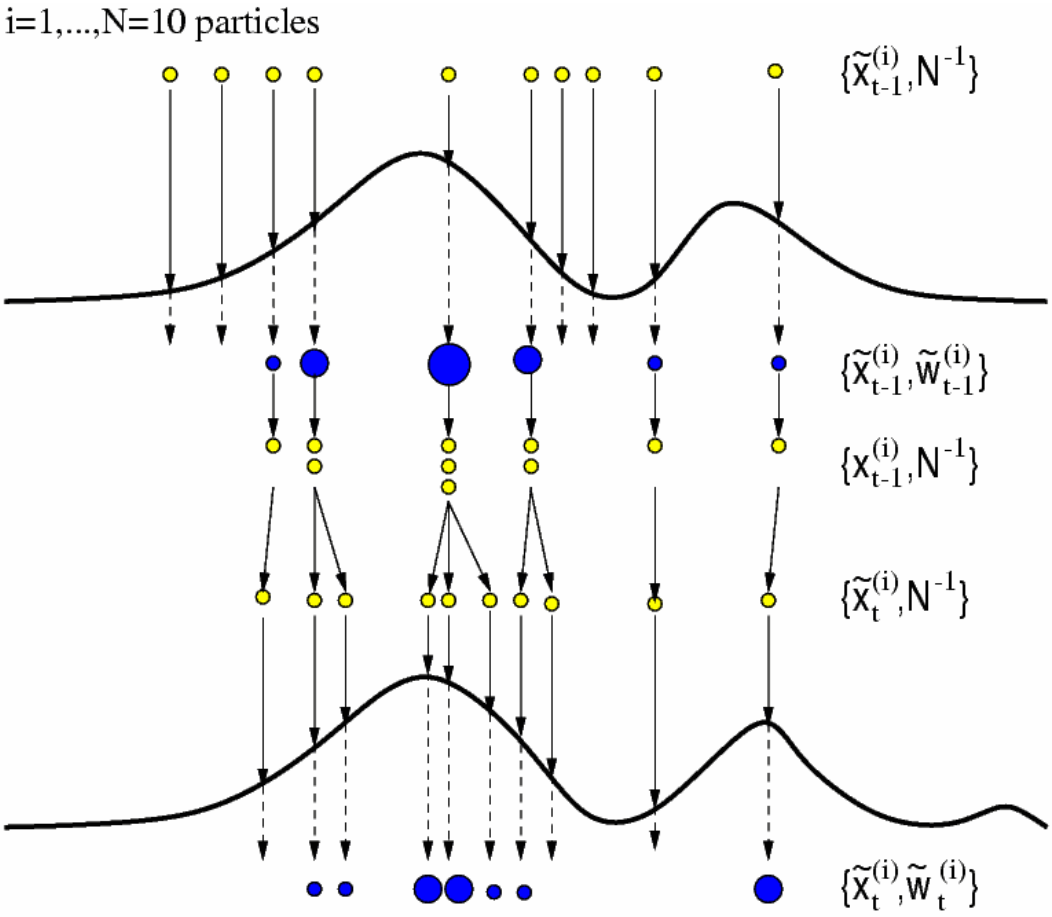

Figure 2.6: Particle Filter Algorithm. N number of particles are chosen with initial weights. Then, they are propagated through a non-linear function. Weights are modified during time update. Finally, particles are rearranged and weights are set to initial during resampling [12].

popular choice of proposal distribution:

$$\mathbf{q(x_k|X_0^{k-1}, Y_0^k) = p(x_k|x_{k-1})} \tag{2.69}$$

This makes the weighting update equation depend on the $\mathbf{p(y_k|x_k)}$ density such that $\mathbf{w_k = w_{k-1} p(y_k|x_k)}$. However, a particle starvation problem occurs here after a few iterations, since there will be numerically insignificant weights. This problem can be handled by using a resampling step. Otherwise, almost all of the particles' weights tend to zero. One resampling technique is called sampling importance resampling (SIR) and the steps are:

- Randomly replicate N number of particles proportional to their weights and generate a new set of $\mathbf{x_0(i^*)}$,
- Replace current particle set with $\mathbf{x_0(i^*)}$.

2.8.1 Particle Filter Algorithm. The implementation of particle filter algorithm has five steps as presented below:

1. *Initialize particles*,

   Generate N random samples of particles $x_0^{(i)}$ from the prior $p(x_0)$

2. *Compute the weights*,

   Use Equation 2.68 to evaluate the importance weights. $p(y_k|x_k)$ is crucial in modifying the weights

   Normalize the weights.

$$\mathbf{\tilde{w}_k^{(i)} = w_k^{(i)} \left(\sum_{j=1}^{N} w_k^{(j)} \right)^{-1}} \tag{2.70}$$

3. *Apply SIR steps for resampling*,

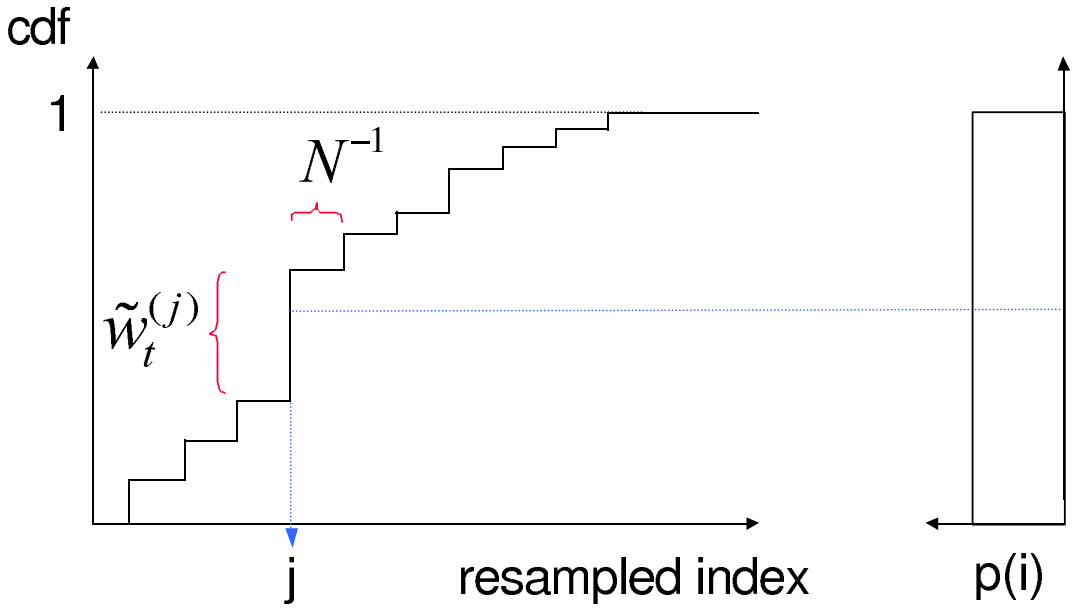

Figure 2.7: Resampling Algorithm. A new set of particles are generated randomly with respect to their weights [5].

This is to generate a new particle set to prevent particle starvation as shown in Figure 2.7

Reset weights to N^{-1}

4. *Compute the estimate*,

$$\hat{\mathbf{x}}_\mathbf{k} = \mathbf{E}(\mathbf{x}_\mathbf{k}|\mathbf{Y}_\mathbf{0}^\mathbf{k}) \simeq \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{x}_\mathbf{k}^{(\mathbf{i})} \tag{2.71}$$

5. *Redo from step 2 for next time intervals*.

More effective results can be achieved using true posterior density which is impractical to be used. Thus, proposal density should resemble the true posterior density as closely as possible for more effective results.

2.9 Approach Selection

There are a number of possible approaches to improve the robustness of previous research. In addition to the UKF, both particle filters (as mentioned) and neural

networks are also useful. For some reasons listed below, the unscented Kalman filter is considered as a better approach for current situation:

- it is easier to implement,
- uses same approximation issues as extended Kalman filter,
- contrary to extended Kalman filter, the unscented Kalman filter uses more than one point (sigma points) for estimation,
- sigma points allow better estimation of covariance,
- it allows us to propagate through at least second order linearization of nonlinear function.

III. Algorithm Development

The unscented Kalman filter algorithm is used in this research to recursively estimate the navigation state and associated errors. As in previous work, this method tracks the pixel location of stationary objects in an image-aided inertial system under a number of assumptions given in Chapter 1.

3.1 System Definition

While having the same approximation issues of the extended Kalman filter (e.g., Gaussian random variable), the unscented Kalman filter (UKF) uses a deterministic sampling approach (see Section 2.5.1) in which the state distribution is represented using a minimal set of carefully chosen sigma points around the mean. The main idea is to approximate the Gaussian distribution instead of an arbitrary nonlinear function. The nonlinear function is applied in propagation without losing the true mean and covariance of GRV. As seen in the block diagram of the system in Figure 3.1, an unscented Kalman filter is designed to estimate the errors in the calculated system parameters (see Table 3.1).

Table 3.1: System Parameter Definition

Parameter	Description
$\mathbf{p}^n$	Vehicle position in navigation frame (northing, easting, and down)
$\mathbf{v}^n$	Vehicle velocity in navigation frame (north, east, down)
$\mathbf{C}_b^n$	Vehicle body to navigation frame DCM
$\mathbf{a}^b$	Accelerometer bias vector
$\mathbf{b}^b$	Gyroscope bias vector
$\mathbf{t}_m^n$	Location of landmark m in the navigation frame (one for each landmark currently tracked)

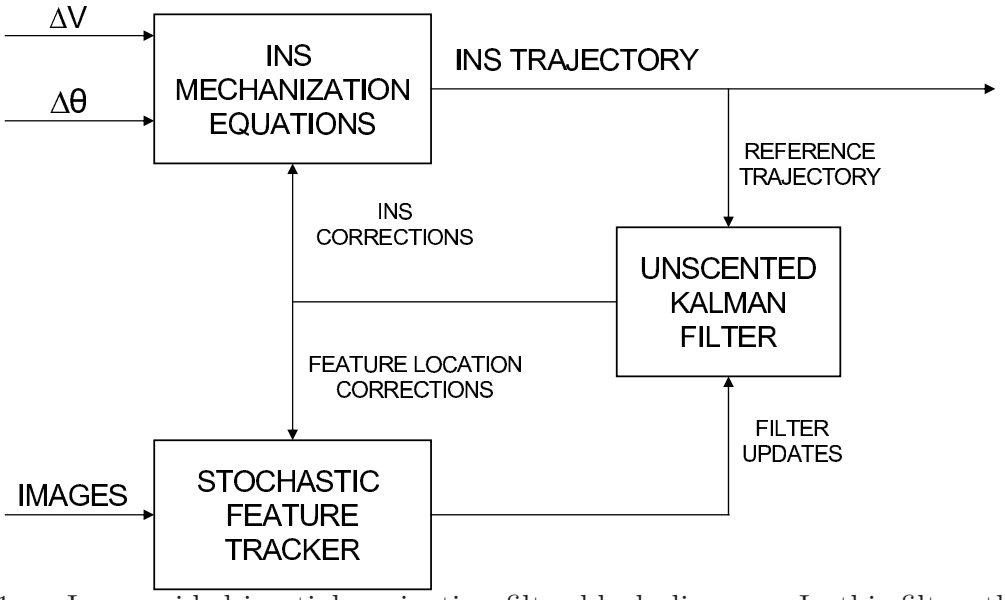

Figure 3.1: Image-aided inertial navigation filter block diagram. In this filter, the location of stationary objects are tracked and used to estimate and update the errors in an inertial navigation system. The inertial navigation system is, in turn, used to support the feature tracking loop [28].

3.2 Navigation State Structure

For the algorithm, a predefined structure is created to define current system parameters. Some of the parameters are listed below:

- $\mathbf{t_{gps}}$: is the GPS time
- $\mathbf{C_b^n}$: a Direction Cosine Matrix (DCM) used for transformation from body frame to navigation frame.
- $\mathbf{C_n^e}$: a DCM matrix used for transformation from navigation frame to Earth frame
- $\mathbf{p^n}$: current position of navigation structure based on *n-frame*
- $\mathbf{v^n}$: current velocity of navigation structure based on *n-frame*
- $\mathbf{P_x}$: covariance matrix of errors
- $\delta\mathbf{x}$: this is the current errors of navigation structure. It is a 15-state matrix including position error, velocity error, angular error, accelerometer bias and gyro bias errors in three dimensions, respectively.

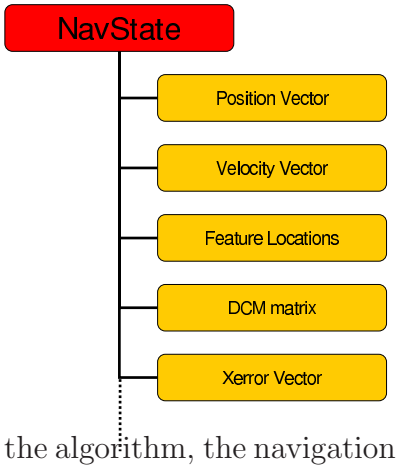

Figure 3.2: For the algorithm, the navigation state structure is created to define current system parameters. It also includes δx which is the vector form of all errors. Sigma points are generated using δx vector.

$$\boldsymbol{\delta x} = \begin{bmatrix} \delta p_x \\ \delta p_y \\ \delta p_z \\ - - - \\ \delta v_x \\ \delta v_y \\ \delta v_z \\ - - - \\ \delta \psi_x \\ \delta \psi_y \\ \delta \psi_z \\ - - - \\ \delta a_x \\ \delta a_y \\ \delta a_z \\ - - - \\ \delta b_x \\ \delta b_y \\ \delta b_z \end{bmatrix} \tag{3.1}$$

After each propagation and update cycle, a correction is applied to navigation state structure by subtracting $\delta\mathbf{x}$ from current system parameters.

3.3 Propagation of Navigation State Structure

After creating the navigation state structure, the whole valued states of navigation state structure can be propagated through strapdown mechanization equations as seen in block diagram (see Figure 3.1). It is desired to find expected $\delta\mathbf{x}$ in propagation cycle while calculating the whole valued states through mechanization.

3.3.1 Finding Errors from Propagated Navigation State Structure. The strapdown mechanization equations are nonlinear equations [24]. Thus, the unscented transformation is used to calculate the statistics of navigation state structure through nonlinear transformation. Before using a nonlinear function, navigation state structure state errors (dimension L) and covariance matrix are used to generate a matrix $\boldsymbol{\chi}$ of 2L+1 sigma points (see Section 2.5.1).

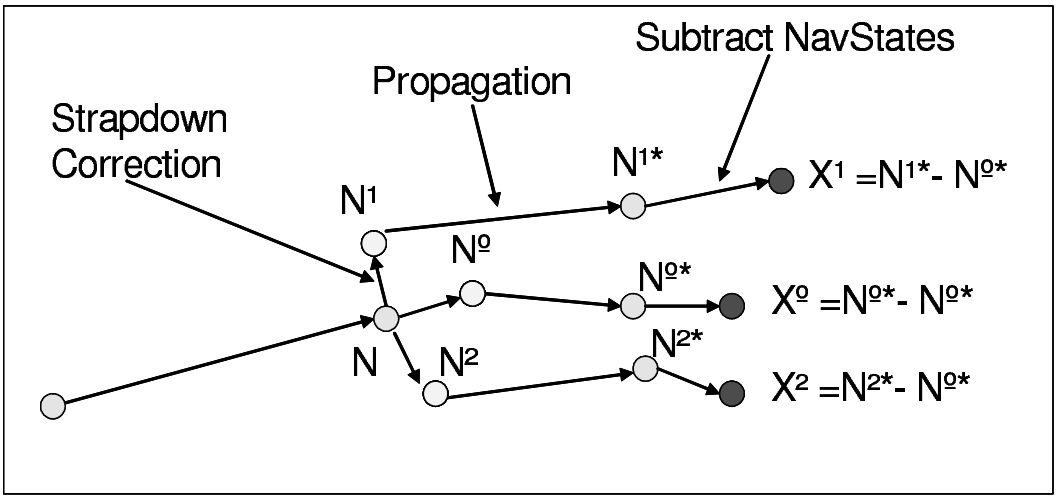

Figure 3.3: Whole-valued navigation state structures are generated using sigma points. These whole-valued states are then propagated through the strapdown mechanization algorithm. Finally, the differences between each predicted navigation state are found by comparing to the nominal, whole-valued navigation state.

While the spread of the sigma points is a function of the error distribution, the strapdown mechanization function propagates the whole-valued navigation state, not

the errors (see [20]). Hence, once the collection of sigma points $\boldsymbol{\chi}$ is computed about the nominal, each sigma point is transformed to and from whole-valued navigation state sigma points, $\mathbf{N_i}$, using a generalized differencing operator:

$$\mathbf{N_i} = \mathbf{N_0} \bigoplus \boldsymbol{\chi_i} \tag{3.2}$$

$$\boldsymbol{\chi_i} = \mathbf{N_i} \left[\bigoplus \right]^{-1} \mathbf{N_0} \tag{3.3}$$

Despite the fact that for most of navigation state structure parameters (e.g., position, velocity, etc.) the differencing operator is simply the standard vector addition operator, it uses a different approach to calculate small angular errors. Differencing orientation states, namely the $\mathbf{C_n^b}$ direction cosine matrix, are based on the notion that small angular changes can be appropriately represented by a simple rotation error vector ($\boldsymbol{\psi}$ in the above equations), whereas this is not true for general orientations. This property is exploited in many navigation state error models such as the well-known Pinson error model, which represents angular errors as a three-dimensional error vector about some nominal orientation DCM or quaternion. This method is both elegant and efficient.

The relative navigation state to error model equations are:

$$\delta p_i^n = p_i^n - p_0^n \tag{3.4}$$

$$\delta v_i^n = v_i^n - v_0^n \tag{3.5}$$

$$C_b^{n_i} = (I - \boldsymbol{\psi}_i \times) C_b^{n_0} \tag{3.6}$$

$$\delta a_i^n = a_i^n - a_0^n \tag{3.7}$$

$$\delta b_i^n = b_i^n - b_0^n \tag{3.8}$$

$$\delta t_i^n = t_i^n - t_0^n \tag{3.9}$$

where $\boldsymbol{\psi_i}$ is the angular error:

$$\boldsymbol{\psi_i} = \begin{bmatrix} \psi_{n_i} \\ \psi_{e_i} \\ \psi_{d_i} \end{bmatrix} \tag{3.10}$$

which corresponds to the *i-th* sigma point.

Thus, given a collection of angular difference sigma points, the whole-valued body-to-navigation frame DCMs can be calculated using the following steps:

- Convert the angular errors to the equivalent direction cosine matrix [24]. This represents the orientation error from the estimated navigation frame to the nominal navigation frame.
- Multiply this DCM with the nominal body-to-navigation frame DCM.

Applying these steps results in the whole-valued body-to-navigation frame DCM sigma points, represented by $\mathbf{C_b^{n_i}}$.

Up to now, the equivalent whole-valued states are calculated using sigma points and navigation state structures are generated. Next, these navigation state structures are propagated through the dynamics model, which is in this case the strapdown mechanization equation:

$$\mathbf{N_i(t_k)} = \mathbf{f}[\mathbf{N_i(t_{k-1})}, \tilde{\mathbf{u}}_\mathbf{k}, \mathbf{w_k}] \tag{3.11}$$

where $\tilde{\mathbf{u}}_\mathbf{k}$ are the inertial measurements and $\mathbf{w_k}$ is the dynamics noise vector. This function generates a collection of propagated navigation state structures at the time of the next image (see Figure 3.3).

3.3.2 Calculating Statistics of Navigation State. Given this collection of whole-valued navigation state sigma points, the representative statistics (i.e., mean and covariance) can be calculated. Before this step, the error sigma points must be calculated.

The error sigma points can be calculated by subtracting each navigation state structure about the nominal navigation state structure, which is in this case the one that is propagated using the initial sigma point (the first vector of $\boldsymbol{\chi}$). Finding the difference between each predicted whole-valued navigation state sigma point about the nominal navigation state sigma point is shown in the approach outlined in Equations (3.3) and (3.4). Once the representative error sigma points are determined, the error state mean and covariance are approximated using the weighted average and weighted outer product

$$\hat{\mathbf{x}}_{\mathbf{k}}^{-} = \sum_{i=0}^{2L} \boldsymbol{W}_i^{(m)} \boldsymbol{\chi}_{i,k|k-1}^{*} \tag{3.12}$$

$$\mathbf{P}_{\mathbf{k}}^{-} = \sum_{i=0}^{2L} \boldsymbol{W}_i^{(c)} (\boldsymbol{\chi}_{i,k|k-1}^{*} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-})(\boldsymbol{\chi}_{i,k|k-1}^{*} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-})^{\boldsymbol{T}} + \boldsymbol{Q_d} \tag{3.13}$$

where $\mathbf{Q_d}$ is the process noise. The weights are used as given in Chapter 2.

3.4 Update of Navigation State Structure

As in propagation cycle, update equations are also nonlinear functions. In this case, unscented transformation is required to be used for update cycle as well. Initially, new navigation states, following new sigma points, are calculated depending on propagated navigation state structure and error covariance matrix. Then, these structures are used in update cycle as described in following sections.

3.4.1 The Tracking Algorithm. The general concept for the track algorithm is to find landmark tracks which can provide the best navigation information to the filter.

There are three issues related to this algorithm:

- Track selection: The idea is to find landmarks that can be easily and accurately tracked for a long period of time. This implies choosing features that are strong

and well separated in image space. Due to computer restrictions, limited number of landmarks can be tracked. Thus, stale landmarks are pruned by replacing with new tracks.

- Track addition: After identifying the new track, track addition is made with an estimate of location and uncertainty.
- Track deletion: Tracks which have not been successfully updated are identified for replacement.

As described above, the tracking algorithm predicts pixel locations and uncertainty for each navigation state including correction for optical distortion. These are the measurements to be used.

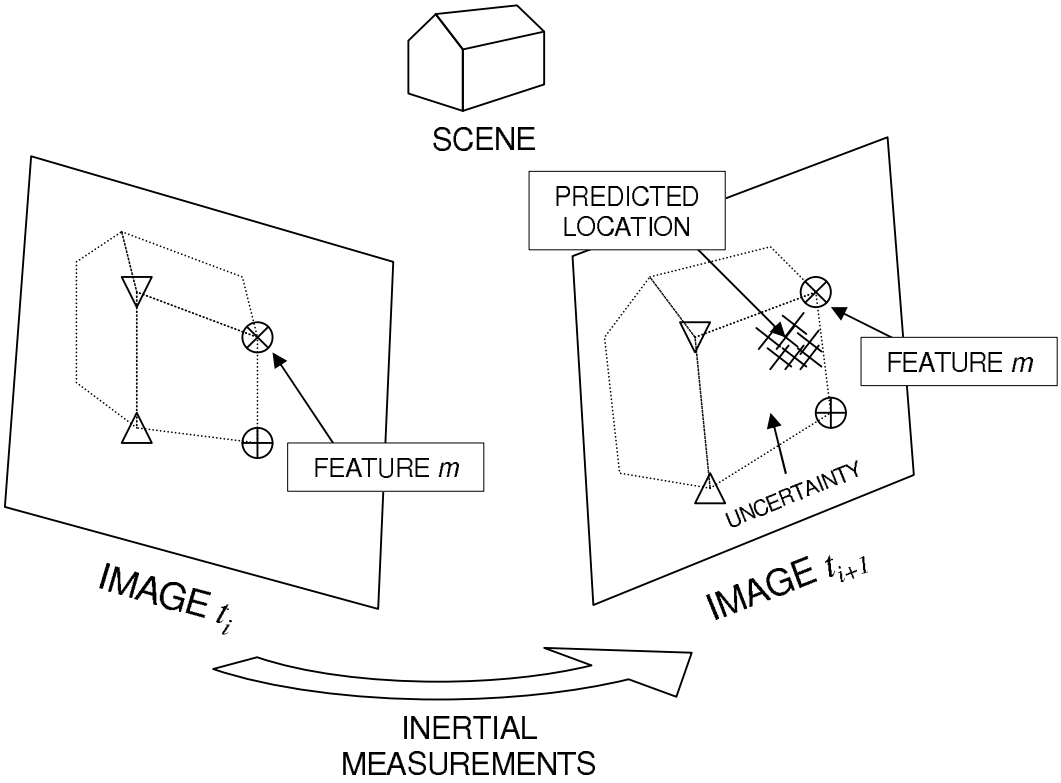

Figure 3.4: Stochastic feature projection. Optical features of interest are projected into future images using inertial measurements and stochastic projections [27].

3.4.2 Prediction of Pixel Location. The propagated whole-valued navigation state sigma points can now be used to predict the measurement sigma points using

the measurement equation:

$$\mathbf{z_i}(\mathbf{t_k}) = \mathbf{h}[\mathbf{N_i}(\mathbf{t_k}), \mathbf{v_k}] \tag{3.14}$$

where $\mathbf{z_i}(\mathbf{t_k})$ is the collection of predicted feature space locations corresponding to the currently tracked landmarks. An illustration of this prediction is shown in Figure 3.4.

New collection of navigation state structures are used to predict pixel locations by using tracking algorithm including correction for optical distortion [27]. The means of predictions are optical measurements. Calculating the statistics of the prediction is accomplished in a similar manner as with the navigation errors. The relevant statistics are calculated using the following weighted sum:

$$\hat{\mathbf{z}}_\mathbf{k} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{m})} \mathbf{z_i}(\mathbf{t_k}) \tag{3.15}$$

$$\mathbf{P}_{\hat{\mathbf{z}}_\mathbf{k}\hat{\mathbf{z}}_\mathbf{k}} = \sum_{\mathbf{i=0}}^{\mathbf{2L}} \mathbf{W}_\mathbf{i}^{(\mathbf{c})} (\mathbf{z_i}(\mathbf{t_k}) - \hat{\mathbf{z}}_\mathbf{k}^-)(\mathbf{z_i}(\mathbf{t_k}) - \hat{\mathbf{z}}_\mathbf{k}^-)^\mathbf{T} + \mathbf{R^n} \tag{3.16}$$

where $\mathbf{z_i}(\mathbf{t_k})$ is the matrix of predicted feature locations at time $\mathbf{t_k}$ and $\mathbf{R^n}$ is the measurement noise.

3.4.3 Measurement Update. The navigation state is updated after calculating the statistics of measurements. The measurement-update is applied using the measurement covariance and cross correlation matrices.

$$\mathbf{P}_{\mathbf{x_k},\mathbf{z_k}} = \sum_{\boldsymbol{i=0}}^{\boldsymbol{2L}} \boldsymbol{W}_i^{(c)} (\boldsymbol{\chi}_{i,k|k-1} - \hat{\boldsymbol{x}}_{\boldsymbol{k}}^-)(\mathbf{z_i}(\mathbf{t_k}) - \hat{\mathbf{z}}_\mathbf{k}^-)^\mathbf{T} \tag{3.17}$$

$$\mathbf{K_k} = \mathbf{P}_{\mathbf{x_k},\mathbf{z_k}} \mathbf{P}^{-\mathbf{1}}_{\hat{\mathbf{z}}_\mathbf{k}\hat{\mathbf{z}}_\mathbf{k}} \tag{3.18}$$

$$\hat{\mathbf{x}}_\mathbf{k}^+ = \hat{\mathbf{x}}_\mathbf{k}^- + \mathbf{K_k}(\tilde{\mathbf{z}}(\mathbf{t_k}) - \hat{\mathbf{z}}_\mathbf{k}^-) \tag{3.19}$$

$$\mathbf{P_k} = \mathbf{P}_\mathbf{k}^- - \mathbf{K_k} \mathbf{P}_{\hat{\mathbf{z}}_\mathbf{k}\hat{\mathbf{z}}_\mathbf{k}} \mathbf{K}_\mathbf{k}^\mathbf{T} \tag{3.20}$$

Finally, the estimated error is removed from the nominal navigation state and the update is completed.

A conceptual summary of the propagation and measurement cycles for the unscented Kalman filter is as follows:

- Based on the current nominal navigation state estimate, calculate a collection of whole-valued navigation states corresponding to the calculated sigma points.
- Propagate the set of whole-valued navigation state sigma points through the nonlinear strapdown mechanization function.
- Calculate the pre-measurement statistics using the weighted sum of the differences between whole-valued navigation states.
- Predict the feature locations of the current landmark tracks.
- Determine a statistical correspondence between the predicted and measured feature locations.
- Calculate the statistics of the predicted feature location errors. Use the measurement residual and Kalman gain to correct the nominal navigation state.
- Repeat as necessary.

3.5 Decreasing The Computational Cost Using Quaternion

Although the developed algorithm works as expected, the need for a better performance is inevitable due to the complexity and slowness. The main reason for the complexity and slowness is definitely due to the direction cosine matrices, since it is difficult to propagate and update all sigma points at once as explained below.

In the EKF algorithm, euler angles are to be calculated from DCM matrices in each propagation and update cycle. Contrary to UKF algorithm, there was only one point (the mean) to be propagated and updated in recent work. Hence, there was no need to replace DCM. In the UKF algorithm, using the same technique requires repeating propagation and update cycle for each sigma point to move next time step.

One way to make UKF algorithm faster is to propagate and update all sigma points at once. This requires using matrices for all whole state valued vectors since the strapdown mechanization algorithm uses whole state values as mentioned previously. All whole state valued vectors of navigation state structure can be converted to matrices using sigma points in order to propagate every point at once. But, the DCM will continue to increase complexity due to the fact that it is already a matrix and generating an array of DCM matrices will not help much. The computational cost remains as long as DCM is used in new algorithm.

Quaternions are very concise way of applying rotation compared to direction cosine matrices. The following properties are the reasons for its popularity:

- It is only a vector, thus compactly supporting multiple particles,
- It is more compact than the DCM representation and less susceptible to round-off errors,
- Expression of the DCM in terms of quaternion parameters involves no trigonometric functions,
- Using a quaternion product, two individual rotation can be simply combined.

In order to make sure that they correspond to valid rotations, quaternions should be normalized due to rounding errors, as well. However, the computational cost of normalizing a quaternion, is much less than normalizing a 3x3 DCM [8].

IV. Results

As in the previous work, the UKF-based imaging and inertial fusion navigation algorithm is evaluated using both simulated and experimental ground profiles. The profiles are designed to provide a range of image types in order to exercise the feature tracking algorithm. The data collection, simulation and experiment results are presented in the next section.

4.1 Data Collection

This research is based on the principles of previous one. Hence, the data collection system that consisted of a consumer grade MEMS IMU and two digital cameras (see Figure 4.1) and all data used for simulation and experiment are same as used in EKF based work. That makes for easy comparison of results.

In order to validate the development, both simulation and experiments are used for evaluation of the algorithm. Initially, this research is evaluated using a simulation. The simulation is an artificial environment which does not include any unknown errors, which makes it easier for the algorithm to succeed. In the experimental results, the algorithm is evaluated in a real environment where we might encounter unknown errors. The simulation and results are presented in the next section.

4.2 Simulation And Results

The algorithm was tested using a Monte Carlo simulation of a standard indoor profile. The profile consisted of a straight corridor, designed to be similar to the indoor experimental data collection.

An accurate simulation of the navigation environment requires simulating the performance of the sensors in response to a true motion trajectory. The trajectory was generated using ProfGen version 8.19 software package [14]. For each Monte Carlo navigation simulation run, the inertial sensor measurements are regenerated using the true trajectory and an inertial sensor error model.

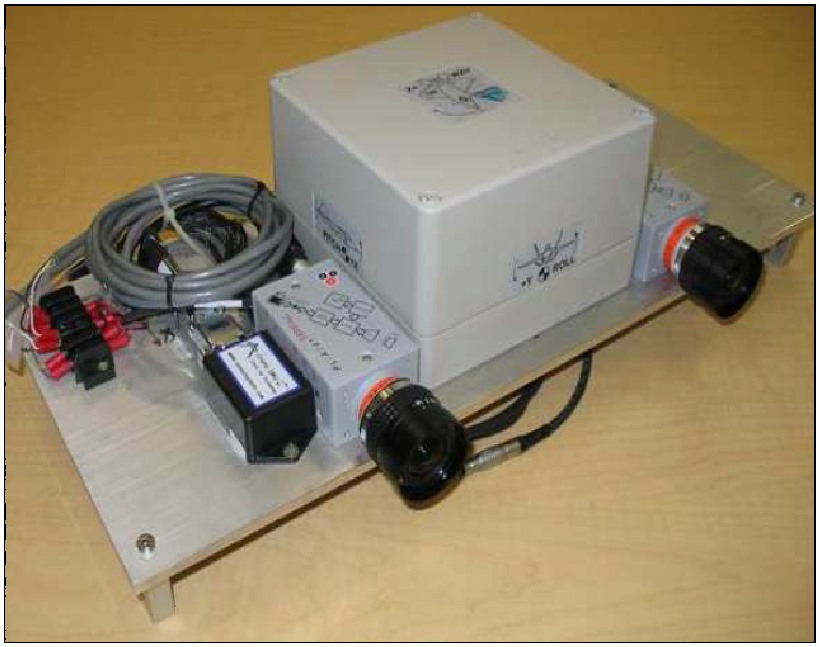

Figure 4.1: Data collection system. The data collection system consisted of a consumer-grade MEMS IMU and monochrome digital cameras. Although not used in this research, a tactical-grade IMU was co-mounted on the platform in order to provide a performance comparison between different grades of inertial sensors.

4.2.1 Simulation Environment. Due to the inherent complexity of the optical environment, it is beyond the scope of this thesis to generate simulated images. Instead, a simulated feature set was created by randomly distributing features along a corridor surrounding the true trajectory. The features are each given random descriptor vectors in order to exercise the feature tracking algorithm. While this optical simulation method is appropriate for testing the image and inertial fusion algorithm, the results are not directly comparable to the real system performance, because imaging issues such as lighting conditions, motion blur, and affine changes in the feature descriptor due to pose changes are not modelled. The simulation model is generated to verify that the algorithm is working properly. It is expected that these results will be optimistic with respect to position error. Nonetheless, the error divergence trends should be observable. Once simulation gives good results, then the algorithm will be ready to be evaluated using real data.

The simulated corridor was 3 meters wide, 3 meters high, and approximately 300 meters long. Features were randomly generated on the walls, floor and ceiling of the corridor with an average spacing of 0.25 features per square meter. Each feature was given a random primary length and orientation, which, combined with the true pose of the sensor, resulted in accurately simulated scale and orientation parameters in feature space.

Before accelerating the platform, an initial 60-second stationary alignment is made. Then, the sensor platform accelerated to 0.5 meters per second and maintained this velocity until the end of the corridor. Finally, it is decelerated to a stop at the end. The platform remained stationary for 60 seconds after coming to a stop. This resulted in a 660-second image and inertial navigation profile. Simulated images are collected at 2 Hz.

4.2.2 Simulation Test. A Monte Carlo simulation was conducted using an inertial sensor model representing the Crista consumer-grade IMU [23]. Each simulation consisted of 60 runs, each with randomly generated inertial measurement

Figure 4.2: Sample image from indoor data collection. The indoor data collection presents the filter with man-made features in an office environment. The crosses and ellipses indicate the locations and uncertainty of currently tracked landmarks [28].

errors due to random walks and sensor bias. In order to mitigate any potential effects due to the location of the features in the simulated environment, the feature locations and descriptors were randomly generated every 20 runs.

Previously, consistent divergence during our long-duration Monte-Carlo simulations occurred since the state distribution was propagated through a first order linearization. The simulated position errors for the EKF and UKF are shown in Figures 4.3 and 4.4, respectively. Deleterious effects of EKF can be seen in Figure 4.3 compared to UKF.

Significant excursions in position are noted in the EKF-based algorithm, which is evidence of the effects of increased attitude errors resulting in destabilizing linearization errors. In contrast, the UKF-based estimator appears to eliminate the departures and is reasonably consistent with the estimated uncertainty. The UKF estimate does, however, appear to be biased. A tuning is applied to the algorithm by changing the $\boldsymbol{\alpha}$ parameter (see Section 2.5.1) in a range of 1 to 10^{-4}. But, the bias remained unchanged. This is not completely unexpected, as the unscented transformation can be shown to produce a biased estimate under non-symmetric nonlinearities. In either case, the effects are relatively small and should be in the noise when processing real data sets.

The reason for biased estimate will be more clear by investigating the velocity and attitude error plots that are given in Figures 4.5 and 4.6, respectively. In the first plot, the velocity errors appear to be very consistent and stable. But, that is not same for following plot. The attitude errors show a different story, especially in heading error. The obvious heading error bias explains the resulting position error bias. Unfortunately, the cause for this heading error bias is unknown and will require further investigation. As mentioned previously, the apparent stability of the UKF-based algorithm should outweigh the effects of small heading bias for real data set.

In the next section, the experimental data collection profile and results are presented and a comparable figure between the EKF and UKF algorithms is shown.

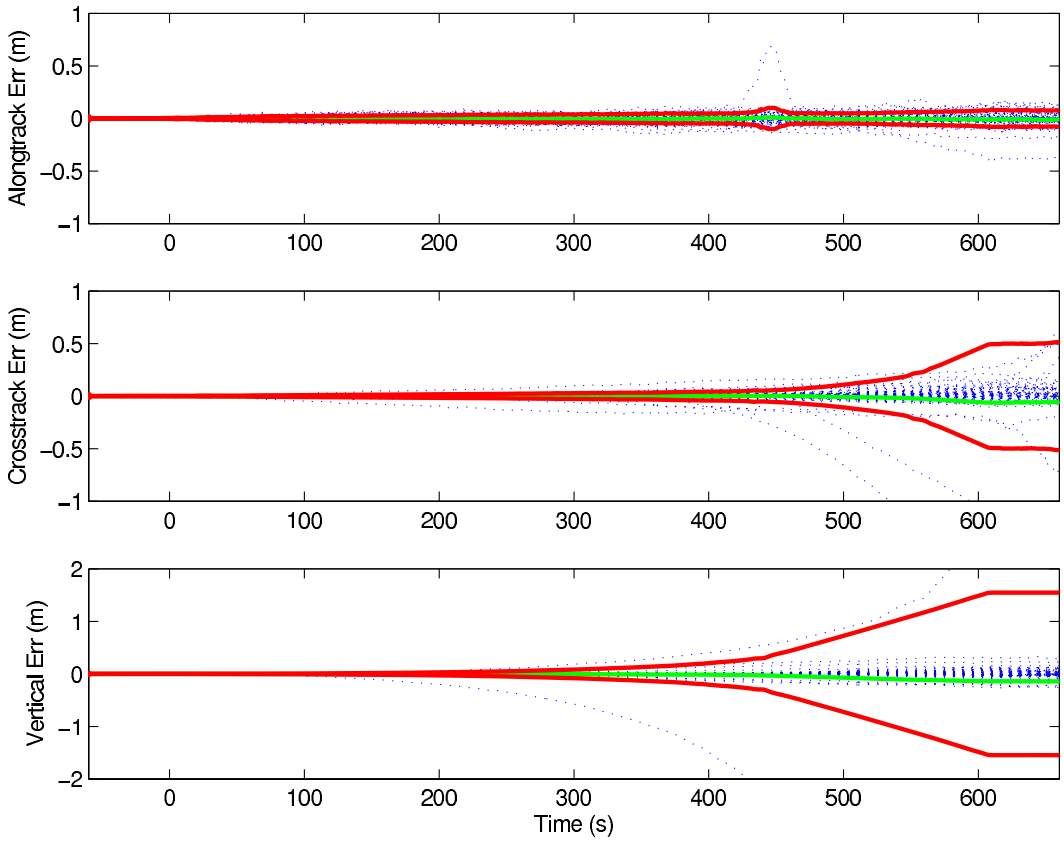

Figure 4.3: Simulated 60-run Monte Carlo position error results for indoor profile with a consumer-grade inertial sensor using an EKF-based image aiding algorithm. The extended Kalman filter algorithm displays a tendency toward divergence due to the cumulative effects of linearization errors.

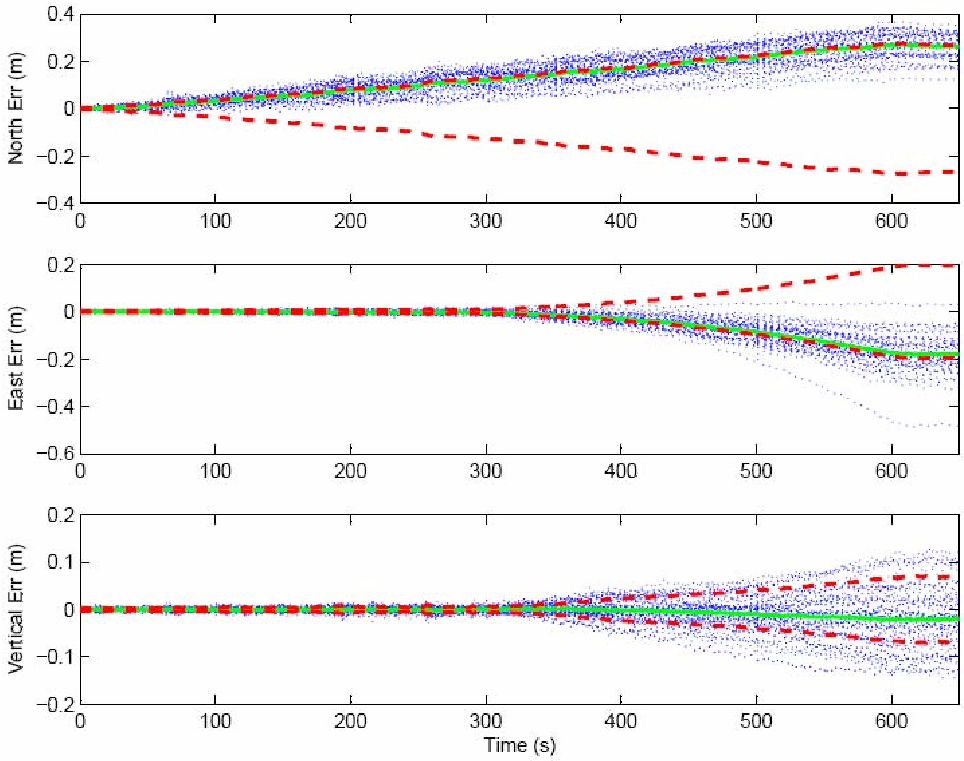

Figure 4.4: Simulated 60-run Monte Carlo position error results for indoor profile with a consumer-grade inertial sensor using a UKF-based image aiding algorithm. The UKF based algorithm shows no indication of rapid divergence, although the estimate appears to contain a bias.

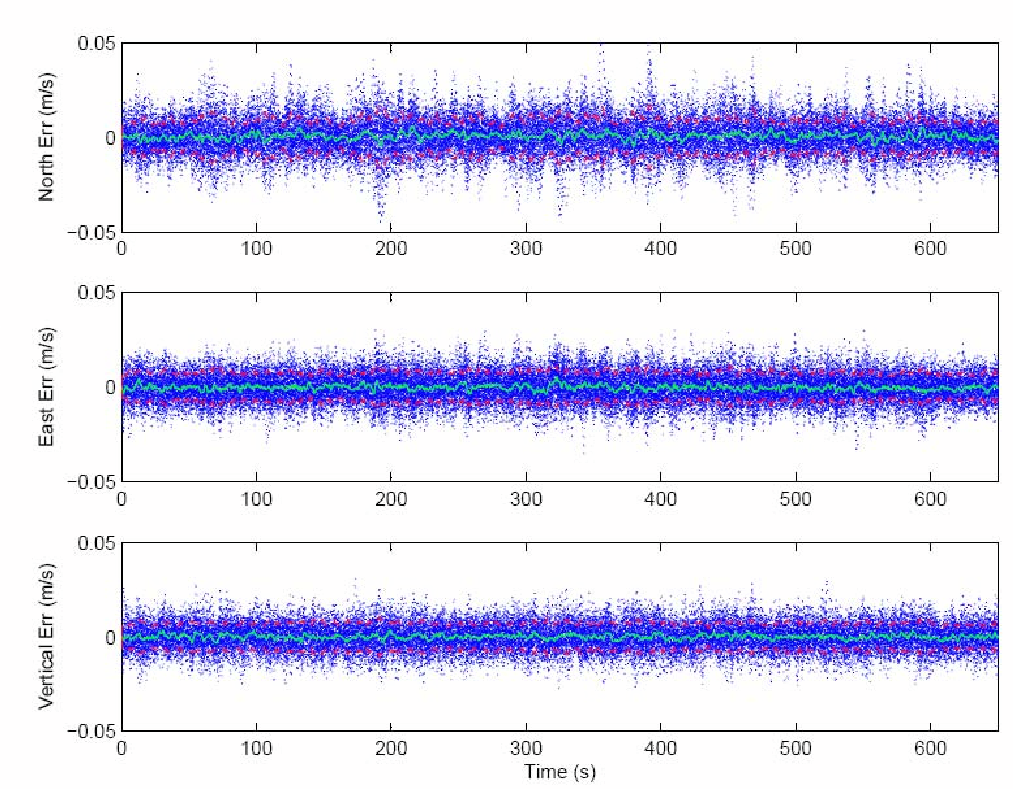

Figure 4.5: Simulated 60-run Monte Carlo velocity error results for indoor profile with a consumer-grade inertial sensor using a UKF-based image aiding algorithm. As expected, the velocity estimates appear consistent and stable.

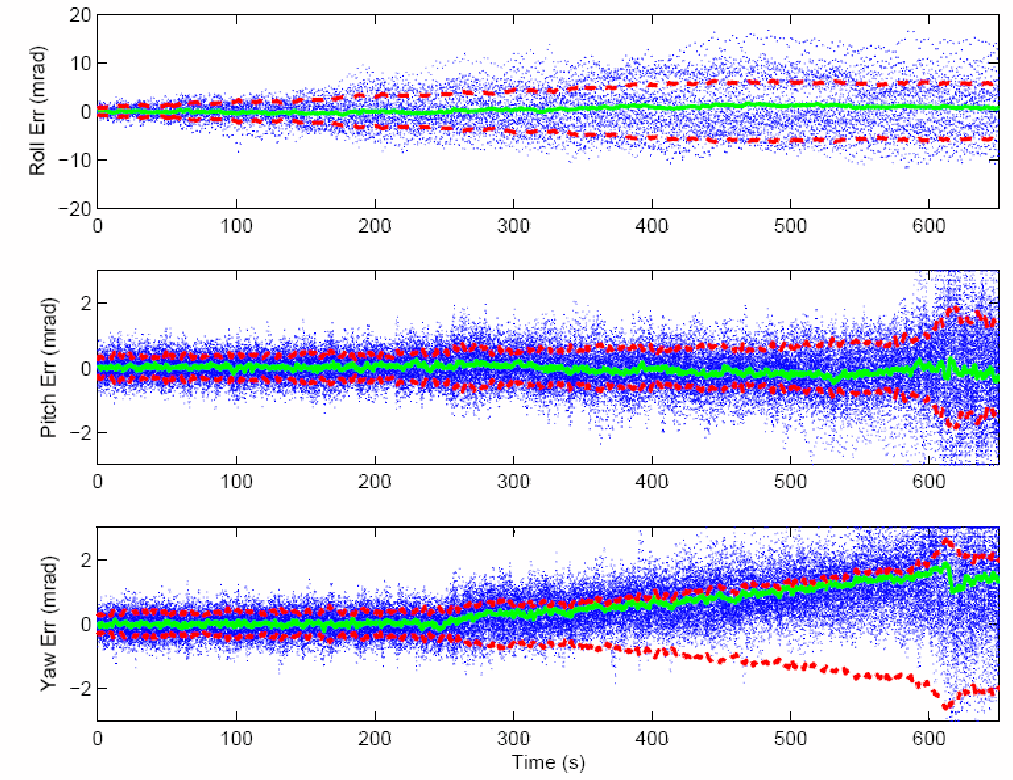

Figure 4.6: Simulated 60-run Monte Carlo attitude error results for indoor profile with a consumer-grade inertial sensor using a UKF-based image aiding algorithm. The UKF-based attitude estimates appear to be relatively stable and consistent. The source of the heading bias is unknown, however this is most likely the root cause of the position error bias.

4.3 Experiment And Results

The evaluation of the simulation results validated the algorithm. Next, a real data set is to be used to verify that the algorithm works properly in the experiment also. As used previously, the algorithm was tested experimentally using a closed-loop ground navigation profile designed to examine the operation of the feature tracking system in a real-world environment and compare the performance between the EKF and UKF implementations. The profile consisted of a closed path in an indoor environment. The path began and ended at the same location.

Similar to the simulation, the data collection began with a 10-minute stationary alignment period. After the alignment period, the sensor was moved in a 10- minute loop around the hallways of the building. No prior knowledge was provided to the algorithm regarding the location of features or structure of the environment. A sample image from the indoor profile is shown in Figure 4.2.

The indoor profile presents the algorithm with different challenges from a feature tracking perspective. The indoor environment consists of repetitive, visually identical features (e.g., floor tiles, lights, etc.), which can easily cause confusion for the feature tracking algorithm. In addition, reflections from windows and other shiny surfaces might not be interpreted properly by the filter and could potentially result in navigation errors. Finally, the lower light intensity levels and large areas with poor contrast (e.g., smooth, featureless walls) presents a relatively stark feature space.

Both filters' estimates of the trajectories are overlayed on a floor plan of the building in Figure 4.7 for the EKF and UKF algorithms. Clearly, the EKF and UKF filters perform well within a range of 3 meter. For both EKF and UKF algorithms, the estimated trajectory generally corresponds to the buildings hallways. While additional testing is required to fully characterize the performance of the algorithms, the navigation accuracy achieved in a real-world environment indicates promise for the UKF based image-aided inertial navigation system.

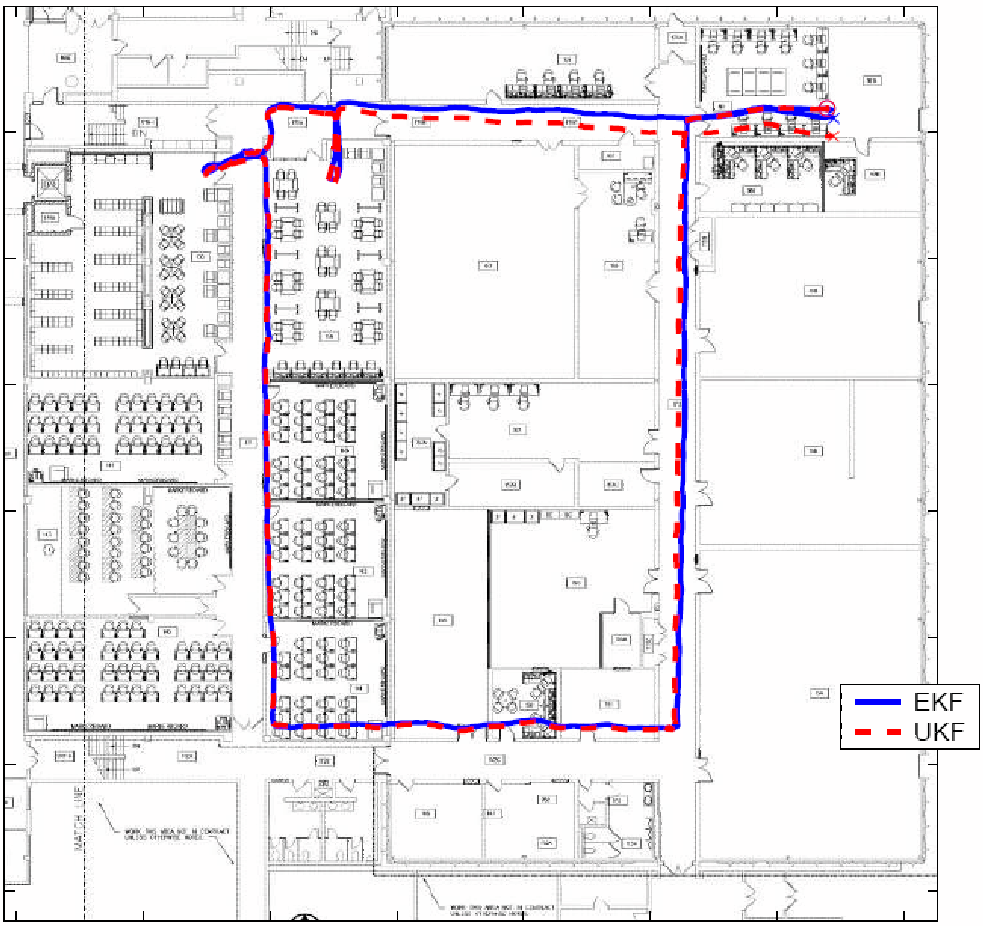

Figure 4.7: Estimated trajectory for the extended Kalman filter (blue) and unscented Kalman filter (red) image-aided inertial algorithm. Both algorithms demonstrate similar performance, within the expected uncertainty of the position state estimate.

4.4 Effect of Quaternion On Algorithm

As discussed in Section 3.5, the purpose of quaternion is to replace DCM with a simple vector. Hence, it leads to use all sigma points for propagation and update cycles at once, consequently, decrease the computational cost of the algorithm. Since it has less round-off errors, it is also desired to have less error than that caused by a DCM rotation.

Although not given in Table 4.1, propagating and updating all sigma points at once using an array of DCM (see Section 3.5) were also applied to the algorithm. The timing results were little faster compared to DCM-Based UKF Algorithm results. But, applying non-linear function to a set of DCM matrices at once was still too slow and required further improvement on the algorithm. That is why it is replaced with quaternion to get rid of arrays. The results are generated using a Windows XP operating system with 2 GB DDR2 RAM and Matlab 2007b software. The type of computer processor was Intel Centrino Duo and the speed of the processor was 1.83 Ghz. It is also noted that Matlab software works faster using matrices while it might not be same for all other programming languages. The following results are attained:

Table 4.1: Time Comparison

Simulation Time	EKF Algorithm	DCM-Based UKF Algorithm	Quaternion-Based UKF Algorithm
10 sec.	30 sec.	75 sec.	34 sec.
30 sec.	93 sec.	210 sec.	105 sec.
60 sec.	178 sec.	410 sec.	198 sec.
100 sec.	302 sec.	697 sec.	334 sec.
650 sec.	1895 sec.	Almost 7200 sec.	2250 sec.

As expected, quaternion allowed more elegant way to represent sigma points and that decreased the computational cost and spending time for both simulation and experiments (see Table 4.1). The fact that EKF algorithm is propagating and updating only the mean through the nonlinear function while UKF algorithm propagates a set

of sigma points (almost 100 sigma points propagated and updated) causes EKF algorithm the fastest one. The results of quaternion-based UKF algorithm are also similar to EKF results while DCM based UKF algorithm shows a very slow performance.

V. Conclusions

In this research, a new approach is presented for fusing optical and inertial sensors to be used in environments where GPS signals are not available. Previous research presented a statistically rigorous method to tightly couple imaging and inertial sensors using an extended Kalman filter. Unfortunately, the estimator demonstrated divergent characteristics during longer-term navigation profile which was attributed to the cumulative destabilizing effects of linearization errors. To address this known weakness of the extended Kalman filter, an image-aided navigation algorithm based on the unscented Kalman filter was designed.

The most significant conclusion for this work was reducing the level of divergence. The filter was evaluated and compared using a combination of simulated and experimental data. During the evaluations, the stability of the unscented Kalman filter was shown to significantly outperform the extended Kalman filter. In addition, the unscented Kalman filter maintained an accurate and consistent position error estimate.

Besides reducing the level of divergence, it was also desired to maintain the efficiency of the EKF. Unfortunately, the unscented Kalman filter was not efficient. Thus, a change was needed to make the algorithm faster. The UKF propagates and updates all sigma points to move next image time while the EKF makes it only for one point (the mean). Instead using one sigma point each time, it was intended to use all sigma points at once. The only obstacle was DCM which was replaced with quaternion to represent rotation with a single vector. Consequently, the optimized UKF algorithm's speed was comparable to the standard EKF algorithm.

5.1 Future Work

In addition to this research, there other issues to exploit the synergy of imaging and inertial sensors. Besides, there is a potential drawback of the UKF for this application to be investigated. The presence of a systematic bias in the position and

attitude estimates is a matter to be solved. Unfortunately, the cause for this heading error bias is unknown and will require further investigation.

There are potential research areas to explore in a deeper level. One potential research area is the particle filter. The particle filter is another technique to exploit the synergy of imaging and inertial sensors. As described in Chapter 2, it is also a non-linear estimation algorithm based on a sequential Monte Carlo method. What is more, there is no need to make a Gaussian assumption.

Once a particle filter is completed, an unscented particle filter, which is a combination of both particle and unscented filters, will be a good approach to explore a deeper level.

Bibliography

1. Bir Bhanu, Barry Roberts and John Ming. "Inertial Navigation Sensor Integrated Motion Analysis For Obstacle Detection". *Proceedings Of The IEEE*, 1990.

2. Brown, Alison and Yan Lu. "Performance Test Results of an Integrated GPS/MEMS Inertial Navigation Package". *Proceedings of ION GNSS 2004*, 825–832. September 2004.

3. Brown, Robert Grover and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons, Inc., New York, NY, 1992.

4. Committee, DMA WGS-84 Development. "Department of Defense World Geodetic System 1984 - Its Definition and Relationships with Local Geodetic Systems". Technical Report 8350.2, Defense Mapping Agency, Washington DC, September 1987.

5. Haykin, Simon. *Kalman Filtering and Neural Networks.* JohnWiley and Sons, Inc., New York, NY, 2001.

6. Hogan, Guy T. "Quaternions and Rotations in 3-Space", September 2001. Norfolk State University.

7. Hrabar, Stefan and Gaurav S. Sukhatme. "A Comparison of Two Camera Configurations for Optic-Flow Based Navigation of a UAV Through Urban Canyons". *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, 2673–2680. September 2004.

8. Janez Funda, Richard P. Paul. "A Comparison of Transforms and Quaternions". *Proceedings Of The IEEE*, April 1988.

9. Liebe, C.C. "Star trackers for attitude determination". *Aerospace and Electronic Systems Magazine, IEEE*, 10(6):10–16, June 1995.

10. Maybeck, Peter S. *Stochastic Models Estimation And Control*, volume I. Academic Press, Inc. Orlando, Florida, 1979.

11. McMahon, Joe. "Linear Algebraic Introduction to Quaternions", 2003. University of Arizona.

12. R. van der Merwe, A. Doucet, N. de Freitas and E. Wan. "The Unscented Particle Filter". *Cambridge University Engineering Department, Cambridge, England*, 46, Aug 2000.

13. Morris M. Kuritsky, Murray S. Goldstein. "Inertial Navigation". *Proceedings Of The IEEE*, 71(10), 1983.

14. Musick, S. "Profgen: PC Software for Trajectory Generation". *Software Package v8.19, Air Force Research Laboratory,Wright-Patterson Air Force Base*, 2004.

15. Pachter, M. "INS Aiding By Trecking An Unknown Ground Object". *Proceedings of the American Control Conference*, 2003.

16. Pissavin, P., J.P. Krebs, P. LeLong, Ph. Vidal, and R. Navoni. "Improved Star Tracker for ODIN Satellite". *Proceedings of ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, 611–616. November 1997.

17. Purll, David, Norbert Gradmann, and Manfred Bollner. "The ROSAT Star Tracker: Flight Experience". *ESA, Spacecraft Guidance, Navigation and Control Systems*, N92-2443215-18:551–556, 1991.

18. Raquet, John F. and Michael Giebner. "Navigation Using Optical Measurements of Objects at Unknown Locations". *Proceedings of the 59th Annual Meeting of the Institute of Navigation*, 282–290. June 2003.

19. Raquet, John F., Michael Giebner, and Michael J. Veth. "Navigation Using Optical Measurements of Objects at Unknown Locations". Air Force Institute of Technology, December 2004.

20. Shin, E. and N. El-Sheimy. "Unscented Kalman Filter and Attitude Errors of Low-Cost Inertial Navigation Systems". *Journal of the Institute of Navigation*, pp. 1–9, 2007.

21. Stergios I. Roumeliotis, Andrew E. Johnson and James F. Montgomery. "Augmenting Inertial Navigation With Image-Based Motion Estimation". *Proceedings Of The IEEE*, 4, 2002.

22. Strelow, Dennis and Sanjiv Singh. "Optimal Motion Estimation from Visual and Inertial Measurements". *Proceedings of the Workshop on Applications of Computer Vision*. December 2002.

23. Technology., Cloud Cap. "Crista Inertial Measurement Unit (IMU) Interface / Operation Document", May 2004. URL: http://www.coudcaptech.com/.

24. Titterton, D.H. and J.L. Weston. *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd., Lavenham, United Kingdom, 1997.

25. Veth, Michael J. "Fusion Of Imaging And Inertial Sensors For Navigation", September 2006. Air Force Institute of Technology.

26. Veth, Michael J. and John F. Raquet. "Alignment and Calibration of Optical and Inertial Sensors Using Stellar Observations". *Proceedings of ION GNSS 2005*, 2494–2503. September 2005.

27. Veth, Michael J. and John F. Raquet. "Two-Dimensional Stochastic Projections for Tight Integration of Optical and Inertial Sensors for Navigation". *Proceedings of the Institute of Navigation National Technical Meeting*, 587–596. 2006.

28. Veth, Michael J. and John F. Raquet. "Fusion Of Low-Cost Imaging And Inertial Sensors For Navigation", 2007. Air Force Institute of Technology.

29. Veth, Michael J., John F. Raquet, and Meir Pachter. "Stochastic Constraints for Efficient Image Correspondence Search". *IEEE Transactions on Aerospace Electronic Systems*, 42(3):973–982, July 2006.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
03/27/2008	Master Thesis	Aug 2006 – Mar 2008

4. TITLE AND SUBTITLE

Tightly Integrating Optical And Inertial Sensors For Navigation Using The UKF

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Sedat Ebcin, 1st Lt, TUAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/08-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL-MN
Dr. Mikel Miller
101 W. Eglin Blvd, Bldg 13
Eglin AFB, FL 32542-6810
Phone: (850) 496-2886 E-mail: mikel.miller@eglin.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approval for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The motivation of this research is to address the benefits of tightly integrating optical and inertial sensors where GNSS signals are not available. The research begins with describing the navigation problem. Then, error and measurement models are presented. Given a set of features, a feature detection and projection algorithm is developed which utilizes inertial measurements to predict vectors in the feature space between images. The unscented Kalman filter is applied to the navigation system using the inertial measurements and feature matches to estimate the navigation trajectory. Finally, the image-aided navigation algorithm is tested using a simulation and an experiment. As a result, the optical measurements combined with the inertial sensors result in improved performance for non-GNSS based navigation.

15. SUBJECT TERMS

Image-aided navigation, inertial navigation, passive navigation, autonomous navigation, computer vision, correspondence search, sensor fusion, tightly-coupled, unscented Kalman filter. non-GNSS based navigation, non-GPS based navigation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Maj Michael Veth, USAF
U	**U**	**U**	**UU**	74	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext:4541; Michael.Veth@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18