3-23-2017

# Small Fixed-wing Aerial Positioning Using Inter-vehicle Ranging Combined with Visual Odometry

Benjamin M. Fain


# SMALL FIXED-WING AERIAL POSITIONING USING INTER-VEHICLE RANGING COMBINED WITH VISUAL ODOMETRY

THESIS

Benjamin M. Fain, Capt, USAF

AFIT-ENG-MS-17-M-027

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

AFIT-ENG-MS-17-M-027

# SMALL FIXED-WING AERIAL POSITIONING USING INTER-VEHICLE RANGING COMBINED WITH VISUAL ODOMETRY

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

Benjamin M. Fain, B.S.E.E.

Capt, USAF

23 March 2017

AFIT-ENG-MS-17-M-027

# SMALL FIXED-WING AERIAL POSITIONING USING INTER-VEHICLE RANGING COMBINED WITH VISUAL ODOMETRY

THESIS

Benjamin M. Fain, B.S.E.E.
Capt, USAF

Committee Membership:

Dr. J. Raquet, PhD
Chair

Dr. D. Jacques, PhD
Member

Dr. K. Kauffman, PhD
Member

Dr. R. Leishman, PhD
Member

AFIT-ENG-MS-17-M-027

# Abstract

There has been increasing interest in developing the ability for small unmanned aerial systems (SUAS) to be able to operate in environments where GPS is not available. Visual odometry (VO) approaches have been demonstrated which provide a form of dead reckoning, but do not determine absolute position. If we consider the scenario where a higher flying aircraft is providing range updates to a lower flying GPS-denied SUAS, and the higher flying aircraft knows its position, the higher flying aircraft can provide absolute position information to the lower flying SUAS. A single range however, is not sufficient for the SUAS to solve for its position as it does not provide full 3-D position information. This thesis investigates the use of a combination of both ranging to a single aircraft and VO to determine an absolute navigation solution. It is demonstrated that a ranging update combined with an aircraft motion model and VO can greatly improve the accuracy of a SUASs estimated position in a denied GPS environment. Commercial-off-the-shelf (COTS) components and a simplified position estimation algorithm were used to estimate the SUASs position. A SUAS fitted with a non-GPS Pixhawk autopilot, a downward facing monocular camera, and a RF ranging device were used to collect real-time flight data. A second SUAS was used to provide RF ranging from a known loiter position. Attitude, body-axis rates, and airspeed data were fed into a simplified aircraft motion model using an Extended Kalman Filter (EKF) for estimation. Using the downward facing camera, VO provided velocity corrections to the aircraft motion model states. Ranging data from a simulated aircraft with a known trajectory was used to update the motion models position estimation resulting in greater position accuracy then the aircraft model and VO alone.

# Acknowledgements

I would like to thank Dr. John Raquet for guiding my thoughts and ideas towards an obtainable end goal. His appreciation and admiration towards students shined through every step of this master's research process. I also want to thank Jeremy Gray, Daniel Marietta, and Rick Patton for their support throughout this process. Above all, to my loving wife who's support at home during times of uncooperative hardware and unyielding code helped me strive to this great accomplishment.

Benjamin M. Fain

# Contents

Page

Page

# List of Figures

# List of Tables

# List of Abbreviations

| Abbreviation | | Page |
|---|---|---|
| CP-DGPS | Carrier Phase Differential Global Positioning System | 19 |
| PTP | Precision Time Protocol | 27 |
| MANET | Peer-to-Peer Mobile Ad Hoc Network | 30 |
| LCM | Light Weight Communication and Marshalling | 31 |
| RMS | Root Mean Square | 54 |
| 2DRMS | 2x Distance Root Mean Square | 54 |

SMALL FIXED-WING AERIAL POSITIONING USING INTER-VEHICLE RANGING COMBINED WITH VISUAL ODOMETRY

# I. Chapter 1 - Introduction

The Global Positioning System (GPS) is a satellite based navigation system designed to provide a position estimate for an object near or on the earth's surface. The GPS system, since becoming operational within the United States Department of Defense in 1993, has been instrumental in the precise positioning of current and past systems [10]. Within the military, GPS is currently the primary tool used for navigation for many types of missions. Current USAF assets, both manned and unmanned, depend on GPS as a means to accurately travel around the world. Growing concerns for the military's dependence on GPS [11] has increased the urgency of developing alternative navigation techniques.

Image aided navigation, an alternative navigation tool, uses an optical sensor to image the world and process the information into usable motion. Optical sensors can be used to estimate a change in pose. Inertial measurement units (IMU) can be used to estimate the change in position or attitude of an object by processing the rotational, vertical, and horizontal forces measured by the sensors. Research has been sought after to combine an optical sensor and an IMU to estimate the positional change of a body as it travels along in space. Varying types of IMUs have been paired with varying types of optical sensors and algorithms to assist or even replace GPS and reduce the dependency of the military on GPS [12][13][14][15][16].

## 1.1 Problem Statement

The research outlined in this thesis looks to contain the inherent drift seen in IMU and optical imagery applications that do not incorporate an absolute position into the position estimate. This thesis aims to present trends in position estimation using a specific aircraft motion model by varying the sensor inputs, including ranges and VO, into a filter estimation.

## 1.2 Motivation

The military has become dependent on small unmanned aerial systems (SUAS) to provide information about the battlefield. SUASs have been used to track enemy movements, locations, and capabilities, as well as, deliver supplies to soldiers on the ground. The list of mission possibilities continues to grow however most, if not all, military SUASs relay on GPS for navigation. Applications of alternative navigation which are not reliant on GPS continue to be explored within the Department of Defense in order to continue the mission in GPS denied environments. As outlined in the Air Force Future Operating Concept for 2035, the Department of Defense and United States Air Force are seeking out alternative navigation that can use avaliable resources beside US based GPS satellites [17]. The research outlined in this thesis continues along the lines of alternative navigation and looks to investigate the addition of absolute ranging information in a combination of situations involving an optical sensor and a simplified fixed wing aircraft motion model.

This research considers the case of a larger aircraft loitering above a smaller GPS-denied SUAS. This larger aircraft is assumed to have greater resources which can overcome the GPS jamming and provide range to a known GPS point as information to the SUAS flying a mission below. Figure 1 is a depiction of a larger flying aircraft providing range measurements and current location to the lower flying GPS-denied

SUAS. The SUAS uses the provided range combined with a downward facing optical sensor and a motion model to estimate its current position. This research seeks to demonstrate that using a ranging update combined with an aircraft motion model and visual odometry (VO) can greatly improve the accuracy of a SUASs estimated position, compared to applications that do not incorporate an absolute position update, in a GPS-denied environment.

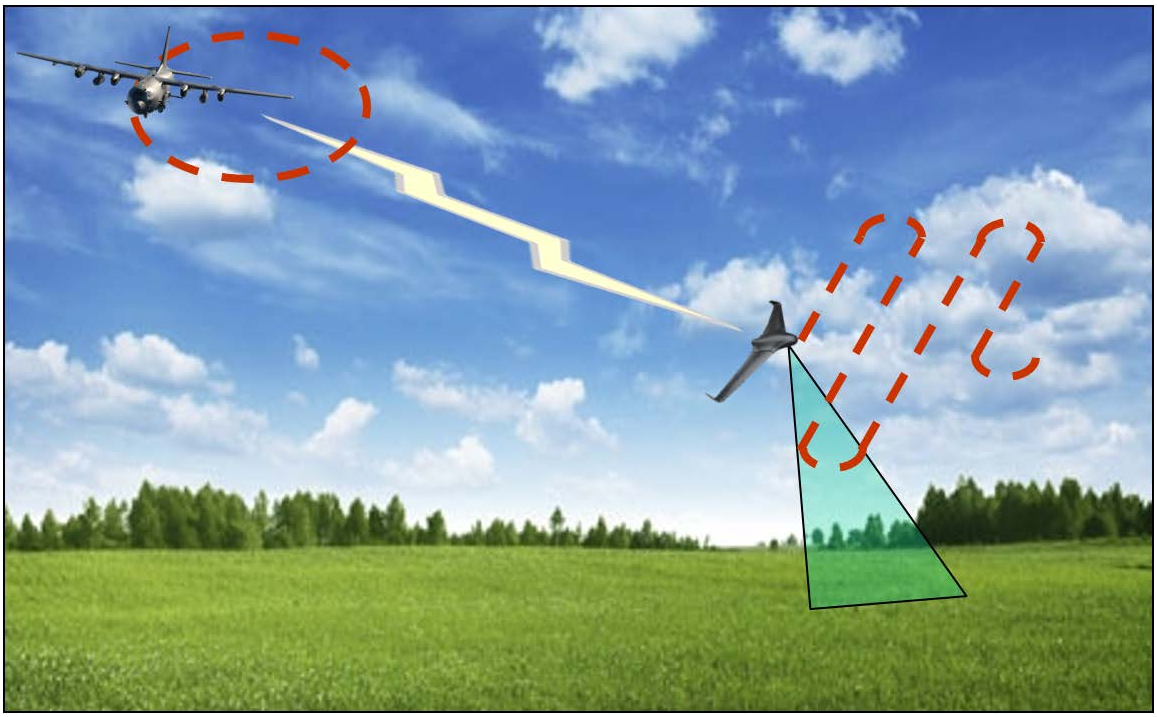


**Figure 1. High Flying Aircraft Providing Range Measurements to a Lower Flying SUAS**

## 1.3 Assumptions

The research described in this thesis assumes a flat earth model, where the terrain beneath the SUAS is at the same height. This assumption is important when using the optical sensors, since height errors can lead to VO errors.

The truth GPS position collected from the autopilot inherently has error in it, given the quality of the GPS sensor being used and the availability of the GPS satel-

lites. Possible changes to the results based on errors in the truth data are not discussed as simulations for ranges were constructed using the GPS position from the Pixhawk autopilot.

The Beard and McLain kinematic aircraft motion model assumed coordinated turns which result in no side slip of the aircraft during turns [10]. The results displayed in this thesis assume the data sets collected adhered to coordinated turns. Results cannot be compared to highly dynamic or inverted flight characteristics as the motion model used in this research would not be consistent with highly dynamic flight characteristics.

## 1.4 Thesis Overview

Chapter II contains the required background knowledge necessary to conduct this research. A summary of reference frames and estimation filters used are presented for the reader's understanding. A set of VO techniques are presented along with a discussion of related research.

Chapter III describes the methodology, the motion model, sensors, and estimation filter. Since this research required hardware for data collection, a description of the SUASs flown are described including the build up of the payloads are also discussed in detail in this chapter.

In Chapter IV an analysis is performed to characterize the performance obtained by varying the sensors used in the estimation filter. Anomalies in the results are also discussed.

Chapter V is a summary of the work stated in the prior chapters. Additionally suggestions for future work are presented.

# II. Chapter 2 - Background

This chapter provides a brief foundation of understanding for the research proposed in future chapters. It begins with reference frame definitions used in this study. Next estimation and filtering framework will be discussed along with various visual odometry methods. Monocular and stereo camera systems and types of image feature detection methods will be presented along with published research similar in nature to the proposed study.

## 2.1 Reference Frames

All navigation problems require a reference frame to be defined. This thesis refers to The 1984 World Geodetic System (WGS-84), Navigation, and Body frames, which are defined.

### 2.1.1 WGS-84 Reference Frame.

WGS-84 models the Earth as an ellipsoid where flattening of the earth occurs at the poles. The ellipsoid is defined using a semi-major axis and eccentricity. Latitude, longitude and altitude are defined relitive to the ellipsoid. Latitude and longitude are typically measured in radians or degrees while altitude is measured in a unit of linear distance [18].

### 2.1.2 Navigation Reference Frame.

The navigation frame also known as the North, East, Down (NED) frame or North, East, Up (NEU) frame is commonly used in navigation as a near by origin can be used. For this thesis NEU is used as the altitude is in terms of meters above the ground. The NEU axes in this thesis are aligned to the magnetic north and east

direction of the chosen origin point. The up axes is aligned orthoganl to the Earth's surface with the origin being on the surface of the Earth and the positive direction being up and away from the Earth's surface. This thesis uses an above ground level (AGL) altitude definition, meaing altitude is calculated in the number of meters from the earths surface the aircraft is in the air [10].

### 2.1.3 Body Reference Frame.

The body frame defines the orientation of the vehicle and the origin is generally located at the center of gravity or location of the guidance system on the vehicle. The X axis typically points out the noise of the aircraft aligned north while the Y axis points out the right wing aligned east. The Z axis points out the bottom of the aircraft. The attitude of the vehicle can then be defined as rotations about body frame axes relative to an aircraft pointing north with wings level. The attitude is then expressed with Euler angles as yaw ($\psi$), pitch ($\theta$), and roll ($\phi$). A given orientation is represented by first applying the yaw then the pitch then the roll [10]. This is also defined in [10] as applying the vehicle frame, then the vehicle-1 frame then the vehicle-2 frame. Also defined in the body frame are the angular velocities p, q, and r which quantify the rate at which the body is rotating. Since the Euler angles are defined according to intermediate vehicle frames, p, q, and r are not simply the time derivatives of the attitude angles yaw ($\psi$), pitch ($\theta$), and roll ($\phi$). They are defined as [10]:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{1}$$

## 2.2 Estimation and Filtering

When collecting data in real world applications it is inherently impossible to collect exact truth data pertinent to the systems. In navigation, sensors are used to collect data about the body or vehicle, whether it is GPS coordinates, acting forces, or orientation. All sensors are corupted by noise or other errors. Since the measurements cannot be taken for truth, a system of estimating the body or vehicle states is needed to filter the estimate as close to the truth as possible [19]. Three forms of filtering used in this research include the Kalman Filter (KF), also known as the Traditional Kalman Filter, the Extended Kalman Filter (EKF), and the Delayed State KF, which is a version of the EKF. All three types of filters are described below.

### 2.2.1 Kalman Filter.

In many applications of navigation, a KF is used to estimate the pose or navigation states of a vehicle. The KF assumes the dynamics equations and measurement equations are linear and Gaussian in nature allowing the probability distribution of the states to be represented with a mean and covariance. The general discrete dynamics model is:

$$\mathbf{x}_k = \mathbf{\Phi}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k \qquad \mathbf{Q}_d = E[\mathbf{w}_k\mathbf{w}_k^T] \tag{2}$$

where $\mathbf{\Phi}$ is the state transition matrix, $\mathbf{B}$ relates the inputs $\mathbf{u}_k$ to the states $\mathbf{x}_k$, and $\mathbf{G}$ is the process noise $\mathbf{w}_k$ added to the states with covariance $\mathbf{Q}_d$. The measurement model:

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \qquad \mathbf{R} = E[\mathbf{v}_k\mathbf{v}_k^T] \tag{3}$$

relates the measurements $\mathbf{H}$ to the states $\mathbf{x_k}$ where $\mathbf{v_k}$ is the estimated added sensor noise with covariance $\mathbf{R}$.

The KF operates in two steps, a propagation step and an update step. The propagation step propagates the mean:

$$\hat{\mathbf{x}}_k^- = \Phi \hat{\mathbf{x}}_{k-1}^+ \tag{4}$$

and the covariance:

$$\mathbf{P}_{k+1}^- = \Phi \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_{d_k} \tag{5}$$

of the estimated states for each time step. The update occurs when a measurement has been taken. A Kalman Gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} \tag{6}$$

weights the new information provided by the sensors. The mean of the states for the given time step is updated using:

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_k [\mathbf{z}(t_{k+1}) - \mathbf{H}_k \hat{\mathbf{x}}_{k+1}^-] \tag{7}$$

while the covariance is updated using:

$$\mathbf{P}_{k+1}^+ = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^- \tag{8}$$

where $\mathbf{I}$ is the identity matrix. A more in depth derivation of a traditional KF can be found in [20].

### 2.2.2 Extended Kalman Filter.

The EKF is used in navigation when the dynamics model and/or the sensor model are no longer linear. Similar to Equation (2) the continuous EKF dynamics equation is:

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \mathbf{G}(t)\mathbf{w}(t) \tag{9}$$

where the Jacobin is taken of the nonlinear $\mathbf{f}$ as:

$$\mathbf{F}_i \triangleq \left.\frac{\partial \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)}{\partial \mathbf{x}(t)}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_k^+} \tag{10}$$

The non-linear EKF measurement equation is:

$$\mathbf{z}(t_k) = \mathbf{h}\left[\mathbf{x}(t_k), t_k\right] + \nu(t) \tag{11}$$

where $\mathbf{h}$, which relates the measurements to the states must also be linearized as:

$$\mathbf{H}_k \triangleq \left.\frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}(t)}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_{i.}^-} \tag{12}$$

Similar to the KF, the discrete EKF mean and covariance are propagated according to:

$$\widehat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \widehat{\mathbf{x}}_k^+ \tag{13}$$

and:

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{Q}_{d_k} \tag{14}$$

respectively. During the update step given a measurement a Kalman gain $\mathbf{K_k}$ is calculated according to:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \tag{15}$$

where $\mathbf{R_k}$ is the measurement uncertainty and the mean is updated using:

$$\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{z}(t_k) - \mathbf{h}(\widehat{\mathbf{x}}_k^-, t_k)\right] \tag{16}$$

and the covariance is updated as:

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k.}^- \tag{17}$$

### 2.2.3 The Delayed-State Kalman Filter.

The delayed-state KF was intended to be used in this research; however, the delayed-state KF within Scorpion, a KF processing tool, was still in development [21]. The delayed-state KF could have been used to incorporate measurements that happened prior to the current filter time. These measurements carry insight into how a state changed over time [22]. Similar to the EKF the Delayed-State Kalman Filter from [23] is used to estimate the states using updates from measurements. The measurement $\mathbf{z}$ for the Delayed-State Kalman Filter is:

$$\mathbf{z} = \int_{t_{i-1}}^{t_i} \mathbf{x} dt \tag{18}$$

where $\mathbf{x}$ is the state. Using the measurement model in Equation (18) the measurement update for a position change can be written as an integrated velocity over the time interval of the position change. This is expressed as:

$$\mathbf{z} = \int_{t_{i-1}}^{t_i} \mathbf{v} dt = pos(t_i) - pos(t_{i-1}) \tag{19}$$

where $\mathbf{v}$ is the velocity. In general form the measurement equation is:

$$\mathbf{z}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{J}_i \mathbf{x}_{i-1} + \nu_i \tag{20}$$

and the delayed-state KF using an EKF update is modeled as:

$$\mathbf{z}_i = \mathbf{h}(\mathbf{x}_i^-, \mathbf{x}_{i-1}) + \mathbf{v}_i \tag{21}$$

where $\mathbf{H}_i$ and $\mathbf{J}_i$ are:

$$\mathbf{H}_i = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_i^-, \widehat{\mathbf{x}}_{i-1}} \qquad \mathbf{J}_i = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_{i-1}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_i^-, \widehat{\mathbf{x}}_{i-1}} \tag{22}$$

The delayed-state KF update equations for an EKF are then:

$$\widehat{\mathbf{x}^+}_i = \widehat{\mathbf{x}}_i^- + \mathbf{K}_i(\mathbf{z}_i - \widehat{\mathbf{z}}_i^-) \tag{23}$$

where $\widehat{\mathbf{z}}_i^-$ is

$$\widehat{\mathbf{z}}_i^- = \mathbf{h}(\widehat{\mathbf{x}}_\mathbf{i}^-, \widehat{\mathbf{x}}_{i-1}) \tag{24}$$

The Kalman gain is then:

$$\mathbf{K}_i = \left[\mathbf{P}_i^- \mathbf{H}_i^T + \Phi_{i-1}\mathbf{P}_{i-1}\mathbf{J}_i^T\right] \mathbf{L}_i^{-1} \tag{25}$$

where the residual covariance $\mathbf{L}_i$ is:

$$\mathbf{L}_i = \mathbf{H}_i\mathbf{P}_i^-\mathbf{H}_i^T + \mathbf{R}_i + \mathbf{J}_i\mathbf{P}_{i-1}\Phi_{i-1}^T\mathbf{H}_i^T + \mathbf{H}_i\Phi_{i-1}\mathbf{P}_{i-1}\mathbf{J}_i^T + \mathbf{J}_i\mathbf{P}_{i-1}\mathbf{J}_i^T \tag{26}$$

and the state covariance update is:

$$\mathbf{P}^+{}_i = \mathbf{P}_i^- - \mathbf{K}_i\mathbf{L}_i\mathbf{K}_i^T \tag{27}$$

The state propagation equations for the delayed-state KF are the same as the EKF where the state estimate is updated using Equation (13) and the state covariance is

updated using Equation (17). A more in depth description and derivation of the delayed-state KF can be found in [23]. An example of using a delayed-state KF can be found in [22].

## 2.3 Visual Processing

Visual Odometry is a process using sets of sequential imagery to determine the change in pose of a vehicle. This is similar to mechanical odometry where the rotations of a wheel are used to calculate the distance a vehicle has traveled. VO is broken down into three primary categories which include traditional VO, Visual Simultaneous Localization and Mapping (VSLAM), and Optical Flow (OF) [24].

### 2.3.1 VO.

Traditional VO is the process of integrating the change in images over time compared with the initial image to calculate a change in pose or position traveled [24]. In an aerial application sequential ground images would be taken from the air and used to calculate an estimated distance traveled over the given integrated time frame.

### 2.3.2 VSLAM.

Visual Simultaneous Localization and Mapping or VSLAM is a visual approach to building a map of the surroundings while simultaneously locating the vehicle within the newly created map. A simple example is walking in circles around a building while trying to locate a position at each step taken. After walking around the first time a visual mental map is built making it much easier to locate orientation and position within the building. As described in the example VSLAM is a great tool used when a body or vehicle traverses over the same area more than once [14].

### 2.3.3 OF.

Optical Flow is the process of measuring pixel intensity flow vs feature tracking of a set of images. A feature in an image could be a flower, while pixel intensity flow is the direction and relative speed the flower is moving in the image plane of the camera. OF is used by insects and birds as a way to measure individual motion relitive to their surroundings [1]. Figure 2 shows as the observer moves forward the scene appears to be moving towards the observer. This suggests the observer is moving forward with the assumption the flowers and grass are not moving.

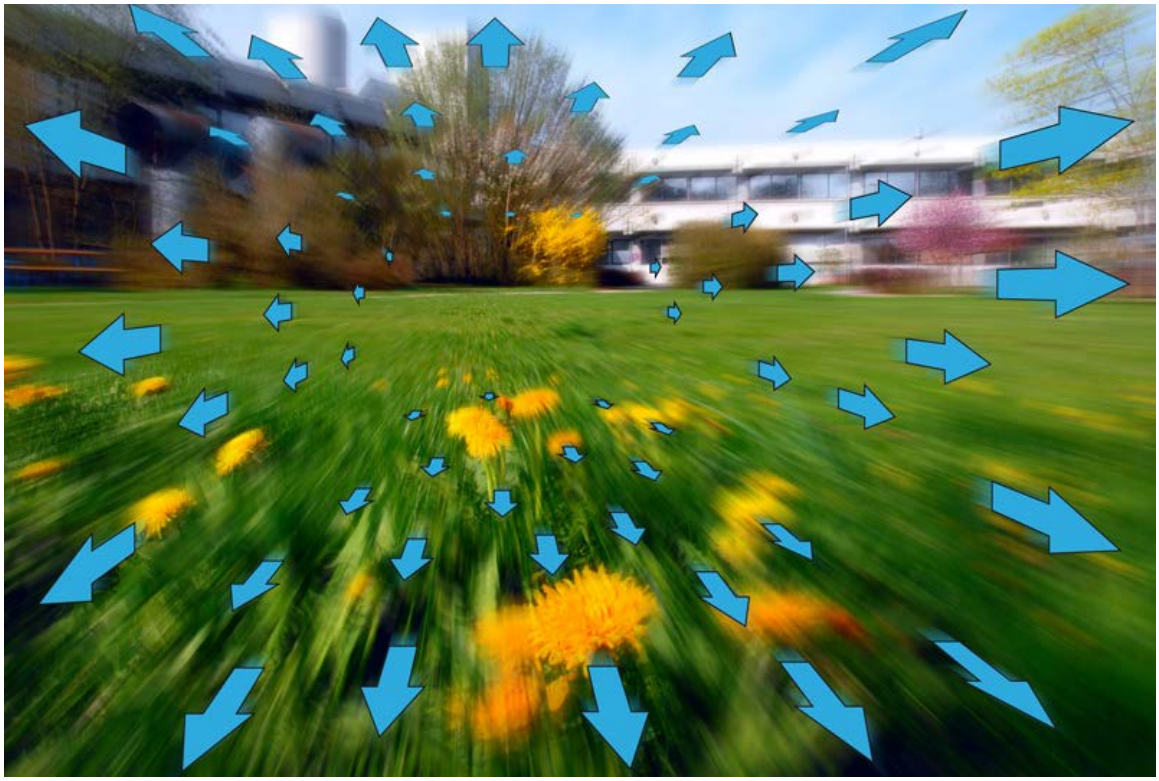


**Figure 2. Optical Flow Example [1]**

## 2.4 Monocular vs Stereo

When navigating using a camera, research has been conducted using a single camera, known as monocular, and with two cameras, known as stereo. In some cases

more then two cameras are used in navigation research, however they are not discussed in this study.

### 2.4.1 Monocular.

Monocular navigation is the process of navigating using a single camera attached to the body or vehicle. Generally a set of images are taken from the single camera as the body or vehicle moves around. From a single camera system the rotation or direction of travel of the vehical can be determined, however the distance the vehical traveled cannot be determined from imagery data alone. In [15] a vehicle was driven along a roadway using a single camera and a version of VSLAM to visually map 18km of suburban roadway. Also using a monocular camera, [2] used a Yamaha Rmax radio controlled (RC) helicopter and VO combined with other sensors to navigate around an area of Sweden. Other examples of monular navigation can be found in [25],[26],and [13].

### 2.4.2 Stereo.

Stereo navigation is the process of using two cameras to produce a set of sequential imagery for navigation. Stereo camera systems can provide rotation and translation information, as well as, scale information based on the known distance between the two cameras. In [27] a stereo camera system using only a VO algorithm was used to navigate an Unmanned Ground Vehicle (UGV) along uneven terrain. With only VO and a stereo camera system, an error accuracy of 0.25% over a 400m distance traveled was observed. More examples of stereo vision used in navigation applications can be found in [28] and [16].

## 2.5 Image Feature Detection

Image feature detection is the process of uniquely identifying features or objects within an image. The features are then used as a way to match similar features in separate images to detect motion. The matching of images and the ability to detect motion can then be translated into navigation states. Corner detection, edge detection, Scale Invariant Feature Transform (SIFT), Speeded Up Robust Feature (SURF), and Accelerated-KAZE (AKAZE) are common feature detection methods further described below.

### 2.5.1 Corner Detection.

Corner detection is the process of searching for physical corners in an image. Corners are usually represented by two intersecting orthoganal lines that end at the intersection. In the scene of a city skyline, a corner detector would likely detect the corners of a building as features in the image. If another image were taken and the corners had moved within the image frame, motion could be derived from the two images. A common corner detector algorithm used today is the Harris corner detector, an advancement of the Moravec detector, and it is actually a combination of a corner and edge detector [29]. Another type of corner detector is FAST, or Features from Accelerated Segment Test. This feature detection algorithm is designed for real-time applications due to its acclaimed processing speed [30].

### 2.5.2 Edge Detection.

Edge detection looks for features in an image along object edges based on sharp changes in intensity or brightness [31]. Again using the case of a city skyline, an edge detector will focus on image features along the edges of the building such as those created by the contrast between the building and the sky. The Canny edge detector

is one version of an edge detector, though it can be computational expensive [32]. Other edge detectors, including Sobel and Prewitt, can be explored further in [32].

### 2.5.3 SIFT/SURF.

In [33] David G. Lowe developed SIFT which uses a process of Gaussian blurring to find features within an image. In [34] SIFT was used as the main image feature detection and the results of SIFT in a SUAS application can be explored further in that study. SURF is very similar to SIFT, except it is computationally more efficient making it a more viable solution for real-time use on large image sets. More information on SIFT and SURF as well as a comparison between the two, can be found in [35].

### 2.5.4 AKAZE.

This thesis used OpenCV, an open source computer vision library, to perform image feature detection [36]. AKAZE, availiable in OpenCV, is an image feature detector built off of KAZE, meaning wind in Japanese. Unlike SIFT and SURF that use Gaussian blurring for feature detection, AKAZE is designed to detect features in nonlinear scale spaces with the ability to blur small details while protecting object boundaries of an image [37][38]. A k-Nearest Neighbor (kNN) image feature matching algorithm was used that matched a known image feature to a threshold-ed number of image feature neighbors to determine a match [39]. The AKAZE feature detector and kNN feature matching algorithim implemented in this thesis can be found in [40].

## 2.6 Background Research

Similar research was found incorporating various combinations of navigation sensors to include an IMU, monocular VO, georefrence imagery, and/or supplementary

UAV ranging information. No research could be found that incorporated all aforementioned forms of navigation simultaneously to create an absolute position estimation in real time.

### 2.6.1 VO & Georefrenced Imagery.

In the area of image registration, also known as georefrenced imagery, and VO, Conte & Doherty in 2008 successfully used VO loosely coupled with georeferenced imagery and inertail sensors to aid a Yamaha RMAX helicopter state estimate [2]. As seen in Figure 3 from [2] a traditional KF was used to combine IMU measurements, a georeferenced image feature database, and VO to update the helicopters state estimate. INS and VO alone caused the estimated states to drift over time. Adding a georeferenced position estimate reduced the drift in the helicopters position. The georeferenced image matching measurement used a Sobel edge detector algorithm to match the real-time imagery with georeferenced imagery. When a match was found a measurement update was fed into the KF. In comparision to [2], the research described in this thesis uses an EKF instead of a traditional KF, and a range measurement from a known location was used to update the state estimate. Similar to Conte & Doherty's work, this thesis used a range update to a known location as an absolute position update. The range update, like the gerefrenced imagery update, both provided absolute position data into the navigation solution which help to contain the drift in the IMU and VO alone.

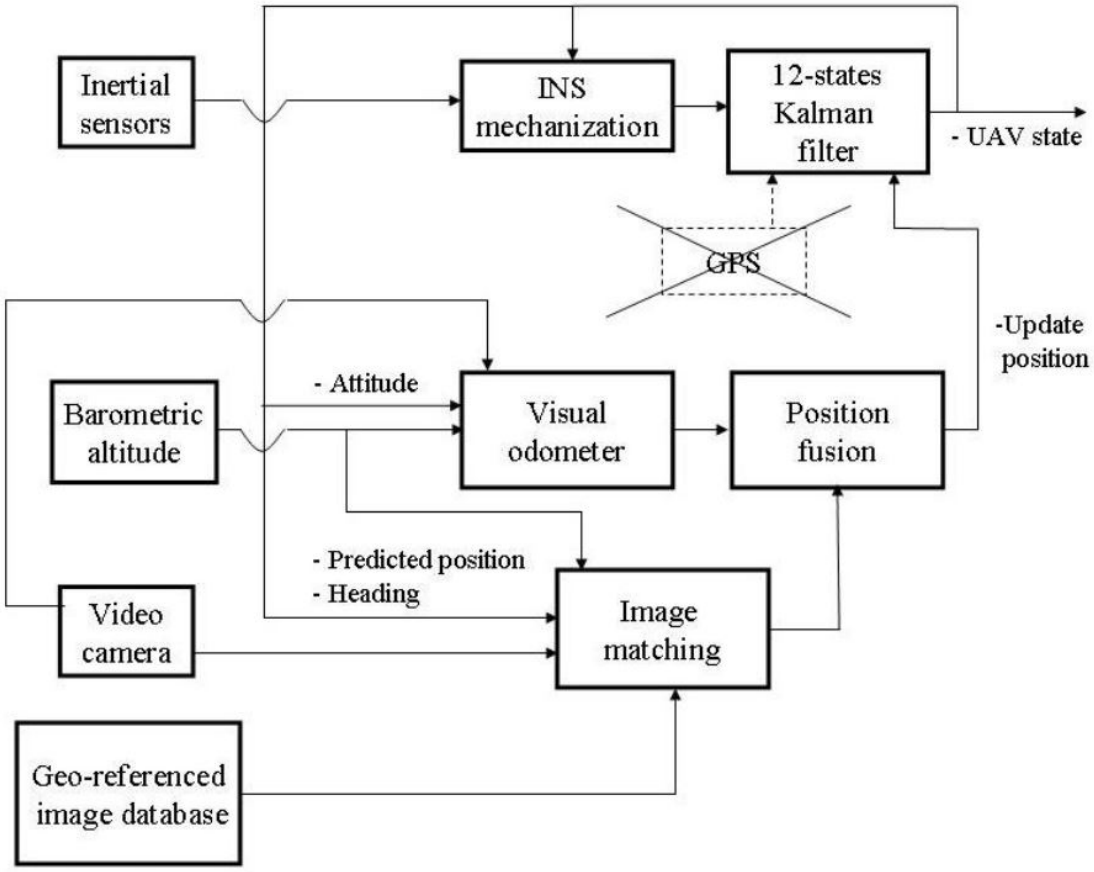



**Figure 3. Vision-aided Sensor Fusion Architecture [2]**

In [34] a UAV position estimate was calculated by comparing aerial images of the ground with georefrenced imagery, called Perspective-n-Point (PnP), using SIFT as a feature detector based on an algorithum described in [41]. Using a 3 degree of freedom solution, where the altitude of the aircraft was determined by the autopilot, the lowest distance root mean squared (DRMS) error was 8.22 meters. This means the location of the aircraft would be known to 8.22 meters 67% of the time. [34] focused on comparing georefrenced imagery with real-time aerial monocular imagery. From [34] it was noted that using the autopilot attitude in the 3 degree of freedom solution allowed for a better position estimate. Allowing the software to estimate the attitude (6 degrees of freedom) produced a worse position estimation then the 3

degrees of freedom solution. In this research the same autopilot was chosen and the autopilot attitude was used in the VO to track the camera rotation between images as it appeared to be sufficient in [34].

### 2.6.2 LOS Ranging.

Another aspect of this research was incorporating line of sight (LOS) ranging information into the EKF, to update the aircraft's state estimate. Although not on a UAV, Wu & Davidson [42] developed a diverse ranging concept that could be used for airlines in the event of a GPS outage. The system used ground based transmitters and receivers, as well as, transmitters and receivers on the commercial aircraft. Using this setup, aircraft would operate in Direct-Reply (DR) mode providing real-time aircraft location information to the ground stations. The majority of aircraft operating in Non-Reply (NR) mode, then used the ranges to the ground stations and surrounding DR aircraft range to calculate a current position. Compared to the Wu & Davidson [42] case, the DR mode aircraft in this thesis would be the SUAS providing the ranging information (or signal to calculate the ranging) and the NR aircraft would be the SUAS flown with the VO equipment using the range information to calculate a position described in this research.

Also using ranging information, [43] used peer-to-peer ranging radios for close formation flights. The ranging radios were placed on each air frame and ranging was done relative to each other as a supplement to the Carrier Phase Differential Global Positioning System (CP-DGPS) during times of degraded CP-DGPS. In comparison, this thesis incorporated ranging information between aircraft; however, the ranging information was used to help calculate an absolute position versus a relative formation flight range.

## 2.7 Chapter Conclusion

In summary, a quick introduction of the components comprising this research were presented for the readers understanding. Reference frames including the WGS-84, navigation, and body frames, were discussed. The topic of estimation and filtering including a description of a KF, EKF and delayed-state KF were summarized. VO, VSLAM, and OF were defined and explained with examples. Two camera configurations, including monocular and stereo, were researched and both methods were presented. Image feature detection algorithms were also described. Finally a description of similar research was discussed.

# III. Chapter 3 - Aircraft and Filter Design

In this thesis, two RC aircraft were modified to carry payloads designed to collect autopilot data, images, and range information used for post processing. One aircraft carrying a ranging radio was used to transmit a range and its location to the second aircraft which had a camera for VO, a ranging radio, and a GPS denied autopilot to provide aircraft attitude and attitude rates. The data collection was setup to simulate a lower flying GPS denied UAV navigating using a motion model with updates from VO, a barometer, a heading, and ranges to the higher flying aircraft. The research data collection and subsequent analysis was conducted in two parts. The first part was the build up of the hardware on the aircraft followed by the development of the software and drivers to collect the data. The combination of hardware and software to produce a consistent data output was coined the term smart cable. The second part was using Scorpion, a filter estimation framework, to analyze the data. Chapter 3 discusses the approach and system design taken to conduct this research. The physical build of the payloads used on the RC aircraft will be discussed, as well as, the filter estimation used in this research.

## 3.1 Data Collection

This section discuss the physical hardware and software used to collect the data during flight. Smart cables were built including one for the camera, one for the autopilot, and two for the ranging. The term smart cable encompasses the pairing of hardware and software to interact with a particular sensor and then output the data in a certain format for the system to process or record. A smart cable is designed to be a stand-alone entity with the ability to be used cross platform on multiple different projects. This project also used modified smart cables were multiple sensors

were attached to a single smart cable processor versus having one sensor per smart cable processor.

This project used three main payload configurations for data collection and are all very similar to Figure 4.

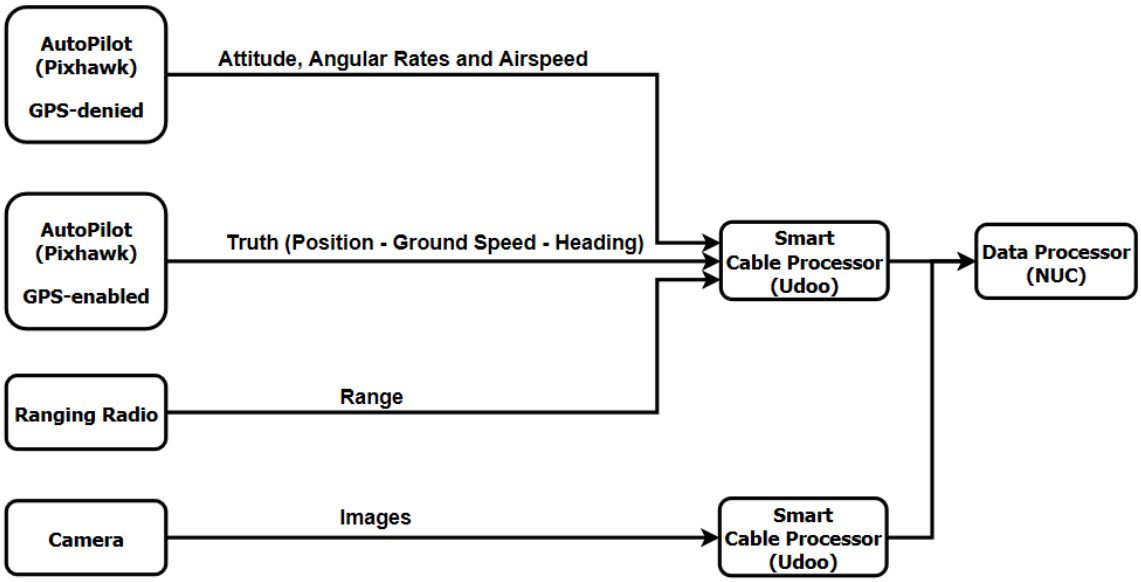



**Figure 4. Payload Configuration Overview**

### 3.1.1 Hardware.

This thesis used two RC aircraft to carry multiple sensors in flight. The sensors were connected to the data processor via smart cables. The first part of a smart cable is the hardware used to physically communicate with the sensor and produce the desired data output. In this section an overview of the hardware used to collect the flight data is presented.

#### 3.1.1.1 UAV Systems.

Two different RC aircraft —a Telemaster and SIG Rascal 110— were used for data collection, and both are commercially available to the hobbyist community.

Both aircraft were fitted with GPS enabled Pixhawk autopilots for navigation and a Wave Relay Board for IP based communication [44].

**Telemaster.** The Giant Telemaster seen in Figure 5 is a high wing aircraft with a 3.7 meter wing span and the ability to carry 14 kg [3]. Its large mid-body bay made it ideal for carrying a downward facing camera, two smart cable processors, an Intel data processing computer [8], a ranging radio, two Pixhawk autopilots [4], and two Ethernet switches. The non-flight secondary Pixhawk was disconnected from a GPS to ensure that it was truly GPS-free. The Giant Telemaster had one testing configuration being used as the lower flying aircraft during data collection. It was used to collect imagery, autopilot data, and range information.

**Figure 5. Giant Telemaster Used to Carry the Payload [3]**

**SIG Rascal 110.** The SIG Rascal 110 seen in Figure 6 is a high wing aircraft with a 2.8 meter wing span and the ability to carry 2.7 kg of payload[45]. The SIG Rascal 110 had two different configurations. The first configuration was used as

the higher flying aircraft providing the range information to the lower flying Giant Telemaster. In this configuration the SIG Rascal 110 was carrying only the additional ranging radio and an Udoo processor with a USB Ethernet switch. The second configuration involved converting the SIG Rascal 110 to be the sole aircraft used in the test. It was loaded with the downward facing camera, a non-GPS autopilot, and the NUC data processing computer with a USB powered Ethernet switch.

**Figure 6. SIG Rascal 110 Used to Carry the Payloads**

#### 3.1.1.2 Pixhawk Autopilot.

The Pixhawk autopilot seen in Figure 7 is a hobby grade aircraft autopilot made by 3D Robotics [4]. The autopilot is capable of flying predetermined routes using GPS waypoints. The sensors on board included an ST Micro L3GD20H 16 bit gyroscope, ST Micro LSM303D 14 bit accelerometer/magnetometer, Invensense MPU 6000 3-axis accelerometer/gyroscope and a MEAS MS5611 barometer. Interface to the Pixhawk was through two of the five UART serial ports [4]. The Pixhawk was also used with a uBlox GPS/magnetometer and an external digital airspeed sensor [4].

Communication with the Pixhawk for preplanned navigation purposes was through a 900Mhz RFD-900 radio datalink and Mission Planner, an open source software package for interacting with the Pixhawk [46][47].


**Figure 7. Pixhawk Autopilot Used for UAV Navigation [4]**

The Pixhawk was used to collect attitude, angular rates, and airspeed for the aircraft motion model, along with the GPS truth data. The Pixhawk autopilot was assumed to not provide true IMU attitude and angular rate data as the Pixhawk autopilot used an internal filtering algorithm to estimate the data that was used in the motion model.

#### 3.1.1.3 Smart Cable Processor.

Udoo Quads were chosen as the smart cable processor due to their low power consumption, small size, and success rate on previous smart cable projects. An Udoo Quad can be seen in Figure 8. The onboard ARM Quad Core 1GHz processor, 1GB of DDR3 ram, Ethernet port, and multiple USB ports made the Udoo Quad versatile with all of the sensors used in this research [5]. The Udoo Quads were operating on a command line version of Debian Linux to reduce the need for graphical processes. Two

Udoos where used on the Telemaster and one Udoo was used in the first configuration of the SIG Rascal 110. On all applications the Udoos were powered using a 12V regulator.

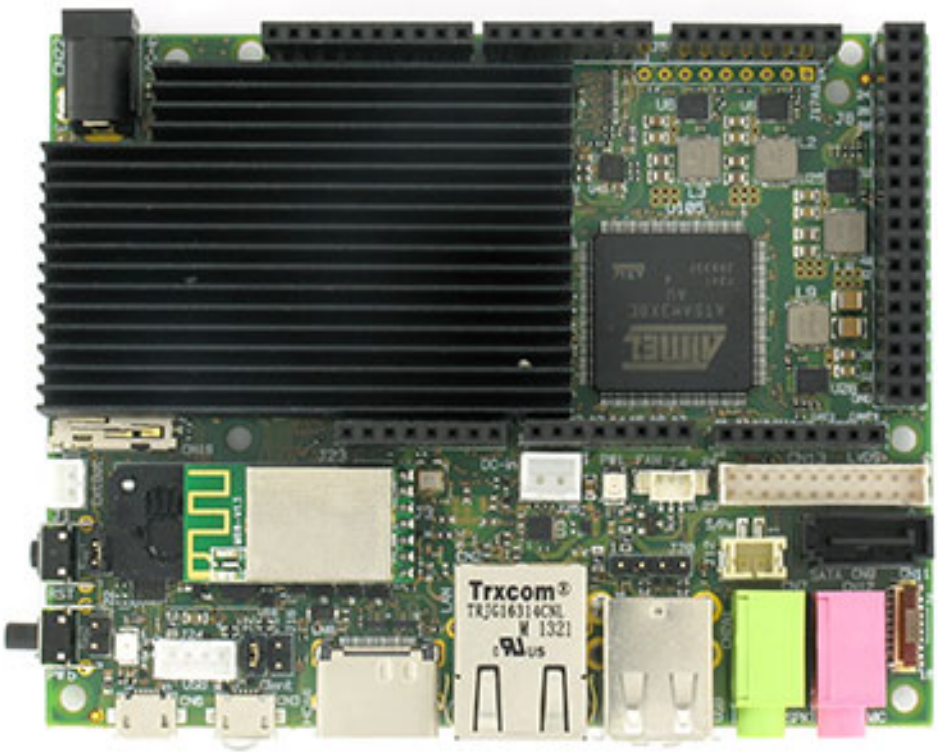



**Figure 8. Udoo Quad Used As a Smart Cable Processor [5]**

In order to connect the Udoo Quad to the Pixhawk autopilot, UART Serial Ethernet modules were programmed using USR-VCOM Virtual Serial Port Server as UDP servers and connected to the Pixhawk telemetry serial port. The Udoo Quads could then communicate with the Pixhawks through Ethernet. Each Pixhawk had a separate UART Serial Ethernet module. The USR-TCP232-T2 UART Serial Ethernet module can be seen in Figure 9 [6]. A diagram of how to connect the UART Serial Ethernet module to the Pixhawk can be found in the Appendix.

**Figure 9. UART Serial Ethernet Module for the Pixhawk Data Collection [6]**

#### 3.1.1.4 Procilica Camera.

The Allied Vision Prosilica GT1290, seen in Figure 10, is a black and white 1.2 Megapixel machine vision camera with a Gigabit Ethernet interface and a resolution of 1280 x 960 pixels. This camera was chosen partly because of its smaller size and precision time protocol (PTP) timing capability. The camera had the ability to time tag the image when it was taken using the PTP standard [7]. The stock fixed focus 1.4/3.5mm lens was chosen so the intrinsic camera calibration could be controlled. The camera was powered using a 12V regulator through pin 1 and 2 of the Hirose I/O port [7].


**Figure 10. Allied Vision GT1290 Camera [7]**

#### 3.1.1.5 NUC Processor.

The Intel NUC D53427RKE, seen in Figure 11, with an Intel Core i5 processor and a 500GB SSD was chosen as the data processor and main system that interacted with the Udoos smart cable processors [8]. The NUCs small size, low power consumption, and processing power made it ideal as the server. Ubuntu 14.04 was installed on the NUC and ithe NUC was powered by a 19V regulator.

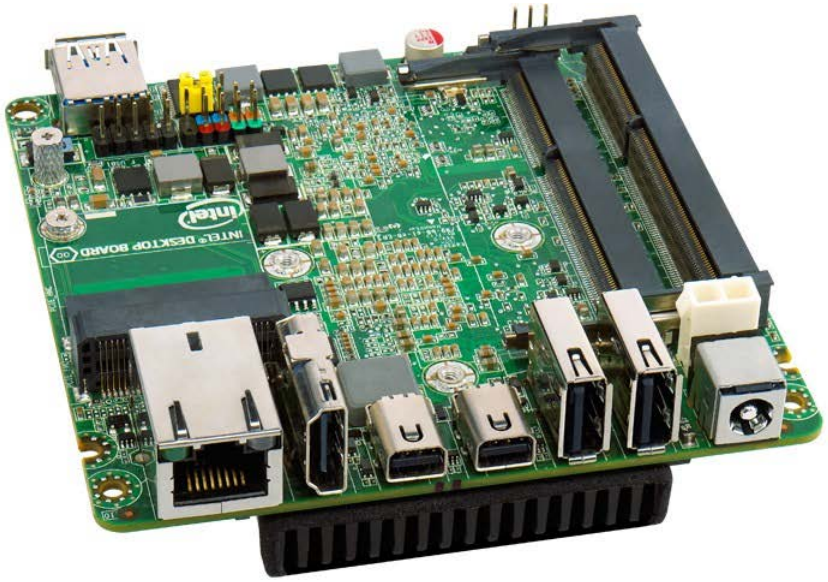


**Figure 11. Intel NUC Used As a Server and Data Processor [8]**

#### 3.1.1.6 Ethernet.

Ethernet was the main form of communication between the Udoos, NUC, Pixhawks, Wave Relay and GT1290 camera. To handle the Ethernet traffic two Netgear 5 port Ethernet switches were used on the Telemaster and 1 USB powered 5 port Ethernet switch was used on the SIG Rascal 110. The SIG Rascal had two different payload configurations and in both configurations a Dualcomm USB powered 5 port Ethernet switch was used [48]. An Ethernet diagram for the Telemaster and two SIG Rascal 110 configurations can be found in the Appendix.

#### 3.1.1.7 RF Ranging Radios.

ENSCO RF ranging radios were chosen to provide the ranging data as they operated at 5.8GHz away from the frequency required for the RC radio and the Pixhawkautopilot communication. They were also chosen due to their long range ability of 1 km or better and accuracy of up to 1 cm RMS in LOS applications. An internal lithium ion battery provided power while in flight. The RF Radios used 5.8Ghz omni-directional SMA antennas with one on each device [9]. One device was placed in the SIG Rascal 110 and one device was placed in the Telemaster connected by USB to one of the Udoo smart cable processors. A single ENSCO RF ranging radio can be seen in Figure 12

**Figure 12. ENSCO RF Ranging Radio [9]**

#### 3.1.1.8 Wave Relay.

Persistent Systems Wave Relay is a Peer-to-Peer Mobile Ad Hoc Network (MANET) designed to allow IP based communication between different nodes [44]. A Wave Relay was placed on the Telemaster and the SIG Rascall 110 to allow for IP based communication to the Linux based systems inside of the aircraft. The ground station seen in Figure 13 was mounted with two dipole antennas to better cover the testing area. Actual transmit speed and connectivity issues resulted in poor function of the Wave Relay leaving out the option to see data collection in real-time. Multiple antenna pattern directions were assessed without any better results.

**Figure 13. Wave Relay Ground Station**

### 3.1.2 Smart Cable Software.

The second part of a smart cable is the software and drivers used to communicate with the sensor and produce properly formatted sensor data. A smart cable was created for all of the sensors used in this thesis.

#### 3.1.2.1 LCM.

Light Weight Communication and Marshalling (LCM)is a tool for real-time systems providing the ability to publish and subscribe to data outputs or inputs directly between the peers as no central database is needed [49]. A smart cable was developed for each sensor publishing the required data on the network using LCM. Data from each sensor was logged on the NUC as the NUC was used to subscribe and record the published messages. Using LCM to log data resulted in a single file with the capability of being played back at varying speeds.

#### 3.1.2.2 PTP Timing.

PTP timing is an IEEE 1533-2008 timing standard that was used in this research for sub-microsecond clock synchronization. It operates on a master/slave concept as the master provides the desired time while the slaves syncronize to the master time [50]. In this research the NUC was set as the PTP master and the GT1290 camera and Udoos were set as slaves. The NUC was set to never update it's time so the slaves could synchronize with the NUC Unix time. All sensor data was stamped with the current unix time from the NUC upon arrival while the camera was able to time stamp the images taken before they were sent out on the network.

#### 3.1.2.3 Pixhawk Smart Cable.

The Pixhawk smart cable was designed to run on a Linux platform using Dronekit to interact with the Pixhawk and LCM to publish the data. Dronekit is a 3D Robotic developer tool for interacting with 3D Robotic autopilots including the Pixhawk [51]. A Python script was used to initiate the Dronekit connection to the autopilot and then publish the requested data onto the local network using LCM. On the Telemaster, an Udoo was used to communicate with the Pixhawk and publish the LCM message. On the SIG Rascal 110, an Udoo was used for configuration one, however, the NUC was used in configuration two due to space limitations. The Pixhawk smart cable was used to publish the GPS location, XYZ velocity in the body frame, XYZ raw acceleration in the body frame, XYZ raw rates of the gyros, attitude in the body frame, airspeed, ground speed, and heading. In the case of the Pixhawk autopilot without GPS, the smart cable would not return GPS coordinates and the ground speed would be inaccurate.

#### 3.1.2.4 Camera Smart Cable.

The camera smart cable was designed to run on Linux type platforms using Allied Vision's Vimba camera control drivers. Through third party sources a python wrapper was created for Vimba and nicknamed Pymba [52]. Using Pymba and LCM a smart cable was used to interact with the camera over Ethernet. The script created for the smart cable would tell the camera to link to the NUC as a PTP slave and then time tag and take images at the desired frequency. LCM was then used to publish the images onto the network as they were taken.

#### 3.1.2.5 RF Ranging Smart Cable.

A pair of RF ranging smart cables were developed with one being for the simulated range and the other for the actual ranging radios. The simulated ranging smart cable took in the GPS enabled Pixhawk published message and used the GPS data to output a simulated range and location from a simulated aircraft on a predetermined loitering trajectory. The location of the loiter point, the radius, and the loiter ground speed could be adjusted. Like the simulated ranging smart cable, the actual ranging radio smart cable on the SIG Rascal 110 (configuration one) used the GPS enabled Pixhawk published message and the range from the ranging radios to publish a range and a location of the known SIG Rascal 110 location. The other ranging radio was on the Telemaster.

### 3.2 Analysis

After the data was collected into playable LCM files, they were processed using Scorpion, an estimation filtering framework. Data in each LCM file was extracted as the LCM messages were played back over the network. Using the LCM files and Scorpion, an EKF was used to propagate an aircraft motion model and then update

the estimated states using the collected measurement data.

### 3.2.1 Kinematic Aircraft Motion Model.

The Beard and McLain kinematic aircraft motion model was chosen for this research [10]. This kinematic model was choosen due to its simplistic design, validity towards fixed wing aircraft, and its ability to track wind. The model assumes the fixed wing aircraft flies coordinated turns, meaning there is no lateral acceleration or side slip of the aircraft in the air [10]. Successful use of the kinematic aircraft motion model was demonstrated in [53]. Since the Beard and McLain model did not track altitude of the aircraft, two extra states where added —an altitude and a vertical velocity. The vertical velocity was modeled as a first order Gauss-Markov process with a standard deviation($\sigma$) of 5 m/s and a time constant ($\tau$) of 2 sec [19].

The states of the motion model are:

$$\mathbf{x} = \begin{bmatrix} P_n \\ P_e \\ V_g \\ \chi \\ w_n \\ w_e \\ \psi \\ Alt \\ Alt_{vv} \end{bmatrix} \begin{array}{c} Position\ North\ (m) \\ Position\ East\ (m) \\ Ground\ Speed\ (m/s) \\ Course\ Angle\ (rads) \\ Wind\ North\ (m/s) \\ Wind\ East\ (m/s) \\ Aircraft\ Yaw/Heading\ (rads) \\ Above\ Ground\ Level\ Altitude\ (m) \\ Altitude\ Vertical\ Velocitty\ (m/s) \end{array} \tag{28}$$

and the inputs to the model are:

$$\mathbf{u} = \begin{bmatrix} V_a \\ q \\ r \\ \phi \\ \theta \end{bmatrix} \begin{matrix} Airspeed\ (m/s) \\ Pitch\ Angular\ Rate\ (rads/s) \\ Yaw\ Angular\ Rate\ (rad/s) \\ Roll\ (rads) \\ Pitch\ (rads) \end{matrix} \tag{29}$$

and the derivatives of the state with respect to time are [10]:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} V_g \cos\chi \\ V_g \sin\chi \\ \dfrac{(V_a\cos\psi + w_n)(-V_a\dot{\psi}\sin\psi) + (V_a\sin\psi + w_e)(V_a\dot{\psi}\cos\psi)}{V_g} \\ \dfrac{g}{V_g}\tan\phi\cos(\chi - \psi) \\ 0 \\ 0 \\ q\dfrac{\sin\phi}{\cos\theta} + r\dfrac{\cos\phi}{\cos\theta} \\ \dot{Alt} \\ -\frac{1}{\tau}\dot{Alt} \end{bmatrix} \tag{30}$$

where $g \cong 9.81 m/s^2$ is the gravitational acceleration. The linear propagation model used to calculate $\Phi$, the state transition matrix used in the EKF covariance, is the Jacobian of the nonlinear model in Equation (30) with respect to the states from

Equation (28) as:

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}} = \begin{bmatrix} 0 & 0 & \cos\chi & -V_g \sin\chi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin\chi & -V_g \cos\chi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\dot{V}_g}{V_g} & 0 & \frac{-\dot{\psi} V_a \sin\psi}{V_g} & \frac{\dot{\psi} V_a \cos\psi}{V_g} & \frac{\partial \dot{V}_g}{\partial \psi} & 0 & 0 \\ 0 & 0 & \frac{\partial \dot{\chi}}{\partial V_g} & \frac{\partial \dot{\chi}}{\partial \chi} & 0 & 0 & \frac{\partial \dot{\chi}}{\partial \psi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \tag{31}$$

where:

$$\frac{\partial \dot{V}_g}{\partial \psi} = \frac{-\dot{\psi}(\omega_n \cos\psi + \omega_e \sin\psi}{V_g} \qquad \frac{\partial \dot{\chi}}{\partial V_g} = \frac{-g}{V_g^2} \tan\phi \cos(\chi - \psi)$$

$$\frac{\partial \dot{\chi}}{\partial \chi} = \frac{-g}{V_g} \tan\phi \sin(\chi - \psi) \qquad \frac{\partial \dot{\chi}}{\partial \psi} = \frac{g}{V_g} \tan\phi \sin(\chi - \psi)$$

#### 3.2.1.1 State Propagation.

In order to propagate the motion model the two added altitude state propagation equations were separated from the other seven state propagate equations, mathematically calculated and then combined into the new state estimate. The first seven states

of the model from Equation (28) where propagated according to:

$$\mathbf{x_{MMNEW}} = \begin{bmatrix} V_g \cos\chi \\ V_g \sin\chi \\ \dfrac{(V_a\cos\psi + w_n)(-V_a\dot{\psi}\sin\psi) + (V_a\sin\psi + w_e)(V_a\dot{\psi}\cos\psi)}{V_g} \\ \dfrac{g}{V_g}\tan\phi\cos(\chi-\psi) \\ 0 \\ 0 \\ q\dfrac{\sin\phi}{\cos\theta} + r\dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \Delta t + \mathbf{x_{MMOLD}} \tag{32}$$

where $\mathbf{x_{MM}}$ is only the first seven states of Equation (28).

In the case of the last two states, altitude and altitude vertical velocity, the first order Markov process was propagated using a matrix exponential $\mathbf{\Phi_A}$ of the values in $\mathbf{F}$ from Equation (31) corresponding to the last two states as:

$$\mathbf{F_A} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-1}{\tau} \end{bmatrix} \qquad \mathbf{\Phi} = e^{\mathbf{F_A}\Delta t} \tag{33}$$

where $\mathbf{F_A}$ represents the described portion of $\mathbf{F}$ and $\Delta t$ is the time interval over which the propogation is to occur. The resulting propagation for the last two states was:

$$\mathbf{x_{ANEW}} = \mathbf{\Phi_A} \cdot \mathbf{x_{AOLD}} \tag{34}$$

with $\mathbf{x_{AOLD}}$ being the prior estimate.

$\mathbf{x_{MMNEW}}$ and $\mathbf{x_{ANEW}}$ are then combined to the current predicted state estimate

as:

$$\mathbf{x(t)} = \begin{bmatrix} \mathbf{x_{MMNEW}} \\ \mathbf{x_{ANEW}} \end{bmatrix} \tag{35}$$

#### 3.2.1.2 Propagation Noise.

In order to describe the change in covariance accurately on each state during an EKF propagation, the original seven state motion model and the two added altitude states $\mathbf{Q_d}$ was calculated as $\mathbf{Q_{MMd}}$ and $\mathbf{Q_A}$ respectively. After calculating them individually, they were again combined into a single $\mathbf{Q_d}$. An initial guess for the noise on each input was created and then tuned. The resulting noise or $\sigma$ on each input from Equation (29) was 0.5 m/s for airspeed, 3 deg/sec for pitch angular rate, 3 deg/sec for yaw angular rate, 1 degree for roll, and 1 degree for pitch.

This resulted in a $\mathbf{Q_{Ud}}$ of:

$$\mathbf{Q_{Ud}} = \begin{bmatrix} 0.5^2 * \Delta t & 0 & 0 & 0 & 0 \\ 0 & 3^2 * \Delta t & 0 & 0 & 0 \\ 0 & 0 & 3^2 * \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 * \Delta t & 0 \\ 0 & 0 & 0 & 0 & 1 * \Delta t \end{bmatrix} \tag{36}$$

where the multiplication of $\Delta t$, the difference between the past propagation time step and the current propagation time step, is a first order approximation to discretize the estimated $\sigma$ on each input. The noise on the inputs was then related to each state using [19]:

$$\mathbf{Q_{MMd}} = \mathbf{B Q_{Ud} B^T} \tag{37}$$

where $\mathbf{B}$ is the Jacobian of the nonlinear model in Equation (30) with respect to the inputs from Equation (29) calculated as:

$$\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{EQ1 \cdot EQ2}{EQ4} & \frac{V_a \sin(\phi) \cdot EQ1}{EQ4} & \frac{V_a \cos(\phi) \cdot EQ1}{EQ4} & \frac{V_a EQ1 \cdot EQ3}{EQ4} & \frac{V_a \sin(\theta) \cdot EQ1 \cdot EQ2}{V_g (\cos(\theta))^2} \\ 0 & 0 & 0 & \frac{g \cos(\chi - \psi)}{V_g (\cos(\phi))^2} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} & \frac{EQ3}{\cos(\theta)} & \frac{\sin(\theta) \cdot EQ2}{(\cos(\theta))^2} \end{bmatrix} \tag{38}$$

where:

$$EQ1 = W_e \cos(\psi) - W_n \sin(\psi)$$

$$EQ2 = r \cos(\phi) + q \sin(\phi)$$

$$EQ3 = q \cos(\phi) - r \sin(phi)$$

$$EQ4 = V_g \cos(\theta)$$

The noise on the altitude and altitude vertical velocity states:

$$\mathbf{Q_A} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{2\sigma^2}{\tau} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{2(5)^2}{2} \end{bmatrix} \tag{39}$$

was then discretized using:

$$\mathbf{Q_{Ad}} = \frac{\mathbf{Q_A} + \mathbf{\Phi} \mathbf{Q_A} \mathbf{\Phi}^T}{2} \tag{40}$$

where $\mathbf{\Phi}$ is as defined in Equation (33).

$\mathbf{Q_{MMd}}$ and $\mathbf{Q_{Ad}}$ are then combined into $\mathbf{Q_d}$ as:

$$\mathbf{Q_d} = \begin{bmatrix} & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & \mathbf{Q_{MMd}^{7x7}} & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \mathbf{Q_{Ad}^{2x2}} \end{bmatrix} \tag{41}$$

### 3.2.2 EKF Measurements.

In order to update the current state estimate and covariance, four measurements were used to help the EKF track the truth state estimate — the heading measurement from the non-GPS autopilot magnetometer, an altitude measurement from the non-GPS autopilot barometer, a range measurement, and a VO measurement. Every measurement had a $\mathbf{H}$ which described how the measurement was applied to the states and a $R$ which described the confidence in the measurement. $R$ can be described in terms of $\sigma$, where $\sigma$ is the standard deviation in the units of the measurement, and $R$ is equal to $\sigma^2$

#### 3.2.2.1 Heading Measurement.

The heading measurement was taken from the non-GPS Pixhawk autopilot magnetometer as to make sure the result of the measurement was not related to any GPS

data. The heading measurement could be directly incorporated into the heading state $\psi$ of Equation (28) as:

$$\mathbf{H_{Heading}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{42}$$

After tuning, a $\sigma$ of 20 degrees was chosen resulting in:

$$R_{Heading} = 20^2 \tag{43}$$

#### 3.2.2.2 Altitude Measurement.

The altitude measurement was taken from the non-GPS Pixhawk autopilot barometer to make sure the result of the measurement was not related to any GPS data. The barometer measurement could be directly incorporated into the altitude state ALT of Equation (28) as:

$$\mathbf{H_{Altitude}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{44}$$

After tuning, a $\sigma$ of 8 meters was chosen resulting in:

$$R_{Altitude} = 8^2 \tag{45}$$

#### 3.2.2.3 Range Measurement.

The range measurement came in two parts with one being the range in meters and the second part a location the range was from. The range location was converted to the EKF estimated NEU frame where the origin was the starting location of the

recorded data. The non-linear range measurement $\mathbf{h_{Range}}$ is:

$$\mathbf{h_{Range}} = \sqrt{(P_n - P_{nM})^2 + (P_e - P_{eM})^2 + (Alt - Alt_M)^2} \tag{46}$$

where $P_{nM}$, $P_{eM}$ and $Alt_M$ are the NEU position of the measurement. $\mathbf{h_{Range}}$ was linearized using a Jacobian of $\mathbf{h_{Range}}$ with respect to the states $\mathbf{x}$ in Equation (28) as:

$$\mathbf{H_{Range}} = \frac{\delta \mathbf{h_{Range}}}{\delta \mathbf{X}} = \begin{bmatrix} \frac{P_n - P_{nM}}{\mathbf{h_{Range}}} & \frac{P_e - P_{eM}}{\mathbf{h_{Range}}} & 0 & 0 & 0 & 0 & 0 & \frac{Alt - Alt_M}{\mathbf{h_{Range}}} & 0 \end{bmatrix} \tag{47}$$

The ENSCO radios choosen to provide a real range were still in the development phase during testing and on average produced range measurements accurate to 1m and below as tested. The physical distance performance of the ranging radios after being fully developed is expected to be greater then a kilometer line of sight [9]. Given the simulated ranges on average were 3500 m or greater in distance an estimated $\sigma$ of 5 m for the simulated range was used as:

$$R_{Range} = 5^2 \tag{48}$$

The actual ranging radios data was not incorporated into the results and therefore a covariance for the actual ranging radio data was not used. In this research two simulated ranges from separate locations were investigated. Both simulated range measurements used the same equations above.

#### 3.2.2.4 VO Measurement.

The VO measurement used the 3Hz images from the downward facing camera and the AKAZE feature detection and kNN matching algorithm to provide an update

to the ground speed. Three VO measurements were used, an inertial aided VO measurement, an OpenCV rotation estimated VO and a simulated VO measurement. Using the feature detection and matching algorithm two images, an altitude, and an interpolated aircraft attitude combined with a camera to body rotation for each image were used to calculate velocities in the NEU frame. The OpenCV rotation estimated VO used OpenCV to estimate the rotation between the images versus the interpolated aircraft attitudes. The calculated velocities could then be applied as updates to the ground speed($V_g$) and course angle($\chi$) as a vector magnitude ($z_{mag}$) and a vector direction ($z_{dir}$) as:

$$z_{mag} = \sqrt{V_x^2 + V_y^2} \qquad\qquad z_{dir} = atan2(V_y, V_x) \tag{49}$$

where atan2 was used to return the vector direction in the appropriate quadrant. $V_x$ and $V_y$ were the VO calculated velocities in the NEU along the north and east axis respectively. During analysis the inertial VO measurement was found to contain a lot of noise resulting in $z_{dir}$ not being incorporated as a measurement into the EKF. The VO measurement for this research incorporated $z_{mag}$ being applied to the ground speed ($\chi$) as:

$$\mathbf{H_{VO}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{50}$$

with a $\sigma$ of 10 meters after tuning:

$$R_{VO} = 10^2 \tag{51}$$

Given the inertial aided VO measurement contained enough noise to make the direction of the velocity vector an unusable measurement, an OpenCV rotation estimated VO and a simulated VO measurement were created. The VO measurement

updates was limited to when the aircraft was not in a turn. The VO update was also set to skip two images before comparing the images. Skipping images and limiting the update based on the aircraft roll helped reduce the error in the estimated VO NEU velocities.

The simulated VO measurement was similar to the actual VO measurement as the update was only applied to the ground speed. The simulated VO measurement applied a Gaussian measuremnt noise to the true velocity magnitude as a $\sigma$ of 1 m/s.

### 3.2.3 Scorpion.

Scorpion is a Bayesian estimation software package that was used to process the EKF equations. Scorpion has the ability to handle a variety of software languages including Kotlin, the language chosen for this project [21]. In order for Scorpion to process the EKF equations the state dynamics and measurement processors had to be written. A state dynamics model was written for the aircraft motion model, and a measurement processor was written for each of the four measurements. A processor was written to handle the LCM messages so that Scorpion could propagate and update the state estimates and covariance upon receiving the LCM measurements. As an example, when Scorpion received a measurement update, the current estimate and covariance were propagated to the measurements time using the state dynamics and then the update was applied.

### 3.2.4 Flight Tests.

The original planned two aircraft flight test included the Telemaster and configuration one of the SIG Rascal 110, where both aircraft would be flying at the same time. The original flight test was not successful due to an electronic failure in the RC radio system. A second single aircraft test involving the SIG Rascal 110 in configuration

two was conducted a few weeks later.

#### 3.2.4.1 Two Aircraft Test.

The two aircraft test was setup to simulated a higher flying aircraft with GPS sending range measurements to the lower flying GPS-denied aircraft below. The Pixhawk used to collect test data on the lower flying aircraft was GPS-denied by disconnecting the autopilot GPS antenna. The higher flying SIG Rascal 110 was used to carry the other ranging radio and report it's current GPS position so that the lower flying Telemaster could record the data as a range measurement update. The SIG Rascal 110 would loiter at 305 m in altitude at a location off to the side of the race track pattern the Telemaster was flying. Given the VO measurement being taken by the Telemaster, the Telemaster altitude would have been varied to capture a variety of ground imagery densities for the image feature detection algorithim. Many specific details of the two aircraft testing plan are left out as the electronic failure suffered by the Telemaster resulted in very little usable research data.

#### 3.2.4.2 One Aircraft Test.

The SIG Rascal 110 was rebuilt into configuration two, where a second autopilot was added and the GPS was disconnected from this autopilot. The camera was also mounted into the belly of the aircraft to preserve the VO capability of the data collection. Instead of actual ranging data, a simulated range was created based on a simulated aircraft loiter trajectory at a designated altitude and an offset GPS location. The SIG Rascal 110 flew a clockwise race track box like pattern (200 m x 600 m) at WPAFB, OH at an altitude of 137 m. Restrictions on the airspace limited the testing to a ceiling of 137m. Figure 14 is a screen shot from Mission planner that shows the location and box like pattern the SIG Rascal 110 flew. Data was collected for the

entire flight and then split out into two LCM logs for post processing.

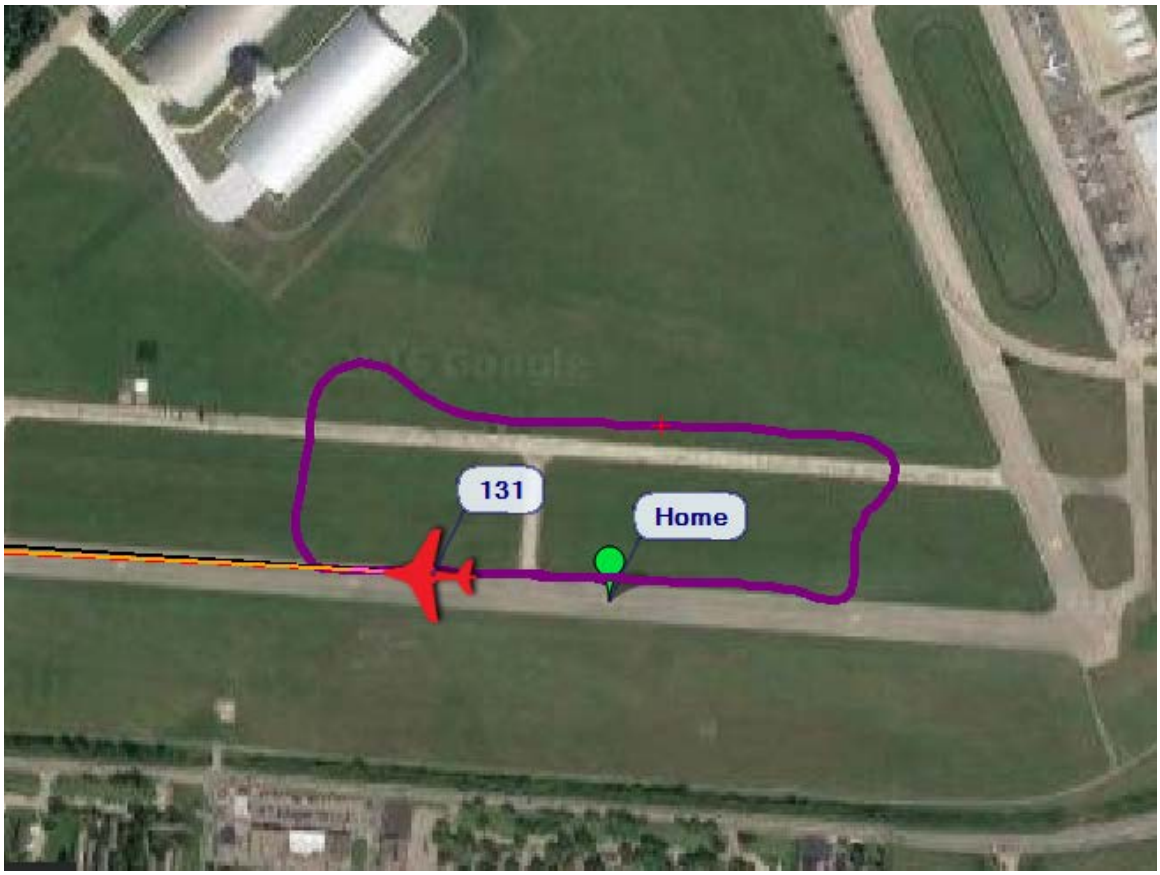



**Figure 14. One Aircraft Test Flight Path at WPAFB, OH**

## 3.3 Chapter Summary

In this chapter the hardware required to conduct this research was described. Multiple payload configurations across two RC aircraft platforms using smart cables and LCM were used to collect flight test data. A description of the Beard and McLain aircraft motion models was defined along with all of the sensors, either built or simulated, that were used for processing the collected flight data.

# IV. Chapter 4 - Results

This chapter discusses the results from the EKF using a combination of the motion model and sensors. For all scenarios, an on-board barometer and magnetometer sensor were used to update the altitude and heading states, respectively. The motion model was always used to propagate the state, and the VO and ranging sensor updates were varied on or off. Two data sets were used called "oneloop", a single oval loop, and "longloop", a longer flight loop. These concepts are described in more detail in the sections that follow.

## 4.1 Data Sets

Two data sets were selected from the larger data collect in order to compare between a more stable, single loop flight and a longer, less stable, with more turns, flight. The differing data sets were used to compare an aircraft generally flying in an oval versus an aircraft changing altitudes and not following a consistent oval pattern.

### 4.1.1 Oneloop Data.

The oneloop scenario consisted of autopilot only control flying a predetermined clockwise direction of four waypoints placed in a rectangular pattern. AGL altitude for the oneloop scenario varied from 124 m to 139 m and was controlled by the autopilot. Figure 15 shows a top down view of the oneloop truth position and the varying altitude sampled from the GPS enabled Pixhawk over the course of the data collect. This flight pattern allowed for longer level straightaways and overall less variance in the attitude of the aircraft than the longloop case (described next). The total flight time for oneloop was 62 seconds. The aircraft data set starts at (0 m North, 0 m East) and flies clockwise, initially heading west down the negative east

axis.

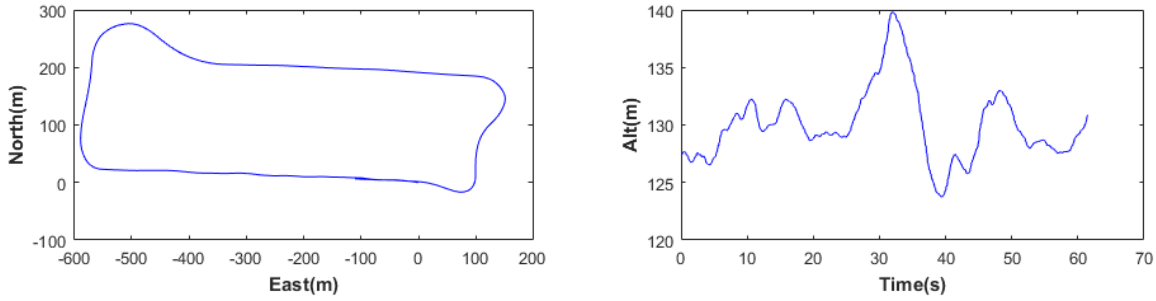



**Figure 15. Oneloop True Position and Altitude**

### 4.1.2 Longloop Data.

The longloop scenario consisted of a combination of autopilot and manual RC control. The aircraft was under manual RC control during take off and then switched between manual and autopilot control multiple times. The aircraft flew a more irregular pattern than the single oneloop scenario, causing the attitude of the aircraft variance to be greater. The aircraft, in contrast to the oneloop case, did not fly long level straightaways over the 128 second flight. Truth position data can be seen in Figure 16. The aircraft started at (0 m North, 0 m East) heading west before making a right banking turn to the east.

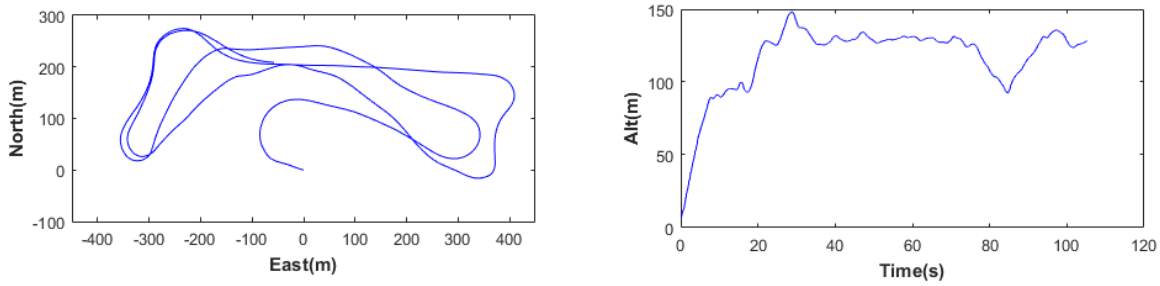



**Figure 16. Longloop True Position and Altitude**

## 4.2 Motion Model Analysis

As discussed in Chapter 3, the Beard and McLain aircraft motion model was used [10], and two additional states were added to allow the motion model to track altitude. First, an analysis of the motion model was conducted to prove the motion model worked and could properly estimate each state. Figure 17 shows all nine states of the motion model using the oneloop data and no additional measurement updates beyond the standard heading and altitude updates. The $P_n$, $P_e$, altitude, ground speed, course angle, yaw, and altitude vertical velocity states are error plots of the estimate minus the truth. The wind north ($\omega_n$), wind east ($\omega_e$), and Altitude vertical velocity are the filters estimated state as no truth data was collected.

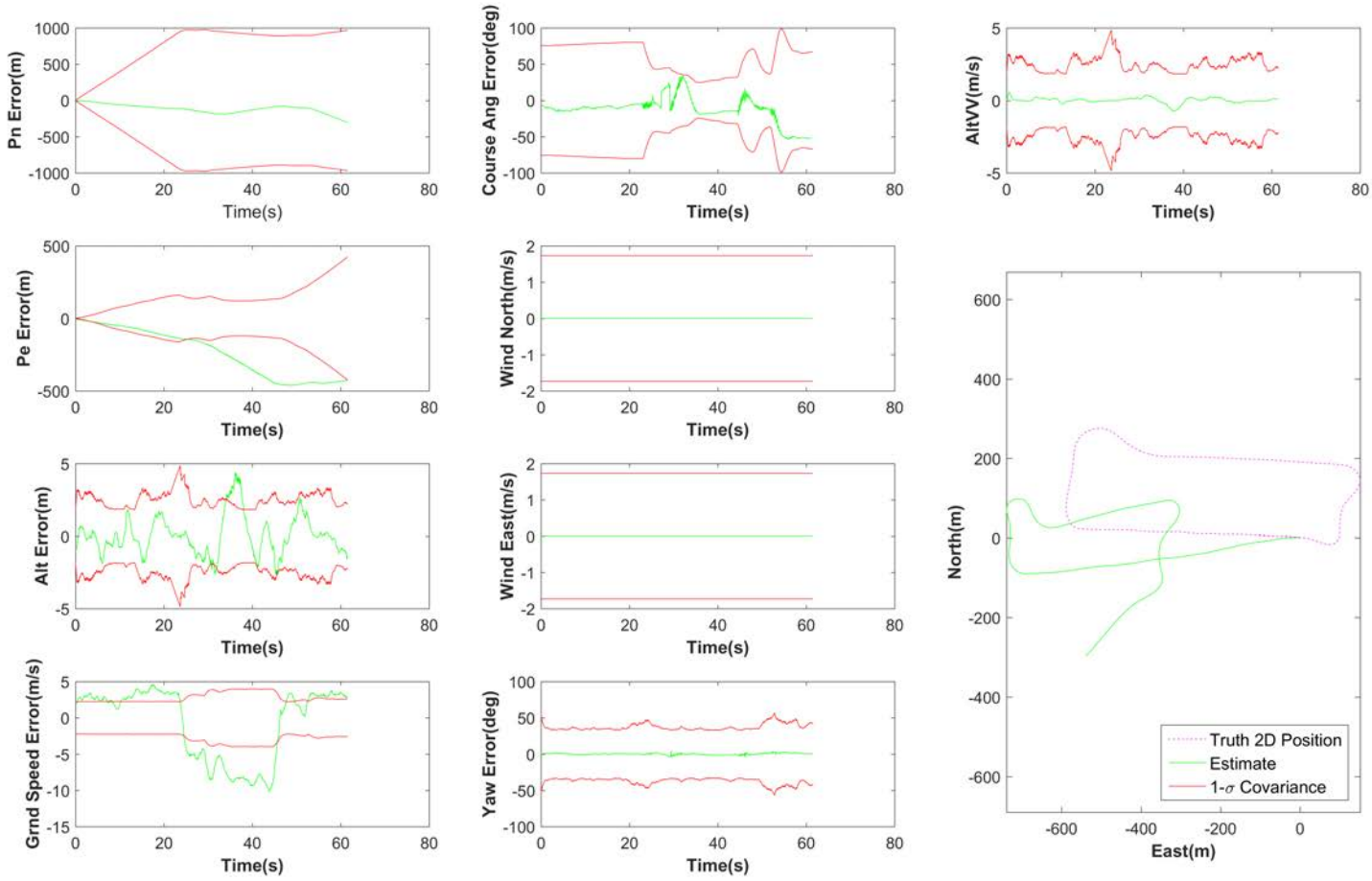



**Figure 17. Filter Performance Using Motion Model Only (oneloop)**

In this case the only updates the EKF was receiving was from the altitude and heading measurements, which directly affect the altitude and heading state. The initial state estimates were set to the current available non-GPS Pixhawk autopilot data at the start of the oneloop or longloop data set. The initial $P_n$ and $P_e$ states were set to 0m as the origin of the NEU frame was set to the initial GPS position from the GPS-enabled Pixhawk. The NEU origin AGL altitude was set to zero to match the Pixhawks calibrated zero AGL. Initial values assumed no wind, so the yaw and course angle were set equal to the Pixhawk's current heading, and the ground

speed was set to the current airspeed. As seen at the bottom right of Figure 17, the estimated 2D position tracks along the initial heading at the initial ground speed. Since no range or VO measurements are incorporated in this case, the EKF cannot estimate the wind and solely relies on the non-GPS Pixhawk attitude and airspeed to update the ground speed and course angle. The end result is the estimated position drifts away from the truth but maintains a similar flight pattern as the oneloop truth data, as seen in the bottom right plot in Figure 17.

Initial covariances for each state in Table 1 were used where all cross covariances where zero.

**Table 1. Initial Uncertainties for the Motion Model States**

| State | Initial Covariance (1-$\sigma$) |
|---|---|
| $P_n$ | 5 m |
| $P_e$ | 5 m |
| $V_g$ | 5 m/s |
| $\chi$ | 100 degs |
| $\omega_n$ | 3 m/s |
| $\omega_e$ | 3 m/s |
| $\psi$ | 64 degs |
| Alt | 3 m |
| $Alt_{vv}$ | $\sqrt{\frac{2(5)^2}{2}}$ |

The filter appears to be tuned properly in Figure 17 as the estimates generally fall within the covariance. After about 23 seconds the aircraft made a right hand turn which can be seen reducing the covariance in the course angle ($\chi$) and the covariance in the position east ($P_e$). After the 25 second mark, the filter begins to underestimate the error on $P_n$ which makes the filter believe the $P_n$ estimate is more accurate then it really is. Since the filter was tuned while incorporating the measurements, discussed

in future sections, additional filter tuning maybe required when using the motion model, barometer, and magnetometer alone.

To test the motion model's ability to track wind, a range update was incorporated to allow the EKF to use the range update to estimate an east and north wind. On the day of the test, the wind was blowing from the south west to the north east on average at 6-7 m/s, based on a portable ground weather station. Figure 18 shows the motion models ability to estimate wind when a range update is incorporated. The motion model using the EKF estimated the wind with a magnitude of $\sqrt{\omega_n^2 + \omega_e^2} = \sqrt{4^2 + 5^2} = 6.4$ m/s and direction of $\tan^{-1}(\omega_n/\omega_e) = \tan^{-1}(5/4) = 51$ degrees from 0 degrees north in the north east direction. The motion model wind estimate trended with what was observed on the flight test day.

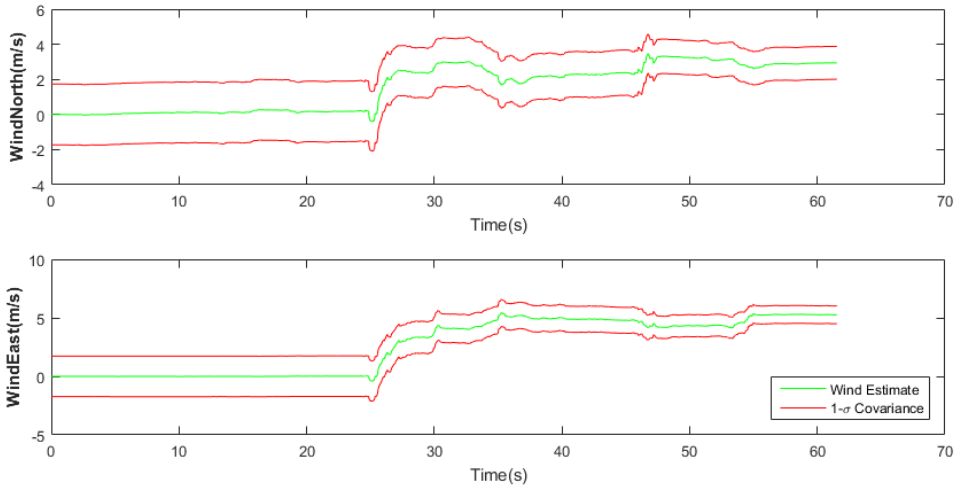



**Figure 18. Example of Motion Model Wind Estimation When incorporating Ranging Measuremeent (oneloop)**

When one range was used, the simulated range was based off of an aircraft loitering about a point 3000 m east (0 m north, 3000 m east) at an altitude of 2000 meters. The radius of the loiter was 1000 m and the ground speed was set to 120 m/s. The aircraft started at the top of the circular loiter in the northern most point and continued to travel clockwise about the circle. When two ranges were used, a second range was

added with the same parameters except the loiter location was placed at 3000 m north (3000 m north, 0 m east). The second range loitering aircraft started to the north of the SUAS. Figure 19 is an illustration depicting the first loiter range and the added second range loiter used when both range sensors were used.

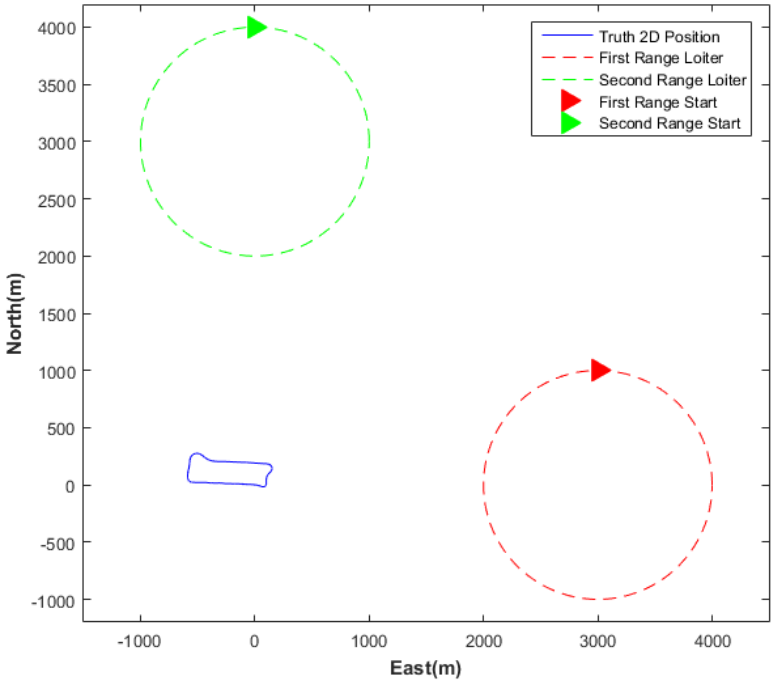



**Figure 19. Graphical Illustration of the Simulated Aircraft Providing Range Updates**

For the case of the ranges, the loitering aircraft made it around the circle 1.18 times in the 62 second run time on the oneloop data set. The simulated loitering aircraft made it around the circle 2.44 times for the 128 second longloop data set.

## 4.3 Data Rates

The Pixhawk autopilots were limited to 10Hz data output for the GPS-denied Pixhawk and 4Hz output for the GPS-enabled Pixhawk controlling the aircraft. The data rates were selected based on performance seen during ground testing of the smart

cables. The flight Pixhawk was set to 4Hz to reduce the chance of an in flight failure due to over clocking the Pixhawk autopilot. The attitude, roll rates, airspeed and barometer measurements were received at 10Hz from the GPS-denied Pixhawk. The truth data, including the GPS position and ground speed, were collected at 4Hz from the GPS enabled Pixhawk. The truth course angle was calculated using the change in GPS position. The simulated range updates were incorporated at 4Hz as this was the data update rate of the GPS enabled Pixhawk.

### 4.4 EKF Position Error Results

A root mean square (RMS) was used to to quantify the error in the north ($P_n$) and east ($P_e$) axis as:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}} \tag{52}$$

where $x$ represents $P_n$ or $P_e$ and the subscript $i$ represents the iteration of $P_n$ or $P_e$ from the first data point to $n$, the total number of $P_n$ or $P_e$ data points. For quantifying two dimensional errors, a 2x distance root mean square (2DRMS) was calculated as:

$$2DRMS = 2\sqrt{\frac{\sum_{i=1}^{n}(P_{n_i}^2 + P_{e_i}^2)}{n}} \tag{53}$$

having a 95% confidence in position.

Results in this section are presented in table format. To reduce the size of the results section all of the plots of the EKF states for each simulation can be found in the Appendix.

## 4.5 Sensor Nomenclature and Description

In the section that follows, results are referenced based on the data set used (oneloop or longloop) and what sensors were used to track the truth data. All data sets used the motion model, labeled as "mm". After analyzing the results it was found that using the inertial (Pixhawk) aided VO did not produce an accurate VO measurement update. A secondary VO measurement was created where the rotation between the images was estimated using OpenCV (the recoverPose function) versus using the attitude information from the Pixhawk autopilot. The OpenCV rotation estimation VO measurement is labeled "$VO_R$". The inertial aided VO sensor is labeled "VO" and the Simulated VO measurement is labeled "SimVO". When only one range was used the range sensor is labeled "R". When Two range sensors were used, this was labeled "RR". As an example, if the data set used was oneloop and a single range and the simulated VO were used as measurements the nomenclature for the filter estimates would be "oneloopmm-SimVOR". As another example, using longloop, where the measurements included two ranges, and the VO it would be labeled '"longloopmm-VORR". For all of the scenarios using a single range, "R", or two range updates, "RR", the measurement standard deviation for each range was set to 5 m. In the cases using the actual VO measurements, "VO" and "$VO_R$", to update the ground speed ($\chi$) the standard deviation was set to 10 m/s. The scenarios using the simulated VO, "SimVO", the standard deviation was set to 1 m/s to simulate a more accurate, than the actual VO measurements, SimVO update to the ground speed ($\chi$).

### 4.5.1 Oneloop Results.

Table 2 depicts the 2DRMS results obtained from a variety of sensor updates using the oneloop data set. For all simulations listed, the motion model was used to propagate the filter state estimates, and the sensors were used by the filter to update

the state estimates. Simulations were varied by using no VO, VO, $VO_R$ or SimVO where each case was then broken out into not using a range followed by using a single range and then using two ranges. The simulations were varied in this way to obtain results on all varying measurement scenarios.

**Table 2. Oneloop Results Across Test Simulations**

| Simulation (Data-Sensors) | $P_n$ RMS (m) | $P_e$ RMS (m) | 2DRMS (m) |
|---|---|---|---|
| oneloopmm | 130.59 | 285.56 | 628.01 |
| oneloopmm-R | 15.23 | 43.76 | 92.66 |
| oneloopmm-RR | 4.48 | 3.54 | 11.41 |
| oneloopmm-VO | 131.19 | 272.33 | 604.57 |
| oneloopmm-VOR | 42.07 | 110.72 | 236.88 |
| oneloopmm-VORR | 3.27 | 3.18 | 9.13 |
| oneloopmm-$VO_R$ | 167.81 | 139.81 | 436.85 |
| oneloopmm-$VO_R$R | 10.73 | 39.75 | 82.35 |
| oneloopmm-$VO_R$RR | 3.05 | 2.56 | 7.96 |
| oneloopmm-SimVO | 129.58 | 124.00 | 358.72 |
| oneloopmm-SimVOR | 12.83 | 35.45 | 75.40 |
| oneloopmm-SimVORR | 3.96 | 3.95 | 11.19 |

General trends seen in Table 2 show that adding a range measurement reduced the 2DRMS of both the motion model alone and the motion model with the VO. Overall it can be stated that adding a range measurement can significantly contain the drift and inherent errors in the selected motion model and VO algorithm. Adding a second range measurement greatly contains the drift and inherent errors in the motion model and VO.

As was stated before, the inertial aided VO measurement, using the attitudes

from the autopilot, did not perform as expected leading to the creation of the $VO_R$ and SimVO measurements. When comparing oneloopmm-R with oneloopmm-VOR it can be seen that adding the inertially aided VO measurement made the 2DRMS error worse. This was an indication that the inertially aided VO measurement was not working properly across the entire data set. A discussion of possible errors in the inertially aided VO and a comparison to the OpenCV estimated image rotation $VO_R$ will be presented later in the chapter.

When comparing oneloopmm with oneloopmm-$VO_R$ and oneloopmm-SimVO it can be seen that adding a visual odometry measurement to the motion model allows the filter to more accurately estimate the SUAS position. This is likely due to the ground speed in the oneloopmm-$VO_R$ and oneloopmm-SimVO cases being more accurate then the oneloopmm case as the VO measurements were used to update the ground speed. To compare the $VO_R$ and SimVO measurements, the RMS of the ground speed error for the oneloopmm-VO was 5.0 m/s while the RMS for the oneloopmm-SimVO was 1.44 m/s. Since both the oneloopmm-$VO_R$ and oneloopmm-SimVO relied solely on the motion model for a course angle the more accurate oneloopmm-SimVO estimated ground speed allowed the oneloopmm-SimVO case to track the truth position sightly better then the oneloopmm-$VO_R$ case.

When a range measurement was added the 2DRMS error decreased when comparing oneloopmm with oneloopmm-R, and oneloopmm-$VO_R$ with oneloopmm-$VO_R$R, and oneloopmm-SimVO with oneloopmm-SimVOR. The range update was able to provide information from an absolute position, which added observability into the position states, which then fed into the other states of the EKF. When comparing the standard deviation of the ground speed and course angle for oneloopmm-R, oneloopmm-VOR, and oneloopmm-SimVOR seen in Table 3 the more accurate SimVO measurement ground speed allowed the EKF to achieve a better position

estimate.

**Table 3. VO & Range Results**

| Simulation (Data-Sensors) | Ground Speed ($V_g$) RMS (m/s) | Course Angle ($\chi$) RMS (deg) | 2DRMS (m) |
|---|---|---|---|
| oneloopmm-R | 2.51 | 10.00 | 92.66 |
| oneloopmm-$VO_R$R | 2.14 | 10.66 | 82.35 |
| oneloopmm-SimVOR | 1.17 | 9.14 | 75.40 |

When comparing the 2DRMS error results from the oneloopmm-$VO_R$RR (7.96 m) and oneloopmm-SimVORR (10.78 m) from Table 2, the non-simulated $VO_R$ measurement EKF preformed better in estimating the position of the SUAS. This was likely due to the $VO_R$ measurement standard deviation being 1 order of magnitude higher at 10 m/s while the simulated VO measurement standard deviation was 1 m/s. With both range measurements on, the EKF was able to weight more highly the range measurement information in the oneloopmm-VORR case, because less confidence was placed on the $VO_R$ measurement. When the simulated VO measurement, having a smaller standard deviation, was used the ground speed estimate in the filters view was more accurate, placing more confidence on the current position estimate versus the new information the ranges were providing.

### 4.5.2 Longloop Results.

The longloop results seen in Table 4 show adding a range measurement while the SUAS is not flying a scripted single loop can also significantly reduce the 2DRMS error when comparing longloopmm with longloopmm-R, longloopmm-$VO_R$ with longloopmm-$VO_R$R, and longloopmm-SimVO with longloopmm-SimVOR. As was expressed earlier, the "VO" measurement did not preform as expected providing inaccurate VO

measurement updates. In the longloopmm-R, and longloopmm-VOR cases from Table 4, it can be seen adding the VO measurement reduced the 2DRMS value. Since the VO measuramnt after further analysis was known to be providing inaccurate measurement updates it can be noted that just because the 2DRMS appears to have reduced does not mean the measurement was working correctly. When comparing longloopmm-VOR with the working $VO_R$ measurement it can been seen that the $VO_R$ measurement reduces the 2DRMS error when compared to longloopmm-R and longloopmm-VOR. Differences in longloopmm-R and longloopmm-VOR 2DRMS errors, where the inaccurate VO measurement appears to be working, maybe caused by the 5 m standard deviation on the range measurement providing varying position information into the filter states. Since the ranges were simulated a random number of strength 5 was added to the truth range and would result in differing levels of accurate measurements in each simulation run.

**Table 4. Longloop Results Across Test Simulations**

| Simulation (Data-Sensors) | $P_n$ RMS (m) | $P_e$ RMS (m) | 2DRMS (m) |
|---|---|---|---|
| longloopmm | 1265.93 | 468.54 | 2699.71 |
| longloopmm-R | 21.74 | 84.13 | 173.78 |
| longloopmm-RR | 5.60 | 3.85 | 13.60 |
| longloopmm-VO | 1058.06 | 219.95 | 2161.35 |
| longloopmm-VOR | 19.37 | 59.28 | 124.73 |
| longloopmm-VORR | 4.94 | 4.02 | 12.74 |
| oneloopmm-$VO_R$ | 618.06 | 1017.11 | 2380.34 |
| oneloopmm-$VO_R$R | 13.96 | 39.50 | 83.79 |
| oneloopmm-$VO_R$RR | 4.73 | 3.61 | 11.90 |
| longloopmm-SimVO | 328.70 | 218.11 | 788.97 |
| longloopmm-SimVOR | 20.25 | 69.65 | 145.07 |
| longloopmm-SimVORR | 4.82 | 3.96 | 12.49 |

In each case of longloop, when adding a single range measurement to the filter estimates that did not use a range measurement, the 2DRMS error reduced significantly. Adding a second range measurement, as seen in the longloopmm-RR, longloopmm-$VO_R$RR, and longloopmm-SimVORR simulations, the 2DRMS error reduced further when compared to the longloop simulations using only one range. Figure 20 shows the increasing certainty of the EKFs ability to predict the truth horizontal position among simulations longloopmm-SimVO, longloopmm-SimVOR, and longloopmm-SimVORR.

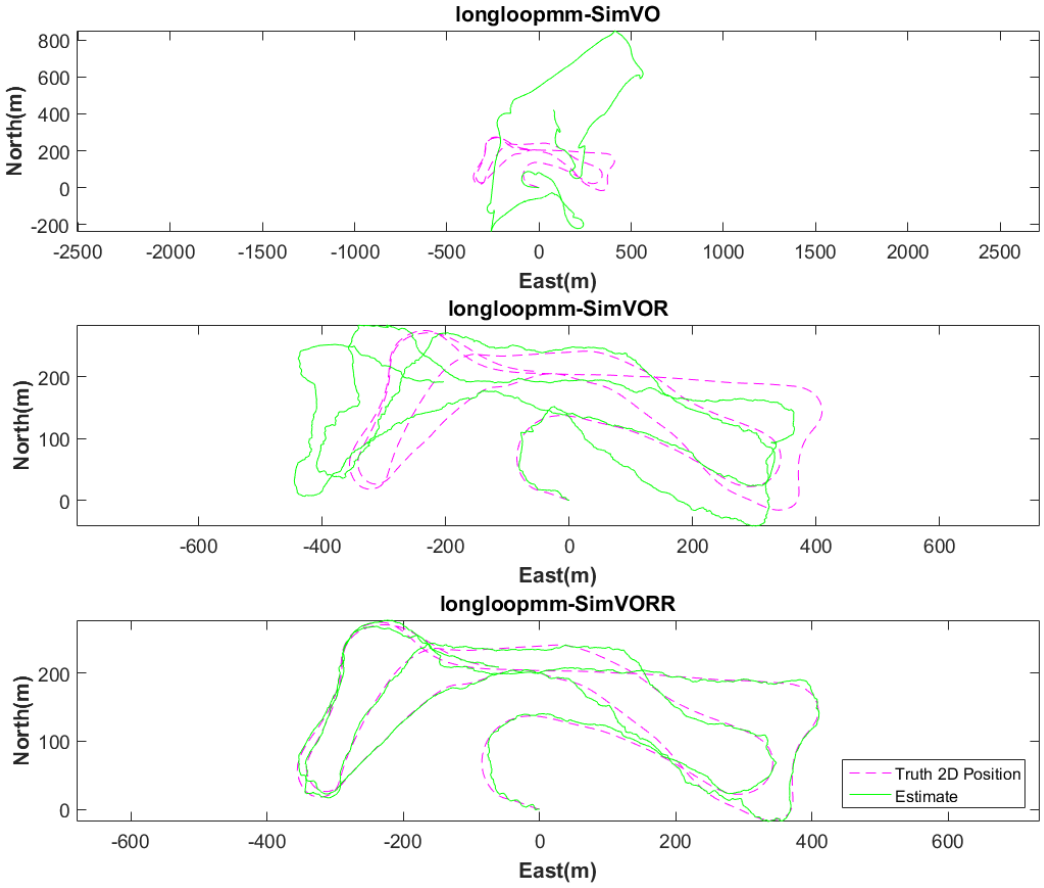



**Figure 20. Horizontal Position Estimates of Longloop with SimVo and Ranges**

## 4.6 Varying Range and SimVO Covariances

An evaluation was done to analyze the effect of varying the range covariances. Varying the covariances for the ranges simulates how the accuracy of a range measurement can affect the 2DRMS error. Table 6 depicts the results for a single and double range measurement when varying the standard deviation from 10 m to 2.5 m. The added noise to the simulated range was also varied by the same standard deviation.

**Table 5. Effect of Varying Range Standard Deviation on 2DRMS Error (m)**

| Simulation | $\sigma_R = 10m$ | $\sigma_R = 5m$ | $\sigma_R = 2.5m$ |
|---|---|---|---|
| oneloopmm-R | 202.38 | 92.66 | 61.66 |
| oneloopmm-RR | 23.27 | 11.41 | 6.58 |

Using a 10 m standard deviation, two times larger then the 5 m used for the bulk of this research, increased the 2DRMS error for the single and double range simulations. This was expected as the higher standard deviation reduced the quality of the absolute position data the ranges offered. Reducing the standard deviation to 2.5 m meant that the ranges were more accurate and were able to provide better range measurements into the EKF. This resulted in a more accurate 2DRMS error for the range and double range case. Of note, by reducing the double range case standard deviation in half the 2DRMS error also reduced by half. The reduction in 2DRMS was even greater for the single range case when going from 10 m to 5 m and reduced by another 33% when the standard deviation was 2.5 m. From this data, when using standard deviations that range from 10 m to 2.5 m and the motion model, it can be seen that it is better to have two ranges with a standard deviation of 10 m then a higher quality single range with a standard deviation of 2.5 m.

Next, the standard deviation of the SimVO was varied from 5 m/s to 0.1 m/s where the noise was also varied on the measurement. The standard deviation of the SimVO measurement represented the accuracy of the ground speed the SimVO was able to provide. Table 6 displays how the 2DRMS error is affected when varying the SimVO standard deviation while the range measurements are also being incorporated into the EKF.

**Table 6. Effect of Varying SimVO Standard Deviation on 2DRMS Error (m)**

| Simulations | $\sigma_{SimVO} = 5\ (m/s)$ | $\sigma_{SimVO} = 1\ (m/s)$ | $\sigma_{SimVO} = 0.1\ (m/s)$ |
|---|---|---|---|
| oneloopmm-SimVOR | 372.38 | 75.40 | 45.79 |
| oneloopmm-SimVORR | 10.20 | 11.19 | 8.06 |

For the single range measurement it can be seen that a more accurate SimVO measurement reduced the 2DRMS error from 372 m to 46 m. In the double range case the SimVO 2DRMS error went from 10 m to 8 m as the standard deviation decreased showing that the range measurements were the dominate factor in the 2DRMS error. The minimal change in the 2DRMS error when comparing the two SimVO double range cases when $\sigma_{SimVO} = 5\ (m/s)$ and $\sigma_{SimVO} = 1\ (m/s)$ hints at the idea the motion model 2DRMS error, using the EKF, does not always become smaller when the ground speed ($\chi$) state is more accurate, depending on what other measurements are available.

## 4.7 Error Analysis

An error analysis of the results was conducted to present to the reader possible sources of error that contributed to the final 2DRMS error values in each case. Some of the errors were known, while some of the errors are mentioned and hypothesized upon but could not be definitively proven.

### 4.7.1 EKF Tuning & Ranges.

Given that the estimation was done using an EKF, errors in the proposed data sets could be partially attributed to EKF errors due to tuning as the filter was tuned to a certain degree. Continuing to tune the EKF would not have changed the results dramatically or changed the effect of the range measurement on the montion model or

VO simulations, but could have reduced the 2DRMS error some. Since the ranges and SimVO simulations were simulated the measurement covariances for simulations in Table 2 and Table 4 could have been reduced producing a more accurate measurement and likely producing a decrease in the 2DRMS error.

### 4.7.2 Motion Model Assumption.

The motion model in this research assumed coordinated flight where zero side slip is achieved during turns. In a coordinated turn the main force on the aircraft is straight down the aircraft's z-axis in the body frame [10]. In this research, the aircraft was assumed to always adhere to coordinated flight. If this assumption of coordinated flight was broken, this would have attributed errors into the Yaw ($\psi$) state as the attitude and angular rates in:

$$\dot{\psi} = q\frac{\sin\phi}{\cos\theta} + r\frac{\cos\phi}{\cos\theta} \tag{54}$$

from Equation (30), would have incorrectly predicted the true change in yaw. The error in the aircraft yaw would have then been translated into the other states of the motion model.

### 4.7.3 Camera Alignment.

In this research, the camera was mounted inside the aircraft pointing down the z-axis ($Z_B$) of the aircraft's body frame. The x-axis ($X_C$) of the camera frame was aligned with the y-axis ($Y_B$) of the aircraft body frame and the y-axis ($Y_C$) of the camera frame was aligned with the negative x-axis ($-X_B$) of the aircraft body frame. If the aircraft were traveling wings level in a forward direction, the images taken by the camera would make the aircraft appear to be moving forward as features in the image would be moving down the y-axis in the camera frame. Figure 21 depicts how

the camera and aircraft body frames were aligned.

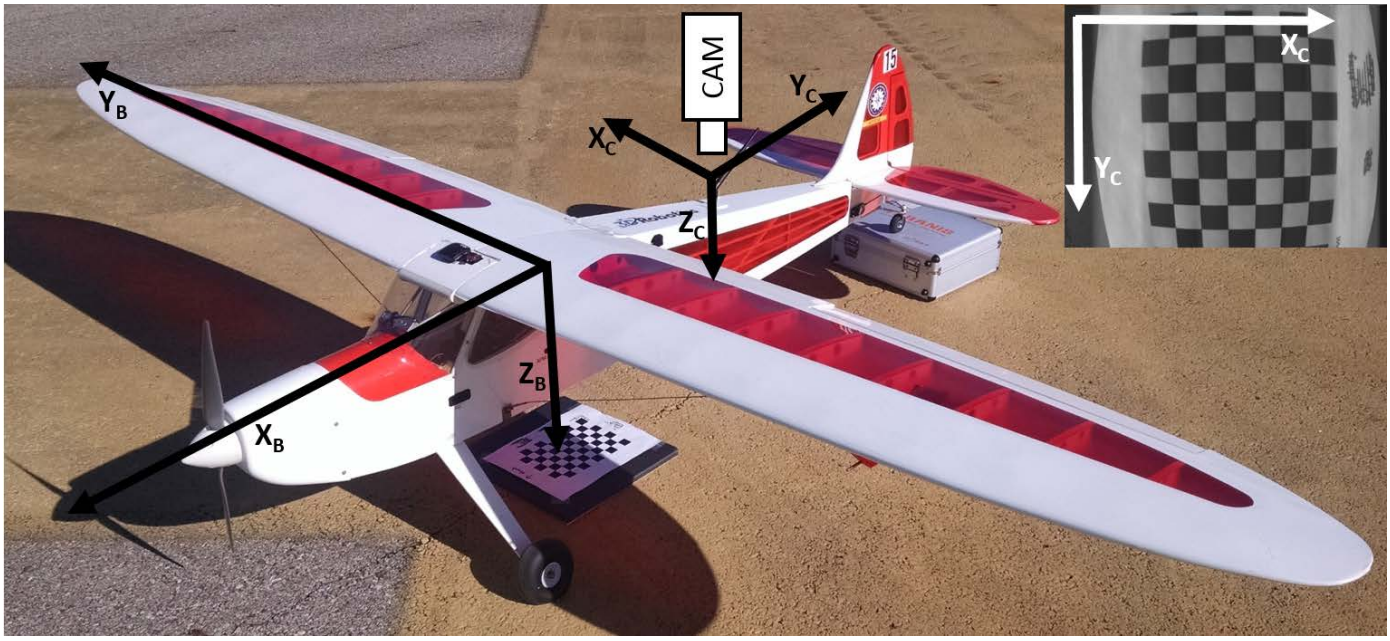



**Figure 21. Aircraft Body and Camera Frame Alignment**

A camera to body rotation was used to rotate the VO calculated velocities into the aircraft body frame. This was done by rotating the camera frame velocities to an approximate body frame and then to the corrected body frame which accounted for physical mounting errors in the alignment of the camera frame to the body frame. The error in the camera mounting was calculated by taking a picture of a checkered board placed below the aircraft after the autopilot had been calibrated to level. The checker board contained a dot that was aligned with a weight that was hanging down from the camera via fishing line to account for the true aircraft body frame z-axis ($Z_B$). Since the autopilot was leveled, the dot on the checkered board provided a best guess of the true aircraft body $Z_B$ intersection with the checkerboard. The required aircraft attitude offset to align the center of the image seen in the upper right of Figure 21 with the dot on the checkerboard was calculated. The pixel difference in the image between the center of the image and the dot combined with the intrinsic calibration of the camera provided a best guess estimate of the error in the camera

mount relative to the aircraft body frame. This calculated offset was then applied to accurately rotate the VO velocities from the uncorrected camera frame into the NEU frame. This measurement approach was deemed sufficient since the VO measurement relied on a difference of aircraft attitude between images and a small error of a degree or likely did not cause the error seen in the VO measurement.

### 4.7.4 VO Error.

Figure 22 shows a comparison between the inertial (Pixhawk) aided VO measurements and the OpenCV aided $VO_R$ measurements. The VO algorithm took the velocities seen in the camera frame and rotated the measurement into the NEU frame to be applied by the filter as a VO measurement. As was stated in section 3.2.2.4, the inertial aided VO measurement appeared to contain a large amount of noise as seen in Figure 22. When the estimated rotation between the two images was calculated using OpenCV the $VO_R$ measurement tracked much closer to the truth.

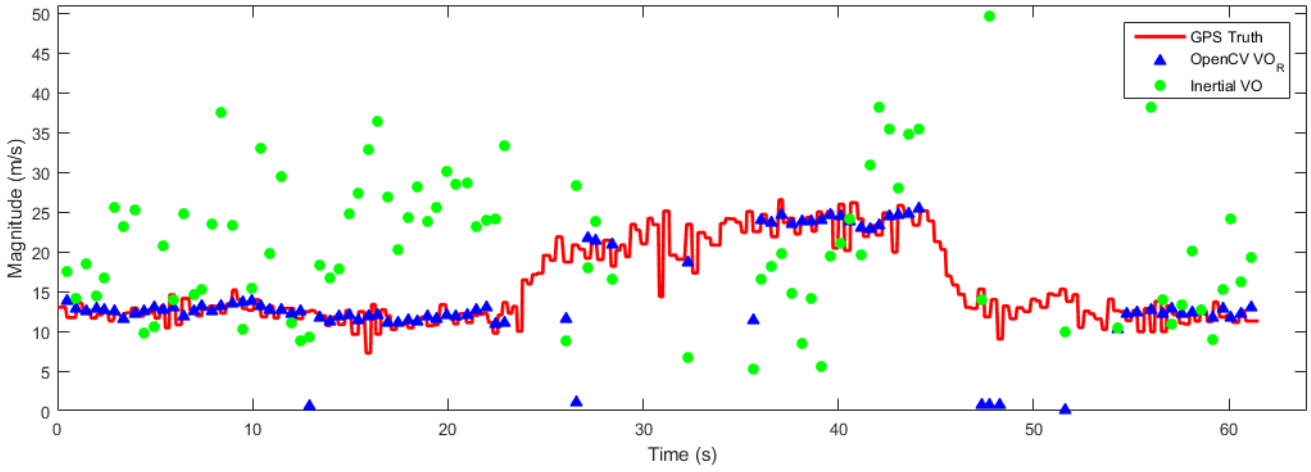



**Figure 22. Comparison Between VO and $VO_R$ Measurements with GPS Truth Velocity**

Figure 23 shows the error between the estimated measurement and the GPS truth at the times when the measurement was applied to the filter. It can be seen that the OpenCV $VO_R$ error, having an RMS of 4.14 m/s error, was better able to estimate

the GPS truth measurement. The inertial VO measurement error, having an RMS of 12.01 m/s error, produced enough noise to increase the 2DRMS errors for the VO sensor cases as seen in Table 2.

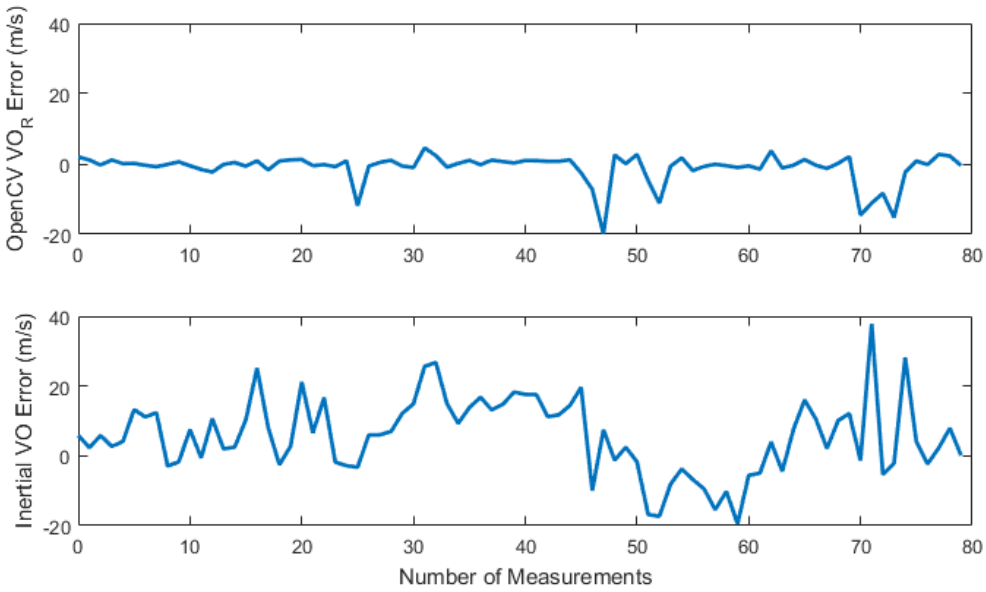



**Figure 23. Error Between VO and $VO_R$ Measurements When Compared with GPS Truth Velocity**

As an insight into the errors seen in the inertial VO measurement, Figure 24 shows the roll, pitch, and yaw collected from the Pixhawk autopilot during the initial 18 second straight flight from oneloop. The data resolution was 10Hz limited by the throughput of the Pixhawk autopilot's external communication ability. From Figure 24 it is noted that the rate of change is the greatest for the roll, varying between -8 and 8 degrees. This dynamic roll could play directly into the inertial VO estimated velocity error. Since the attitude of the aircraft was only captured at 10Hz, the sampling rate may not have been fast enough to capture the true dynamics of the aircraft attitude. Additionall, the Pixhawk generated attitude solution may have additional noise due to the interval updates from the Pixhawk internal estimation algorithm, which would case VO measurement errors to occur. Finally, any timing

errors in the measurements or Pixhawk outputs could also contribute to this error.

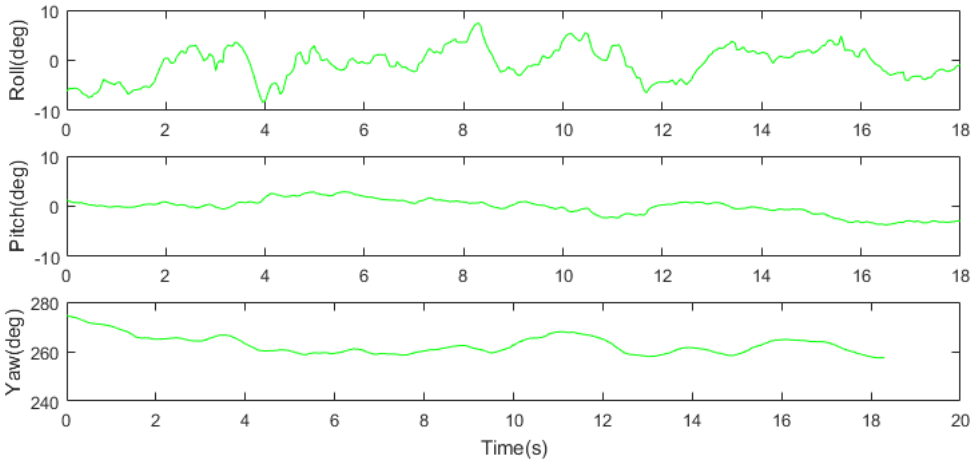



**Figure 24. 18 Second Oneloop Aircraft Roll Pitch Yaw Truth Data**

Given that the images were not taken at exactly the same time as the aircraft attitude update, the attitude of the aircraft at the time of the image was interpolated from the past and future aircraft attitudes. The interpolation was done in Scorpion after the data sets oneloop and longloop were collected. The interpolated attitude was a best guess estimate of the aircraft attitude at the time the image was taken. An analysis was done to compare the interpolated Pixhawk attitude image rotation between a past and current image with an OpenCV image rotation estimator of the same two images. Figure 25 depicts the Euler angle rotation error between two images of the estimated Pixhawk rotation between the images and an OpenCV rotation estimation between the same two images. The blue triangles show the Euler angle rotation difference of the Pixhawk estimation minus the OpenCV estimation for 73 different image compares. The images are from the first 24 seconds of the oneloop data when the aircraft was traveling from east to west in a generally straight flight pattern. Of note is the largest error lies in the roll and yaw, meaning the Pixhawk roll and yaw estimated image rotation differs the most from the OpenCV image roll estimate. When comparing the roll and yaw error to Figure 24 it can be speculated that the

roll and yaw were more dynamic then the pitch meaning the interpolated attitude may have not been able to capture the true dynamics of the roll and yaw. Since the pitch was not as dynamic, the interpolated pitch was likely sufficient resulting in a smaller error for pitch in Figure 25.

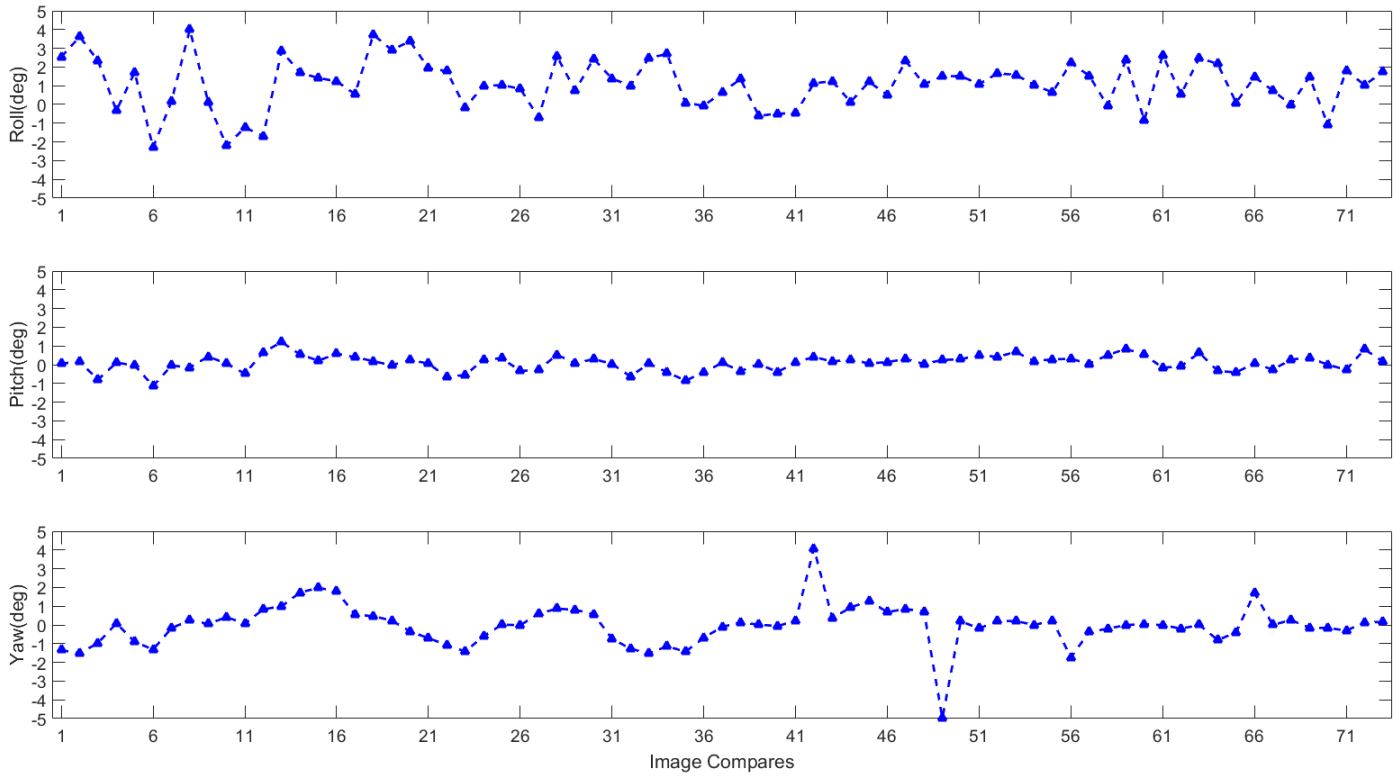



**Figure 25. Error Between Image Rotation and OpenCV Estimated Image Rotation**

An analysis using the truth velocities from the Pixhawk could have been used to reverse the problem and solve for the attitude change in the aircraft between the two images. This would have been an alternative to the OpenCV rotation estimation between the two images, but would have been much more accurate because it would have removed the high correlation between the attitude errors.

Once the inertial VO using the interpolated autopilot attitudes was deemed inaccurate the second measurement $VO_R$ was created where the rotation between the

two images being compared was estimated using an OpenCV algorithum. The estimated camera rotation matrix was then used in the $VO_R$ algorithm. Using OpenCV to estimate the camera rotation produced a more accurate ground speed update than the interpolated attitudes. In contrast, a seperate study in [40] using a tactical grade IMU with an update rate of 100hz, use of the OpenCV estimated rotation made the results significantly *worse* when compared to using the IMU to estimate the camera rotation.

## 4.8 Chapter Conclusion

The ability of the SUAS to determine its position was analyzed using an EKF, an aircraft motion model, VO, and range measurements. Twenty four scenarios covering two data sets and a variation of a single range measurement, two range measurement, and three versions of VO were tested. It was found that adding a range measurement to a known location using the simplified motion model allowed the EKF to incorporate absolute positional data into its solution which reduced the 2DRMS error. The addition of a second ranging measurement reduced the 2DRMS error even further. In the case of the motion model and SimVO using the oneloop data, adding a range measurement reduced the 2DRMS error from 359 m to 75 m. Adding a second range reduce the 2DRMS error from 75 m to 11 m. The ranges provided absolute position data which greatly reduced the 2DRMS errors. Through this research using the motion model, it was also found that it is significantly better to have two range measurements than a single range measurement. The accuracy of the SimVO measurement was also proven to reduce the 2DRMS error as the measurement became more accurate. However, the range measurements were found to be the dominating factor in greatly reducing the 2DRMS error.

# V. Chapter 5 - Summary and Conclusion

The results and final conclusion are discussed in this chapter of the thesis. A broader analysis of the results along with future work are presented.

## 5.1 Results Discussion

Overall the results show that the driving factor in an accurate position estimation was the use of the range measurements. The VO measurements, as applied in this research through velocity updates, only helped the motion model better estimate the ground speed; its effect on reducing the position error was not as great as an absolute range update to the position. When using a single range combined with an accurate SimVO measurement (standard deviation of 0.1 m/s) the 2DRMS error was 46 m. The 46 m 2DRMS error hints at the idea that even with a very accurate ground speed update the single range cannot offer the observability into the course angle($\chi$) that two ranges provide. It was also noted that the accuracy of the range measurement required is highly dependent on the number of ranges being used. In many cases it would be better to have less accuracy in two range measurements then a more accurate single range. Overall this research shows that a range measurement can provide an absolute position update that when combined with an EKF can contain the drift seen in many dead reckoning and VO navigation solutions.

An analysis of different motion models was not conducted so a true evaluation of the Beard and McLain aircraft motion model was limited. Having a motion model that incorporated the velocities in the body from the Pixhawk data may have produced better results. Since only one motion model was used, the Beard and McLain aircraft motion model, limits the definitive answer of the effect of the incorporated measurements. Other motion models may have, using dead reckoning, tracked better

with the truth, therefore allowing the individual effect of each measurement to be better characterized.

## 5.2 Future Research

As the research discussed in the thesis comes to a close, opportunities to advance this research further are presented.

### 5.2.1 Flight Patterns.

In future work an analysis of how the higher loitering range providing aircraft flight pattern can be optimized to present the best possible geometric position solution for the lower flying SUAS. Having the higher flying aircraft loiter in the air may not be the most optimized range measurement for the SUAS EKF.

### 5.2.2 Quality IMU.

This research relied on aircraft attitude and angular rates for the motion model and VO measurement. The attitude and angular rates were used in the motion model to present a dead reckoning solution when measurements were not being incorporated. The attitude was used in the inertially aided VO to predict the rotation in the images due to aircraft body frame movement. As was discussed in Chapter IV, the interially aided VO measurement may not have been as accurate as expected due to the error in the aircraft attitude at the time the image was taken. A higher quality IMU would be suggested as the Pixhawk update rate appears to be to slow for the highly dynamic roll. Having an IMU capable of producing accurate measurements faster then 10Hz may have reduced the errors seen in the inertially aided VO measurement. A more accurate IMU may have also allowed the motion model to be more accurate.

### 5.2.3 Real Range Test.

If the original planned test involving two RC aircraft and the ENSCO ranging radios could be retested, it would give insight into the performance of the range measurements using non-simulated data. An analysis of a real world ranging radio performance is necessary in order to characterize the expected error in a ranging measurement between two flying aircraft. Having a third RC aircraft providing range information would also give more insight into non-simulated data results with two range measurements.

### 5.2.4 PnP Measurement.

During initial discussions of the research one of the measurements available to the EKF would have been a PnP position. Due to time limitations the additon of a PnP measurement is left for future research. In [34] using a downward facing camera and the Pixhawk autopilot, a 2DRMS of 16.44m resulted from the PnP algorithm. Incorporating a PnP measurement could have been a driving factor in a more accurate solution during the single range cases. How the PnP solution would interact with the 2DRMS error of the more accurate two range solutions is left for future study.

### 5.2.5 Filter Estimation.

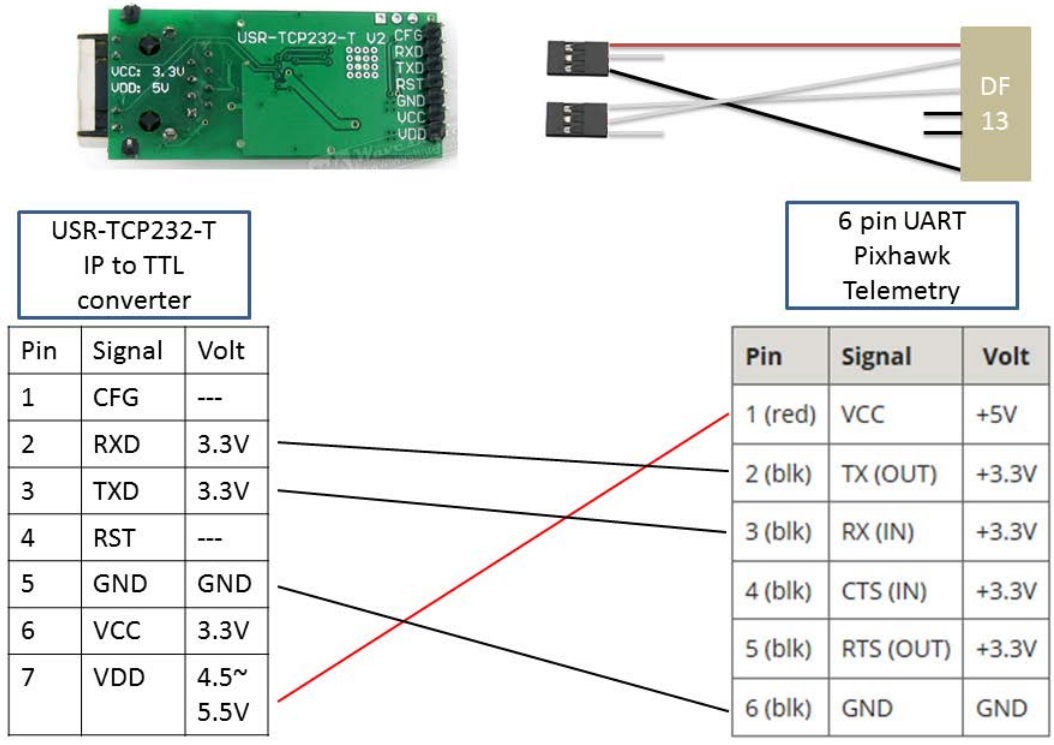
As was stated in Chapter II a delayed state KF was not used in this research due to the limitations of the Scorpion framework at the time. Had a delayed state KF been used over an EKF, then the VO measurements could have been incorporated as a velocity update versus just an input into the ground speed. Due to inaccurate inertially aided velocity estimates the heading VO update was not incorporated. Using a delayed state KF may have allowed the filter to compare the filters estimated position changes between the current state and the delayed state with the measured

position changes from the VO measurement. Implementation using a delay state KF is left as a future research topic.

# Appendix A.

## A UART Serial Ethernet to Pixhawk

UART Serial Ethernet connections to the Pixhawk can be seen in Figure 26. Standard RC servo three wire female connectors were used to connect to the UART.




| Pin | Signal | Volt |
|---|---|---|
| 1 | CFG | --- |
| 2 | RXD | 3.3V |
| 3 | TXD | 3.3V |
| 4 | RST | --- |
| 5 | GND | GND |
| 6 | VCC | 3.3V |
| 7 | VDD | 4.5~ 5.5V |

| Pin | Signal | Volt |
|---|---|---|
| 1 (red) | VCC | +5V |
| 2 (blk) | TX (OUT) | +3.3V |
| 3 (blk) | RX (IN) | +3.3V |
| 4 (blk) | CTS (IN) | +3.3V |
| 5 (blk) | RTS (OUT) | +3.3V |
| 6 (blk) | GND | GND |

**Figure 26. UART Serial Ethernet connections for Pixhawk telemetry port**

# Appendix B.

## B Telemaster and SIG Rascal 110 Ethernet Connections

The Telemaster and two SIG Rascal Ethernet configurations can be seen in Figures 27 to 29. The SIG Rascal 110 configuration one was used when the the Telemaster was also used. Configuration two of the SIG Rascal 110 was when the SIG Rascal 110 was the sole platform for data collection.

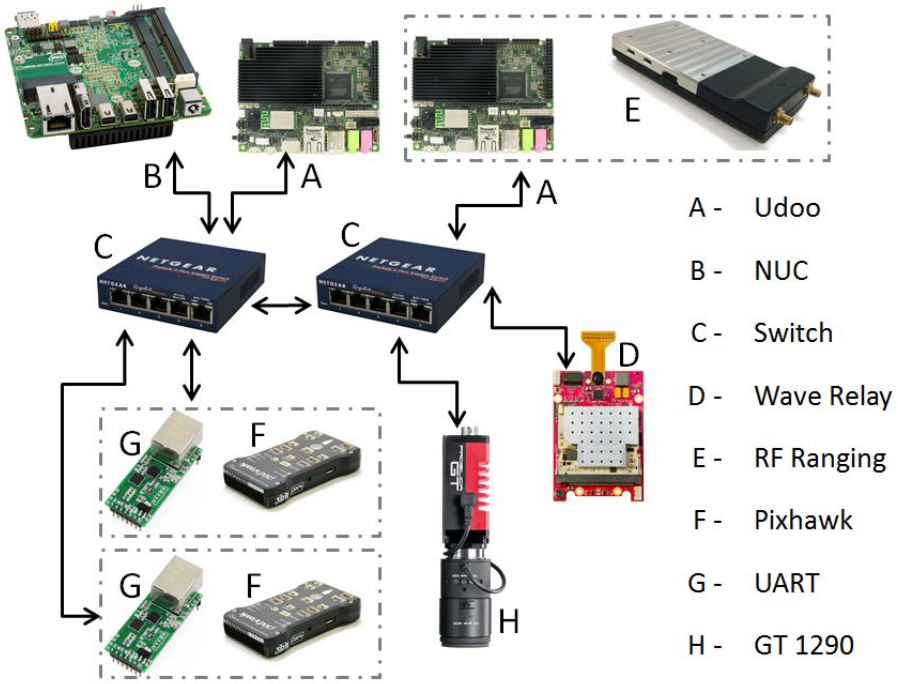



**Figure 27. Ethernet connections on the Telemaster**

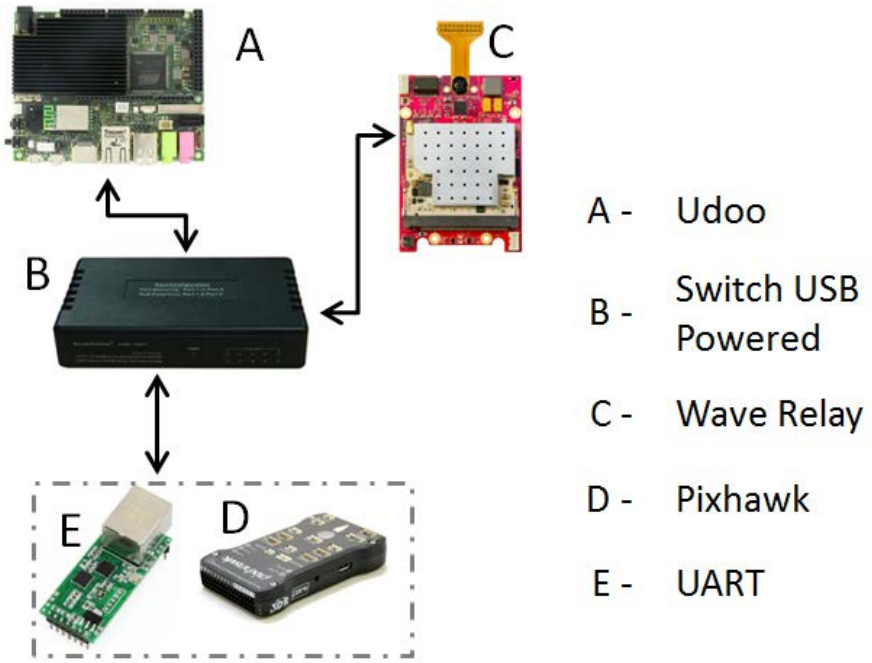



**Figure 28. SIG Rascal 110 Configuration 1**

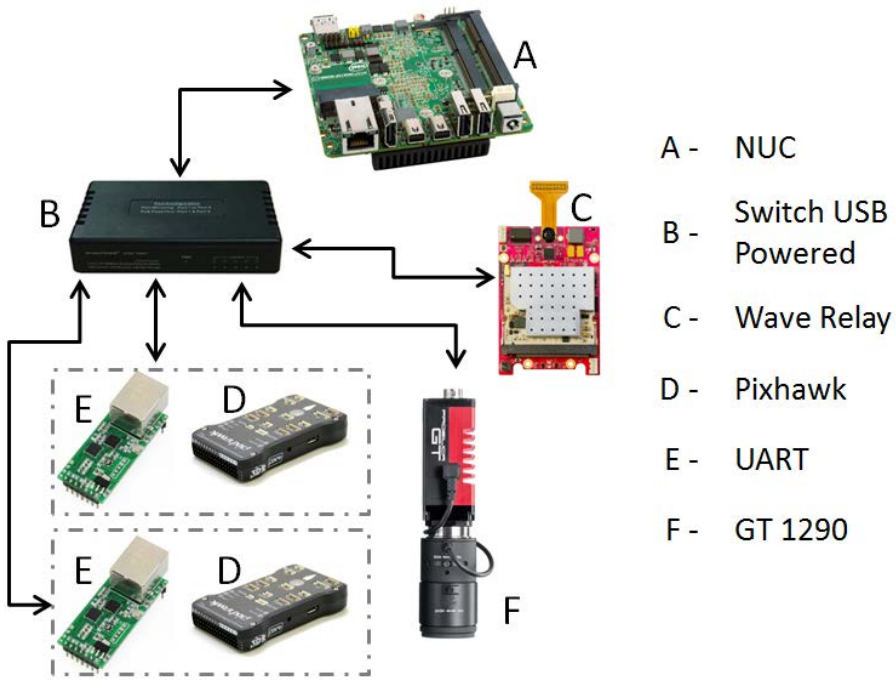



**Figure 29. SIG Rascal 110 Configuration 2**

# Appendix C.

## C Graphical Results For Each Simulation

This section contains a graphical illustration of all of the motion model states for all of the simulations not already displayed in chapter 4. They are ordered in the same sequence as Table 2 for oneloop and Table 4 for longloop.

### C.1 Oneloop Results.

Oneloop simulation results can be seen below.

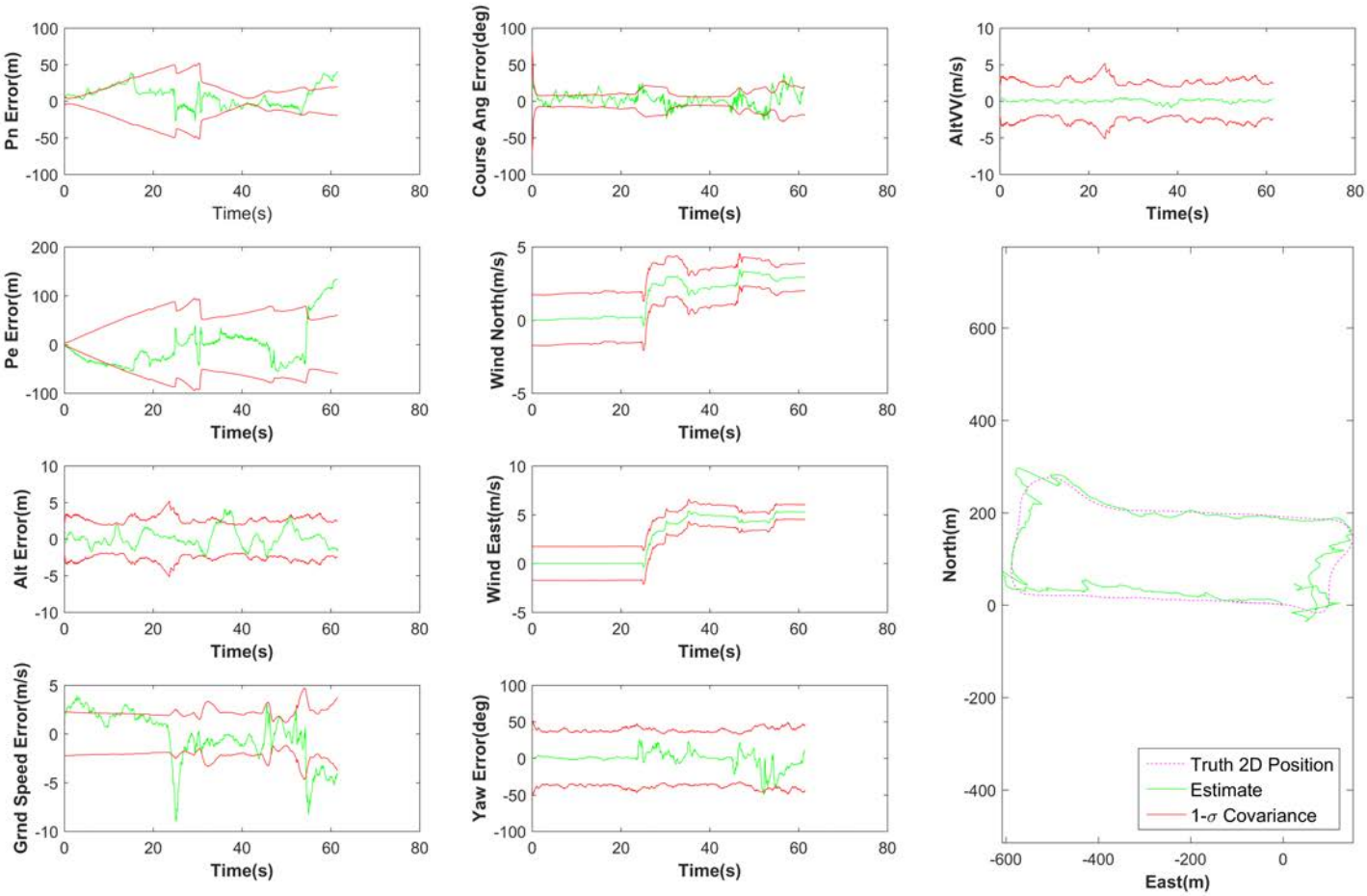



**Figure 30. Oneloop With A Single Range (oneloopmmR)**

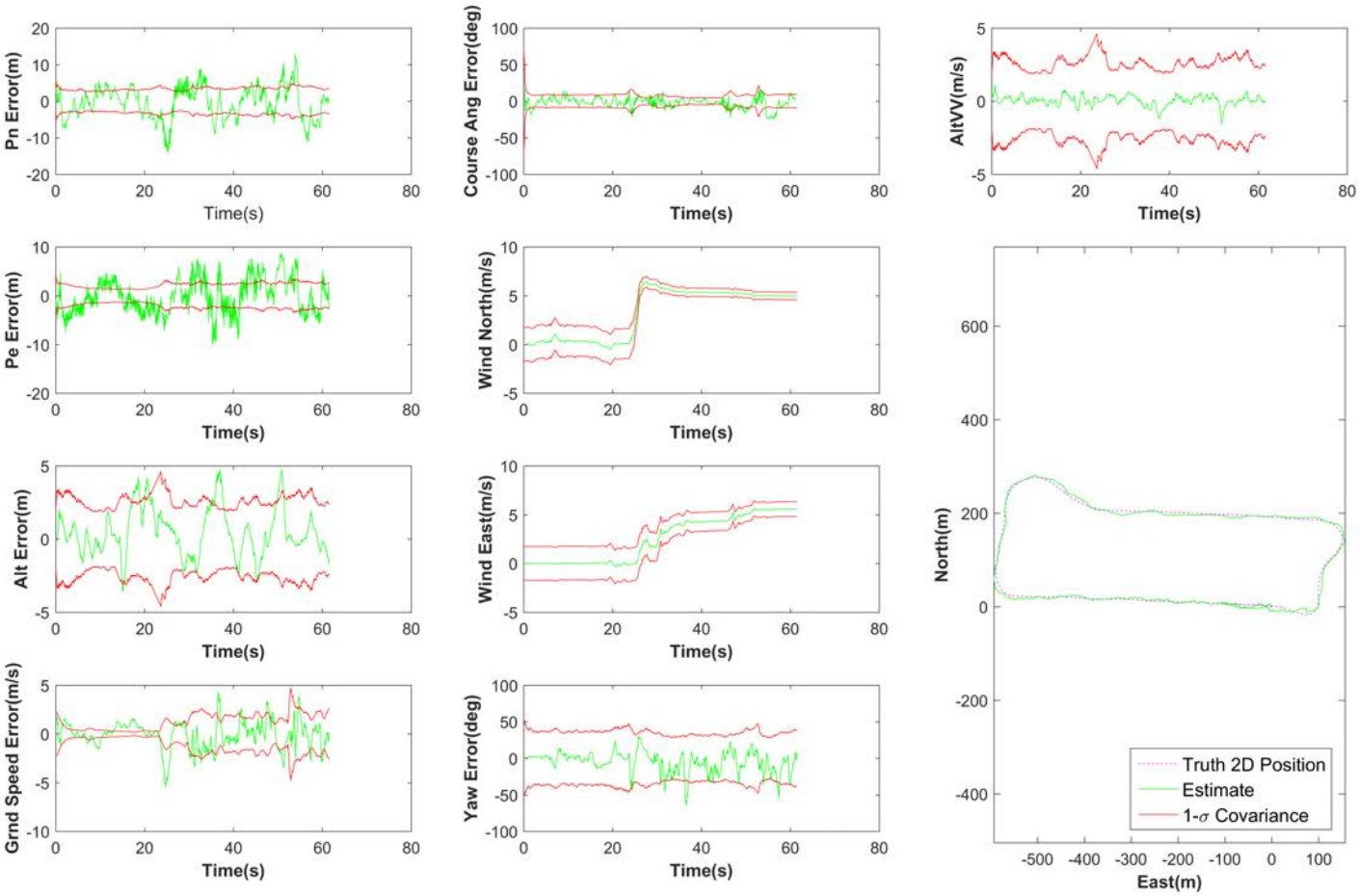



**Figure 31. Oneloop With Two Ranges (oneloopmmRR)**

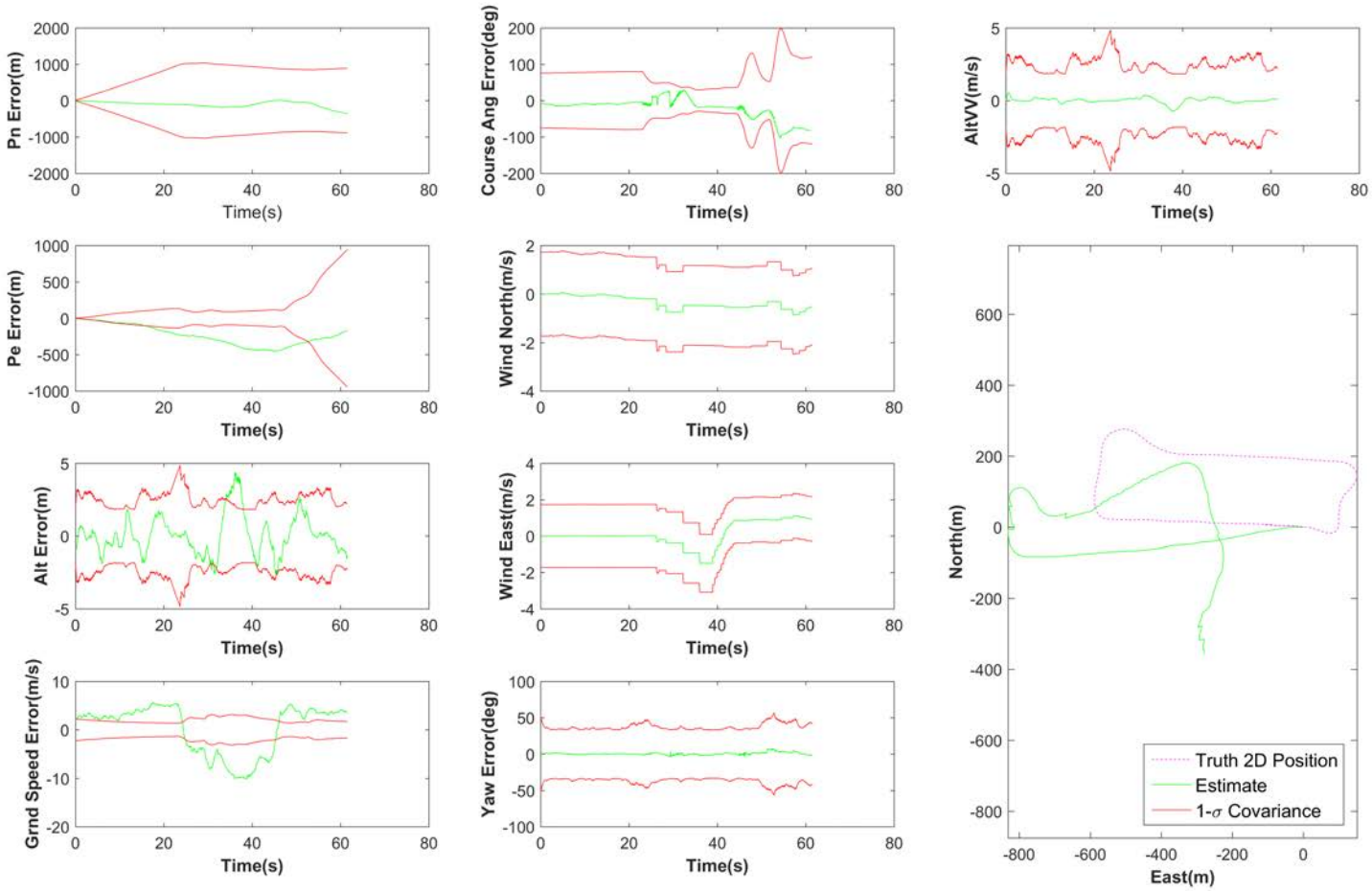



**Figure 32. Oneloop VO Only (oneloopmmVO)**

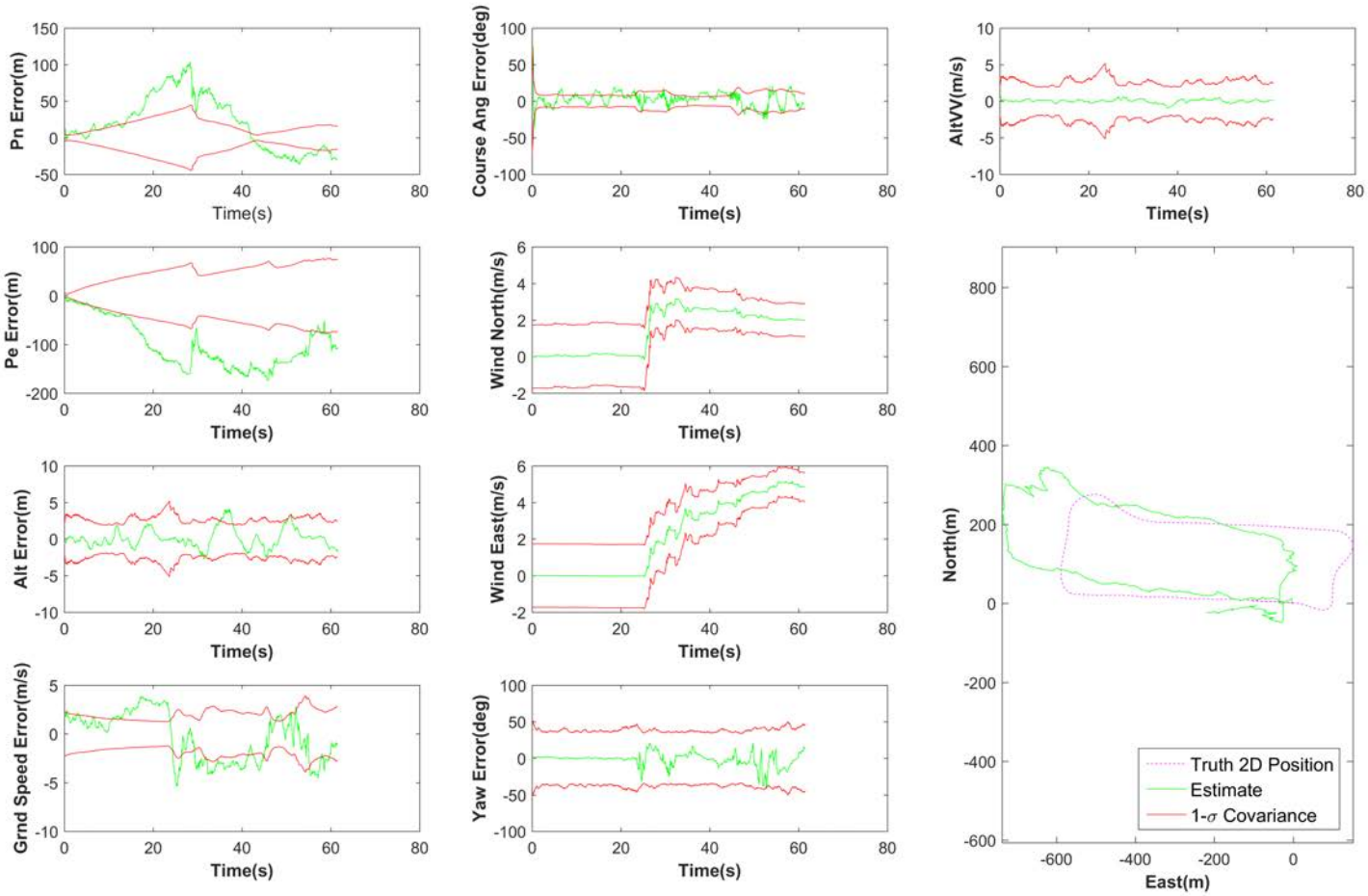



**Figure 33. Oneloop With VO and One Range (oneloopmmVOR)**

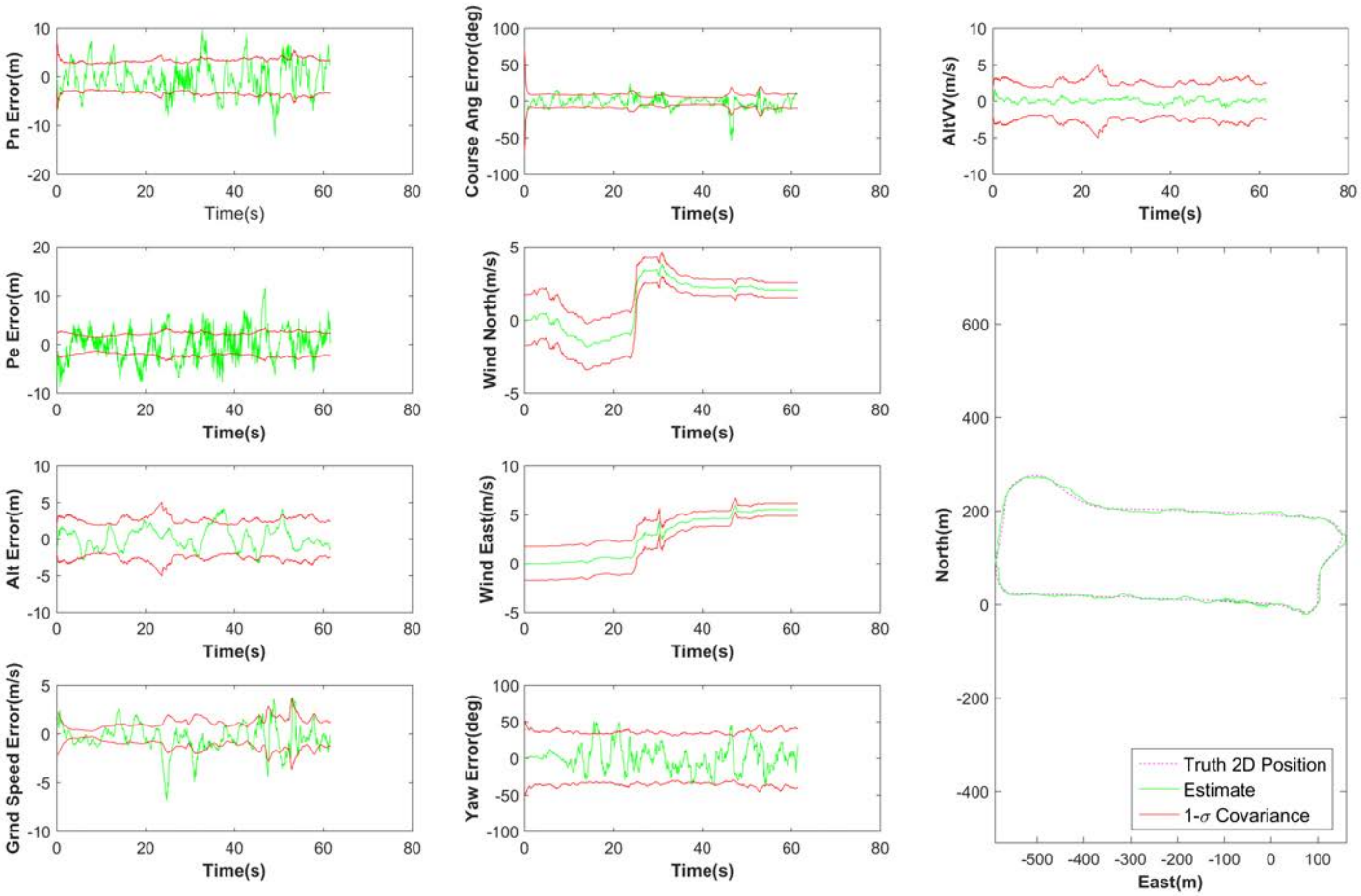



**Figure 34. Oneloop With VO and Two Ranges (oneloopmmVORR)**

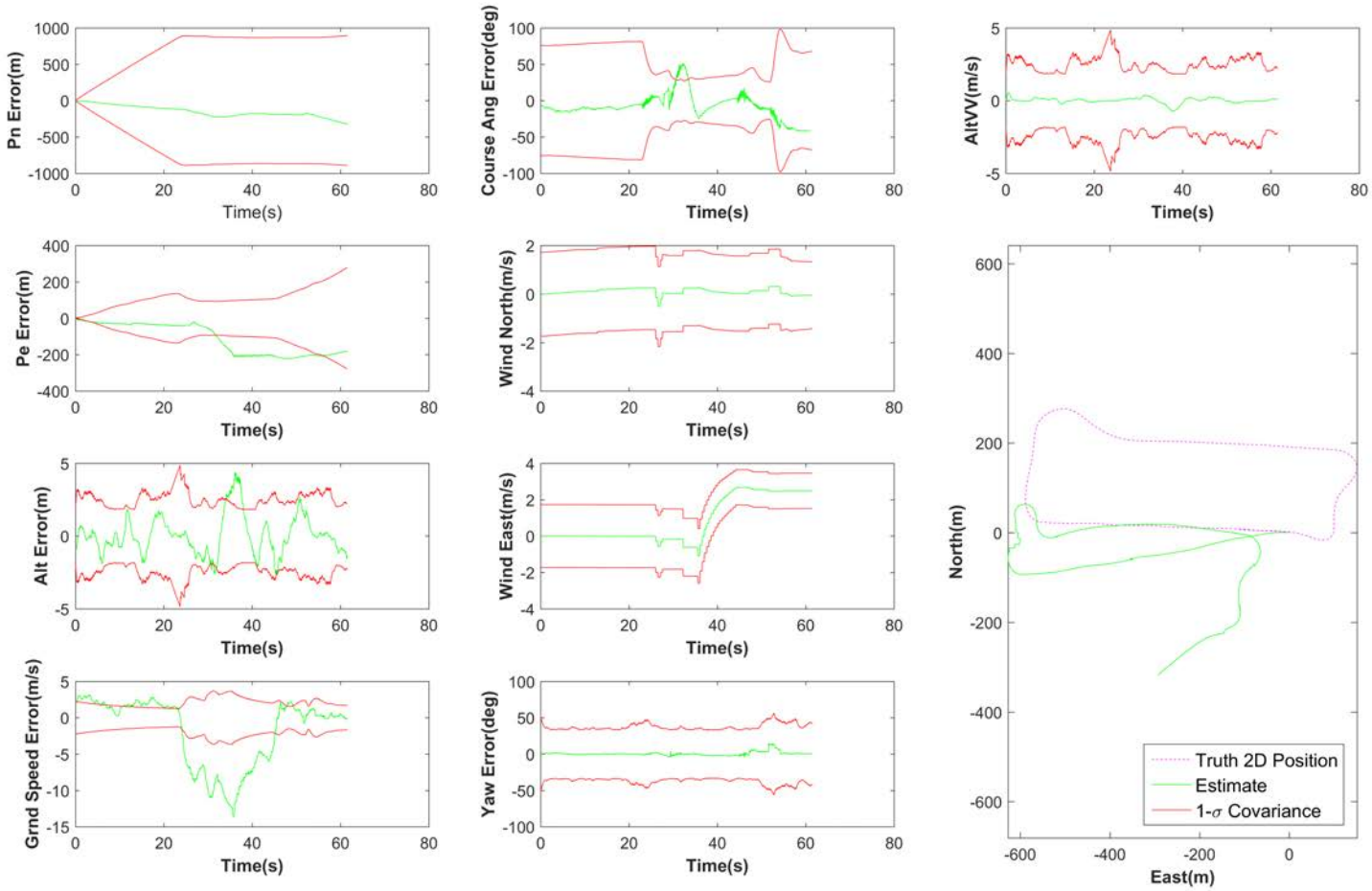



**Figure 35. Oneloop $VO_R$ Only (oneloopmm$VO_R$)**

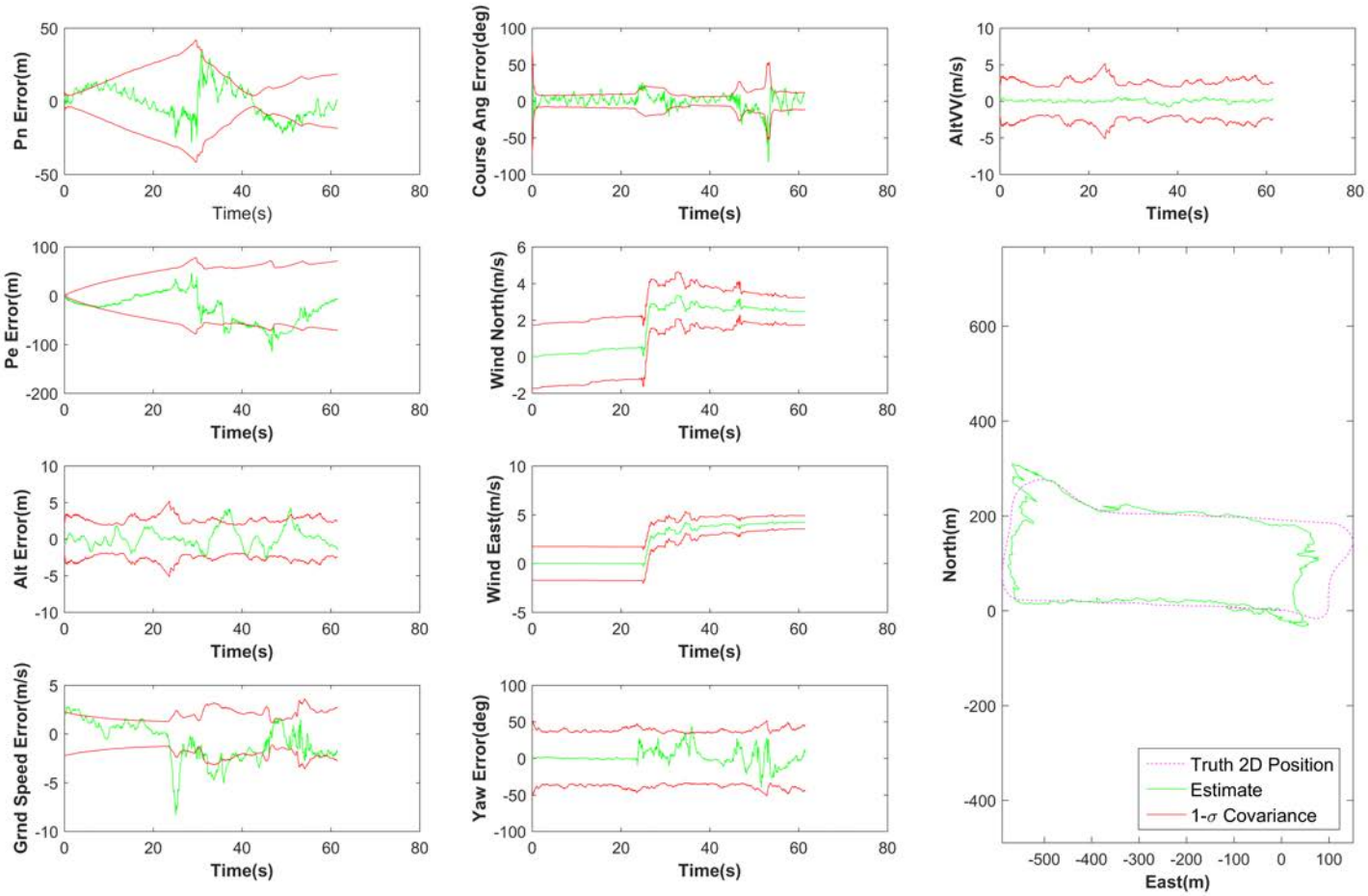



**Figure 36. Oneloop With $VO_R$ and One Range (oneloopmm$VO_R$R)**

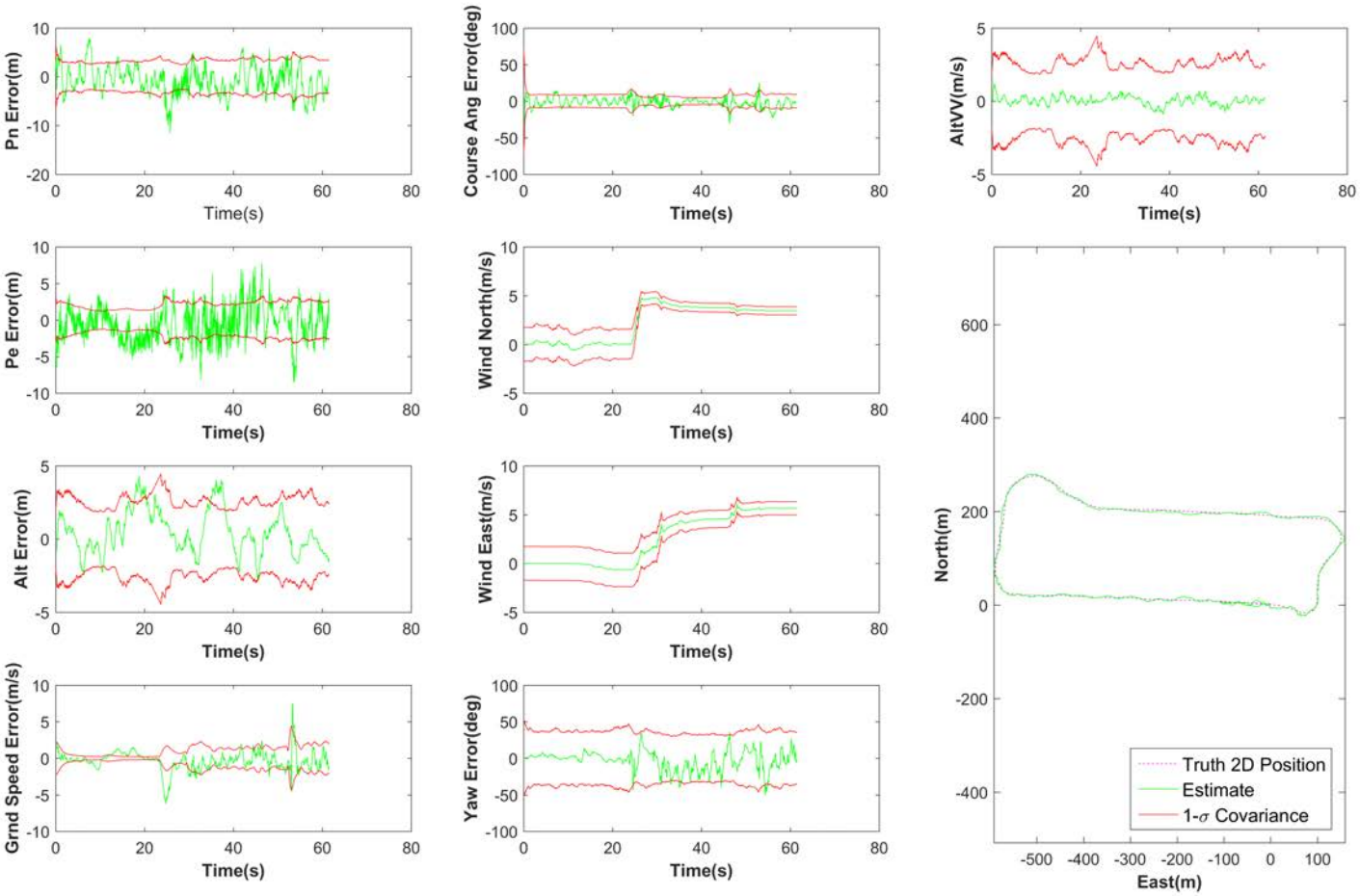



**Figure 37. Oneloop With $VO_R$ and Two Ranges (oneloopmm$VO_R$RR)**

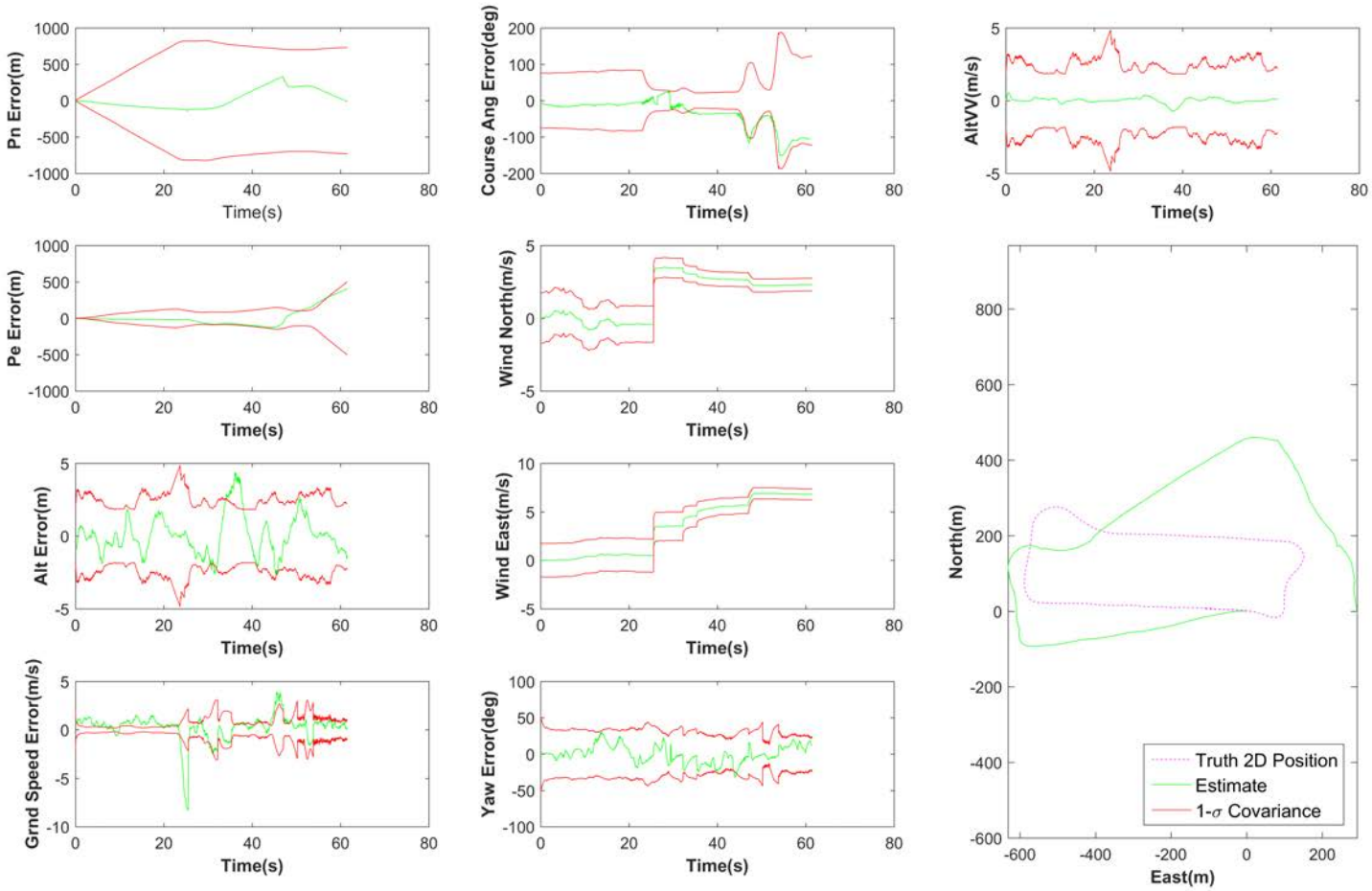



**Figure 38. Oneloop SimVO Only (oneloopmmSimVO)**

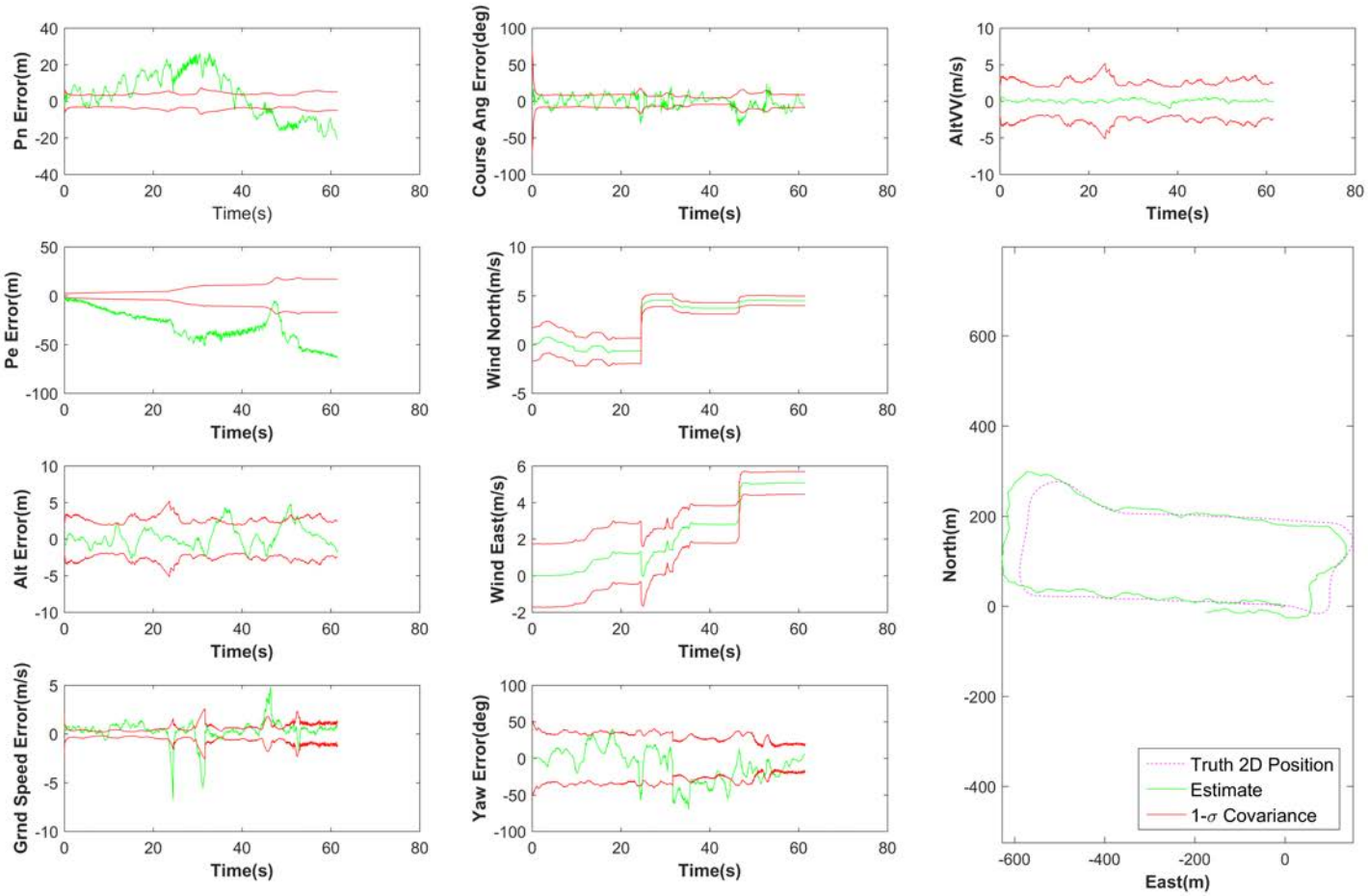



**Figure 39. Oneloop With SimVO and One Range (oneloopmmSimVOR)**

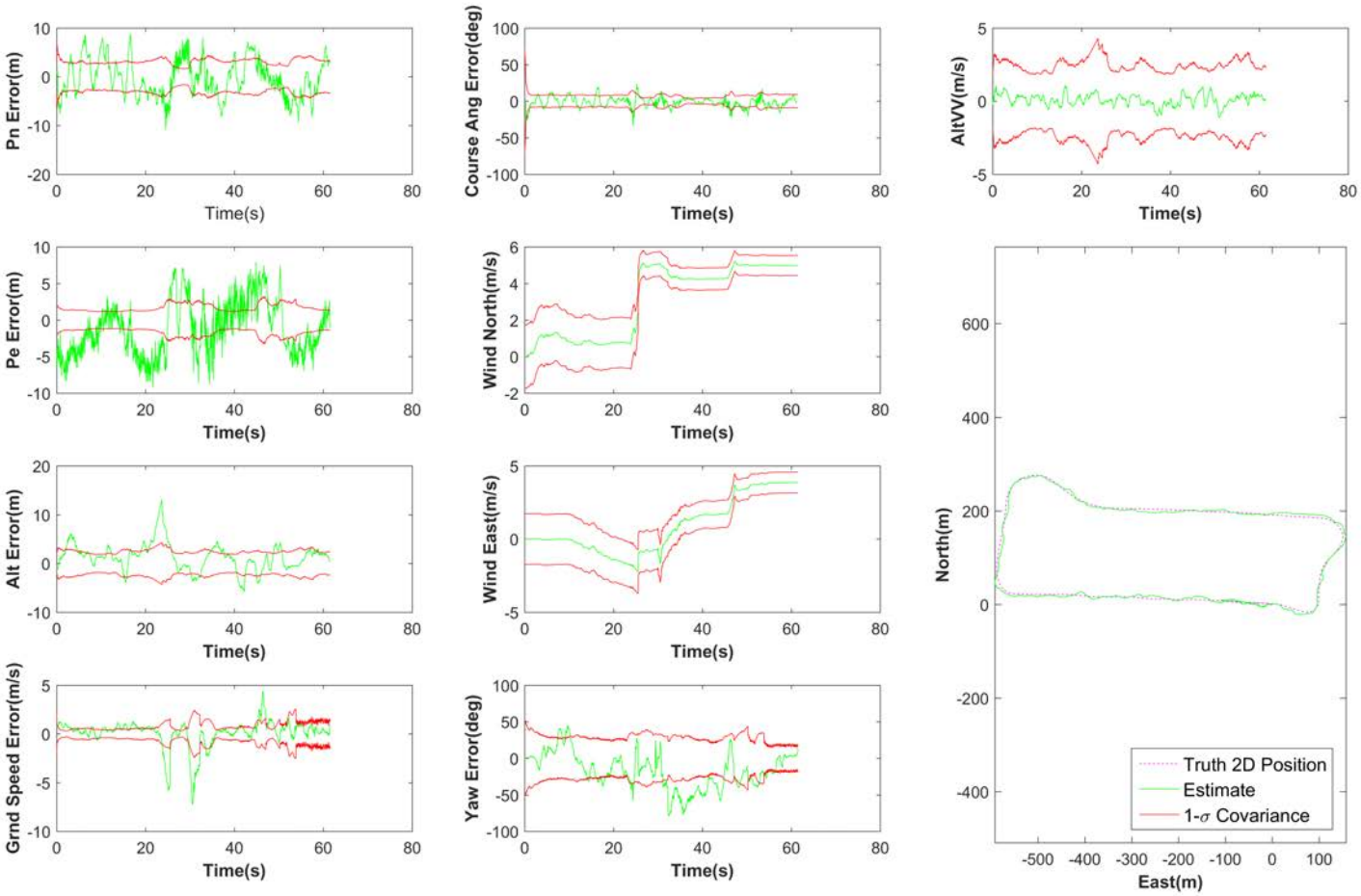



**Figure 40. Oneloop With SimVO and Two Ranges (oneloopmmSimVoRR)**

## C.2 Longloop Results.

This section contain all of the simulation results for longloop and the varying sensor combinations.

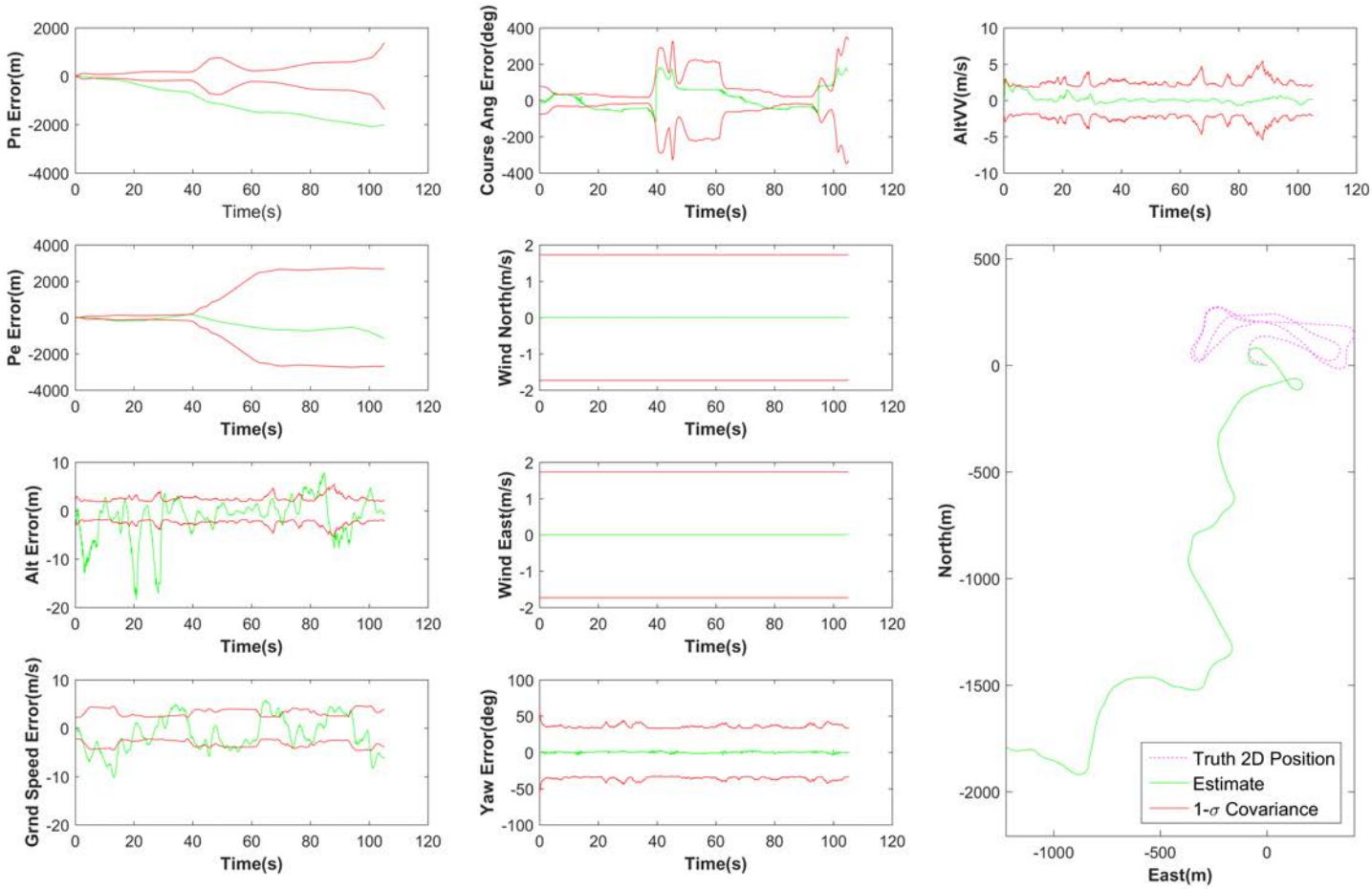



**Figure 41. Longloop With Motion Model Only (longloopmm)**

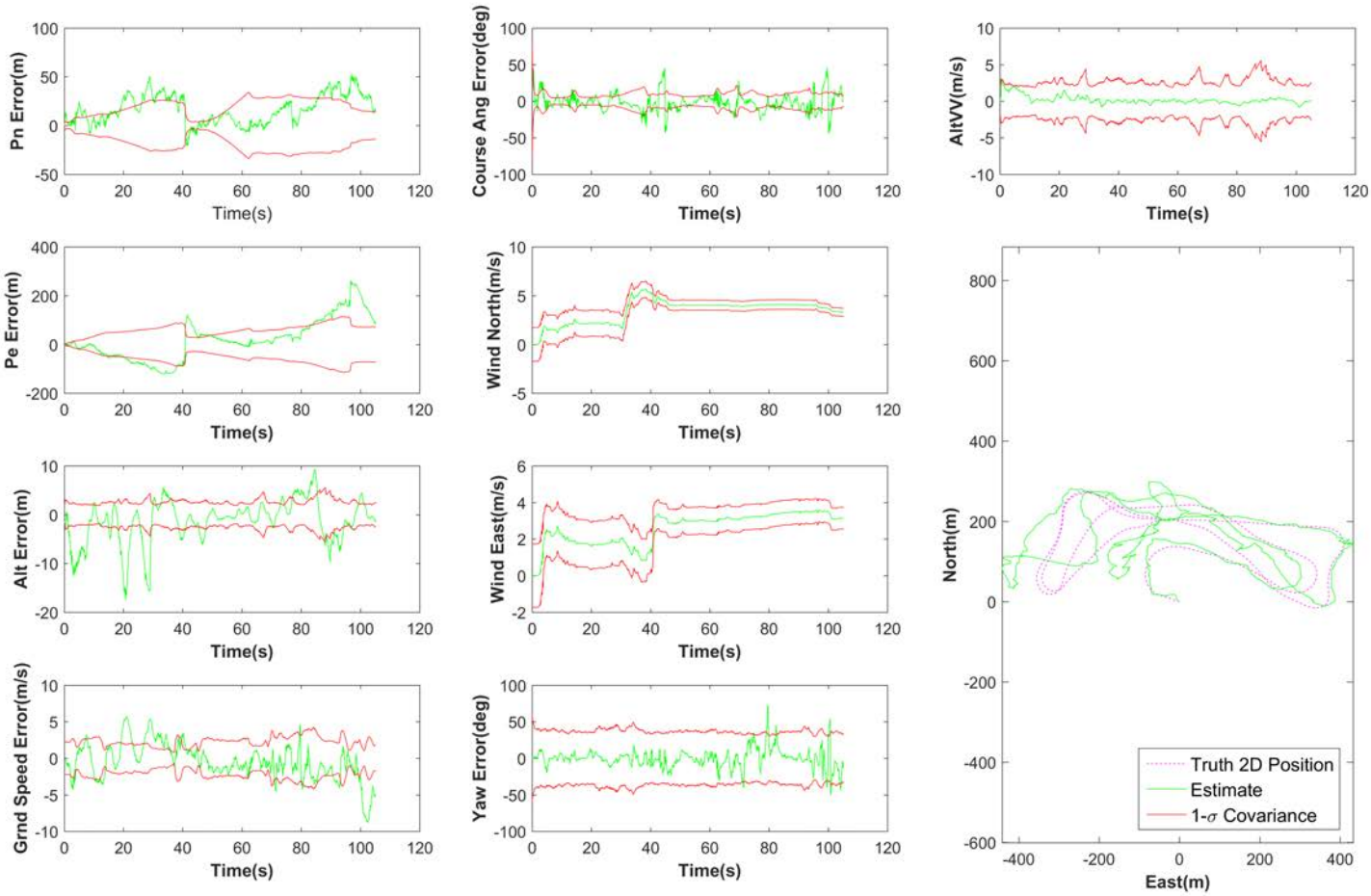



**Figure 42. Longloop With One Range (longloopmmR)**

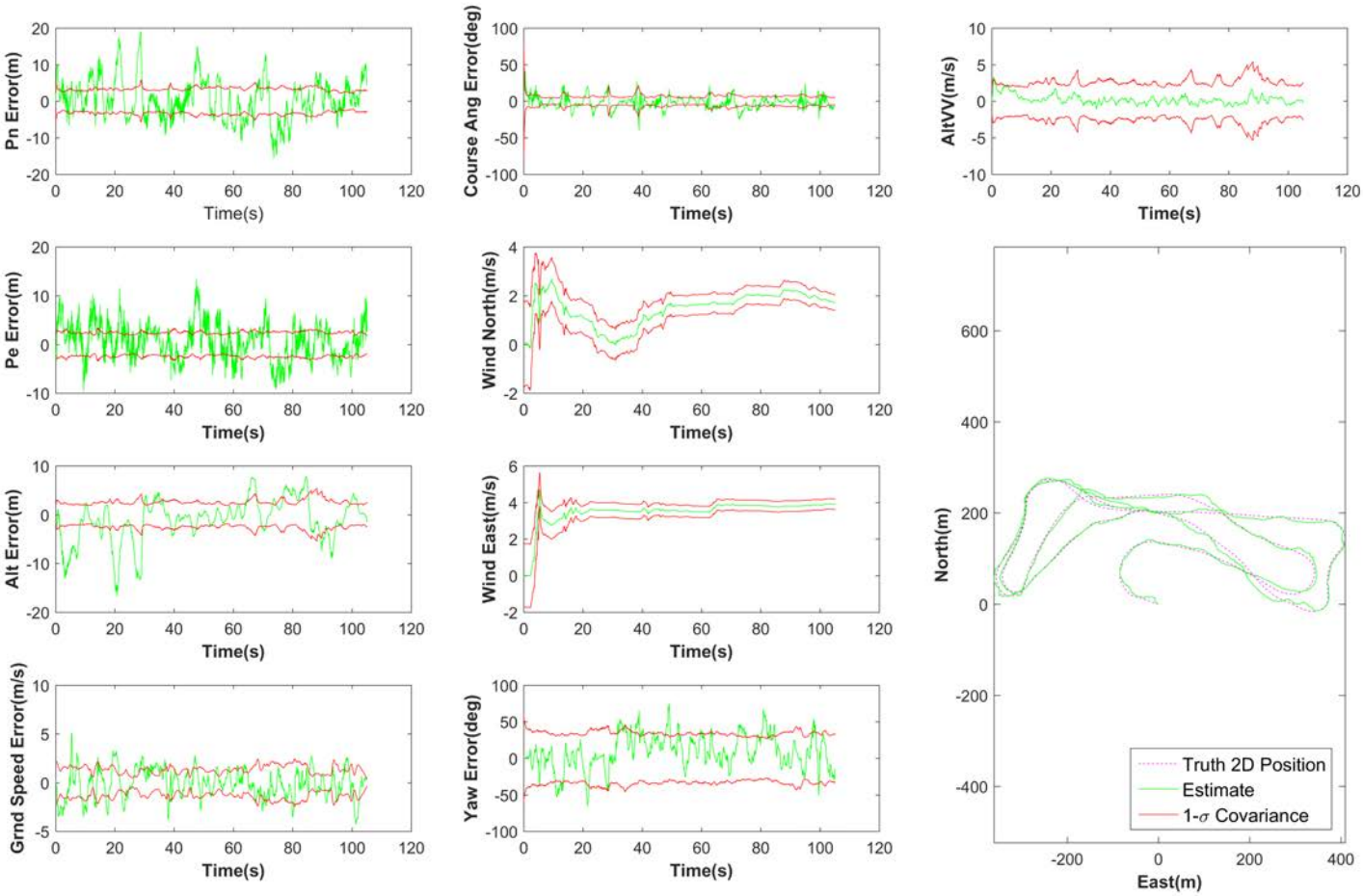



**Figure 43. Longloop With Two Ranges (longloopmmRR)**

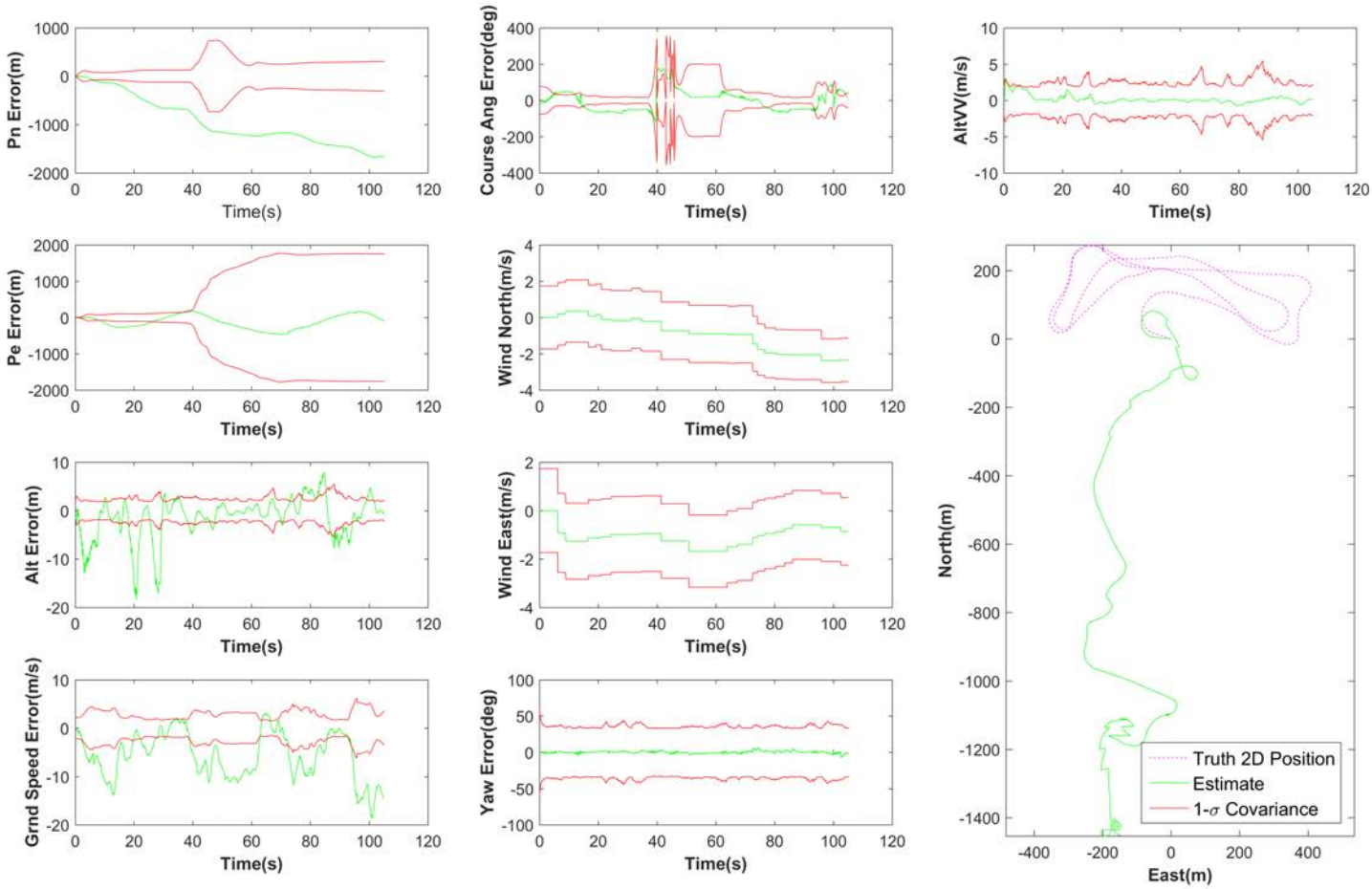



**Figure 44. Longloop VO Only (longloopmmVO)**

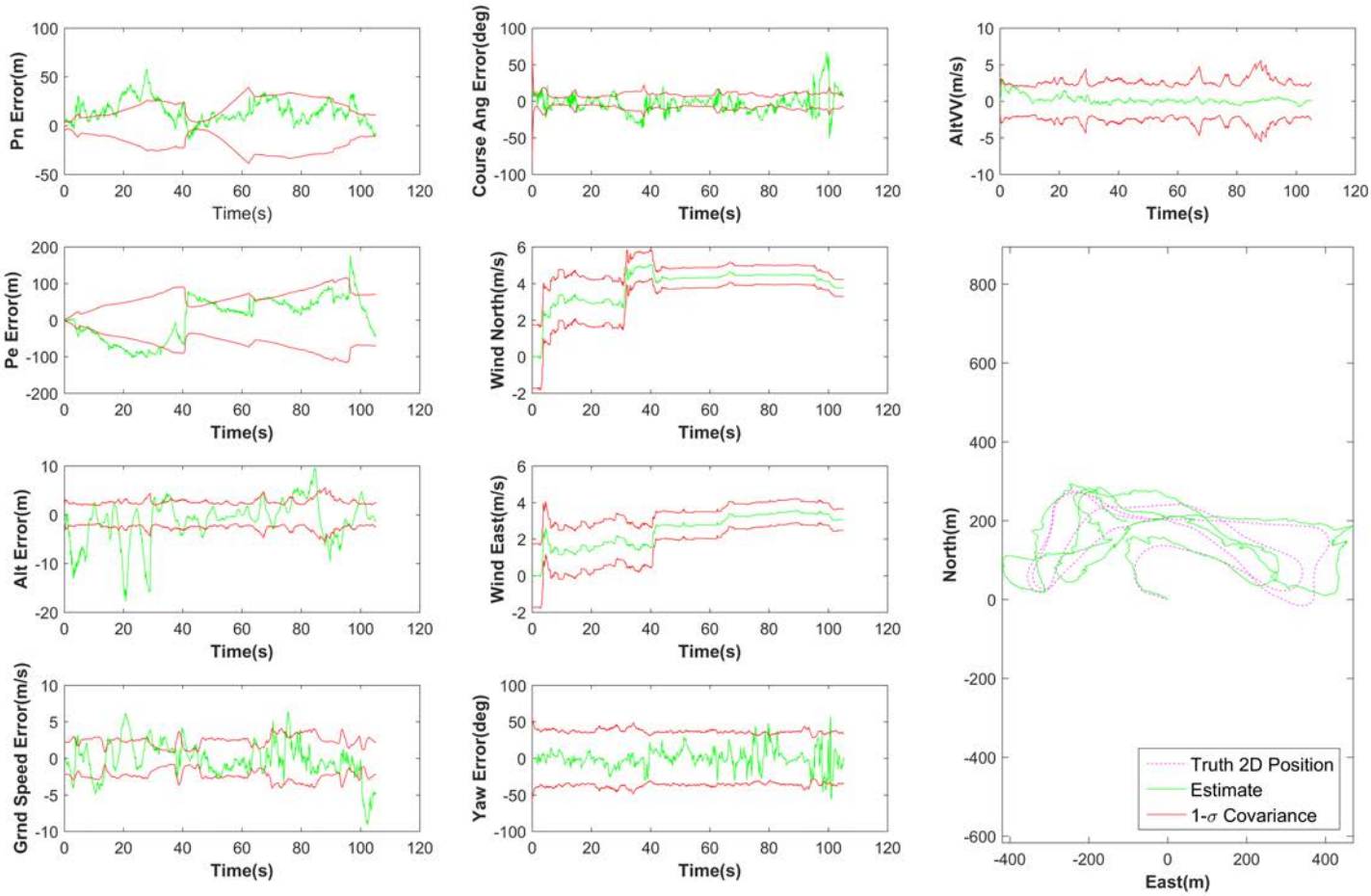



**Figure 45. Longloop With VO and One Range (longloopmmVOR)**

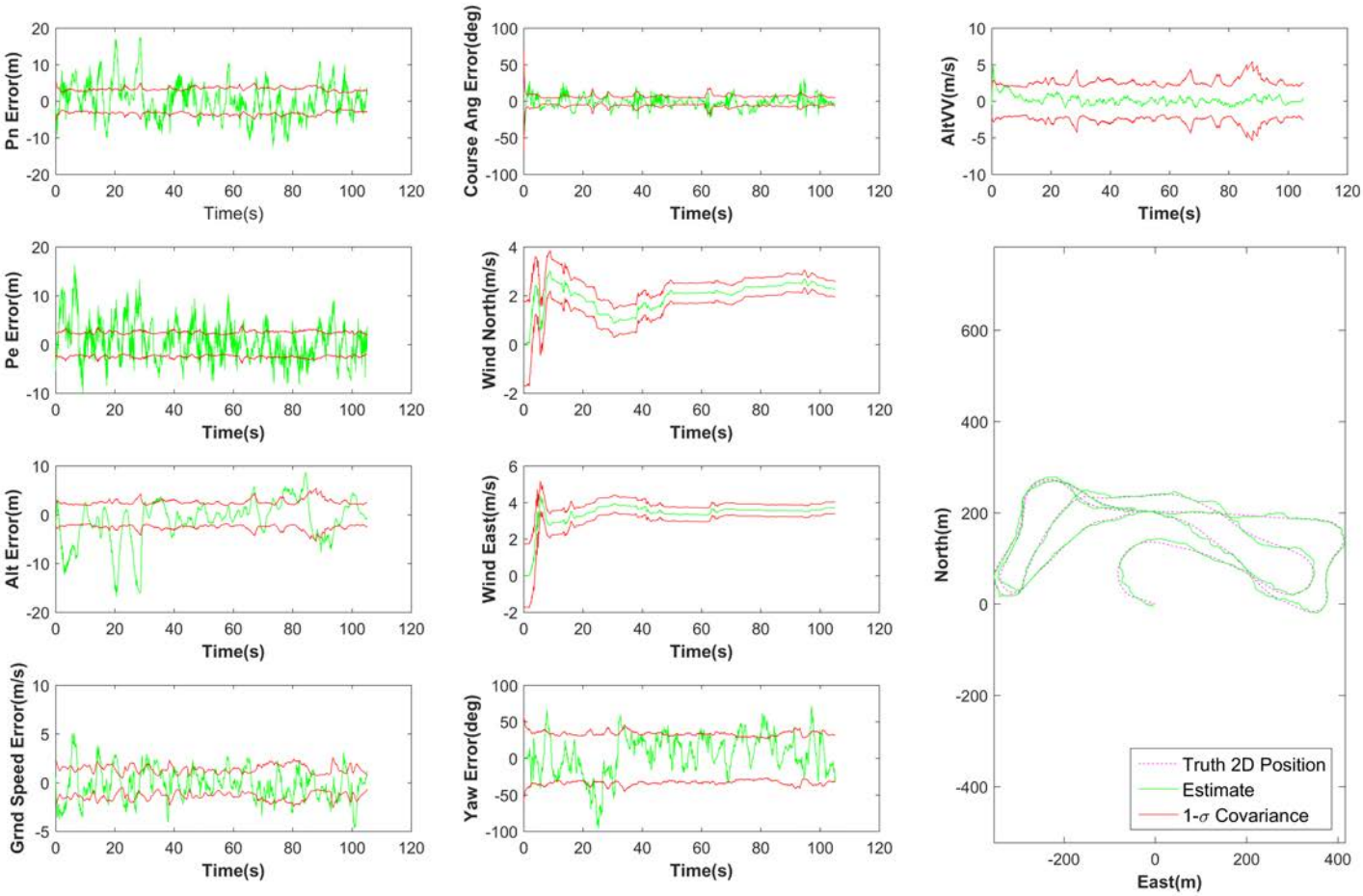



**Figure 46. Longloop With VO and Two Ranges (longloopmmVORR)**

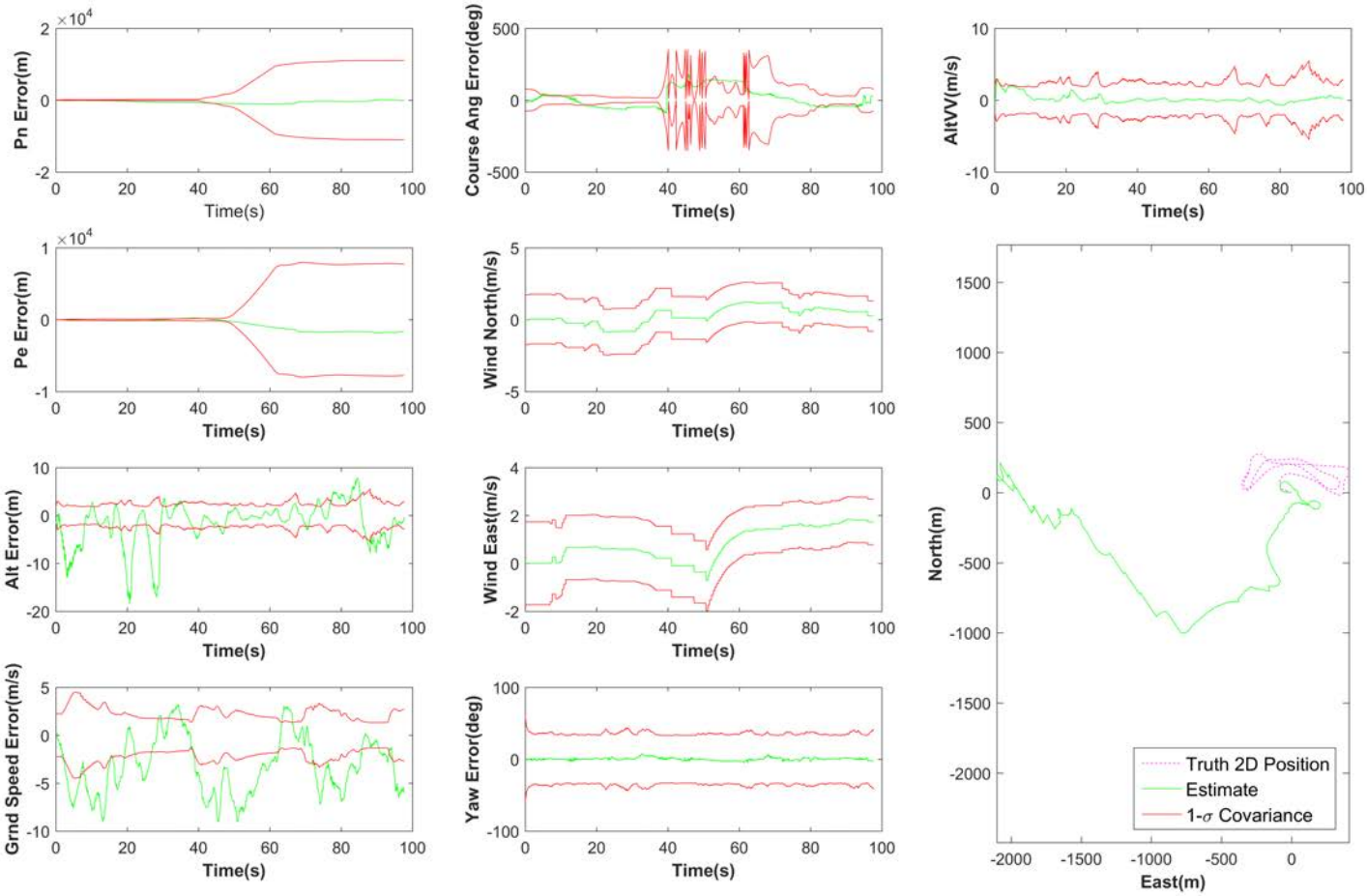



**Figure 47. Longloop $VO_R$ Only (longloopmm$VO_R$)**

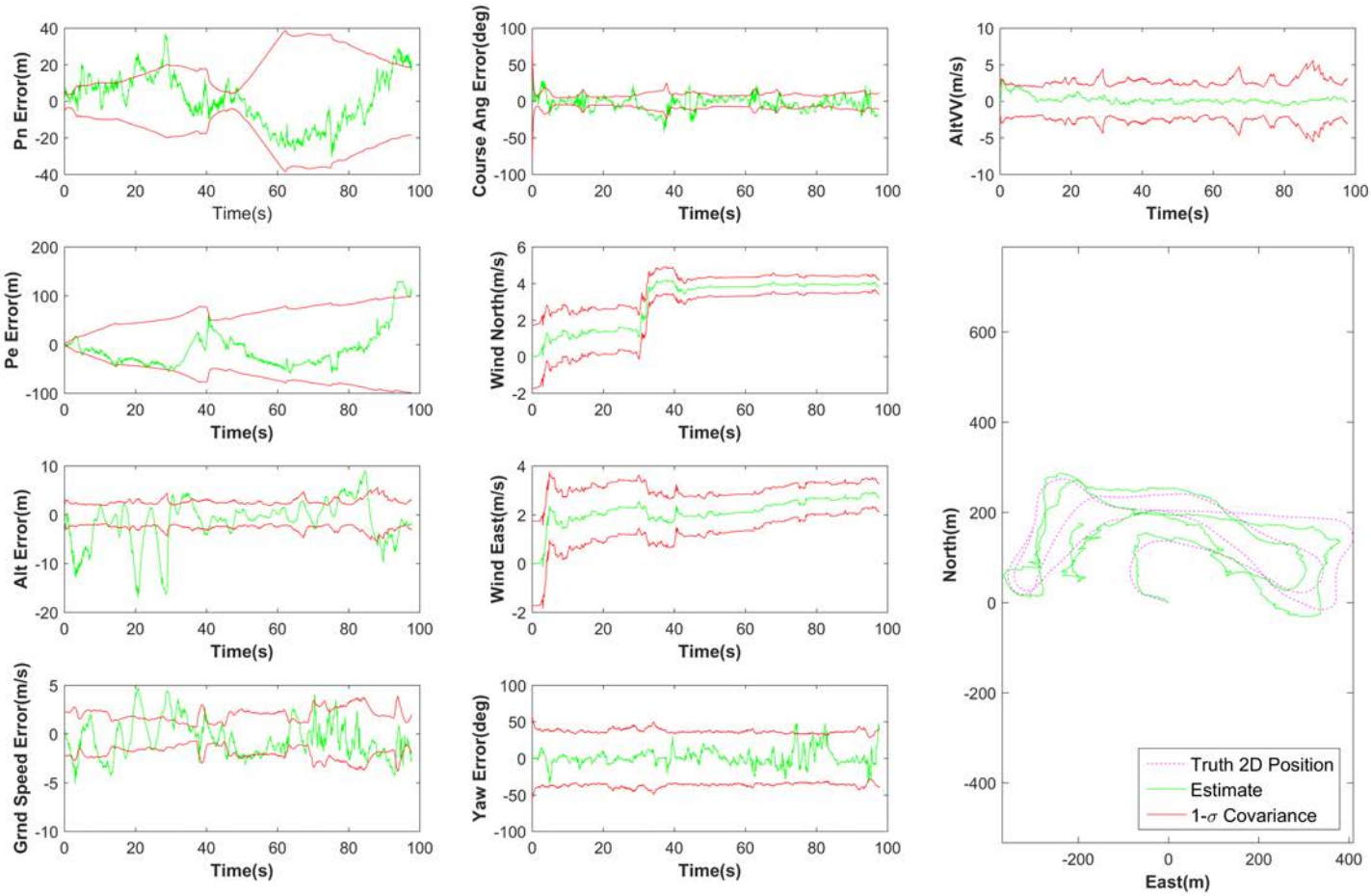



**Figure 48. Longloop With $VO_R$ and One Range (longloopmm$VO_R$R)**

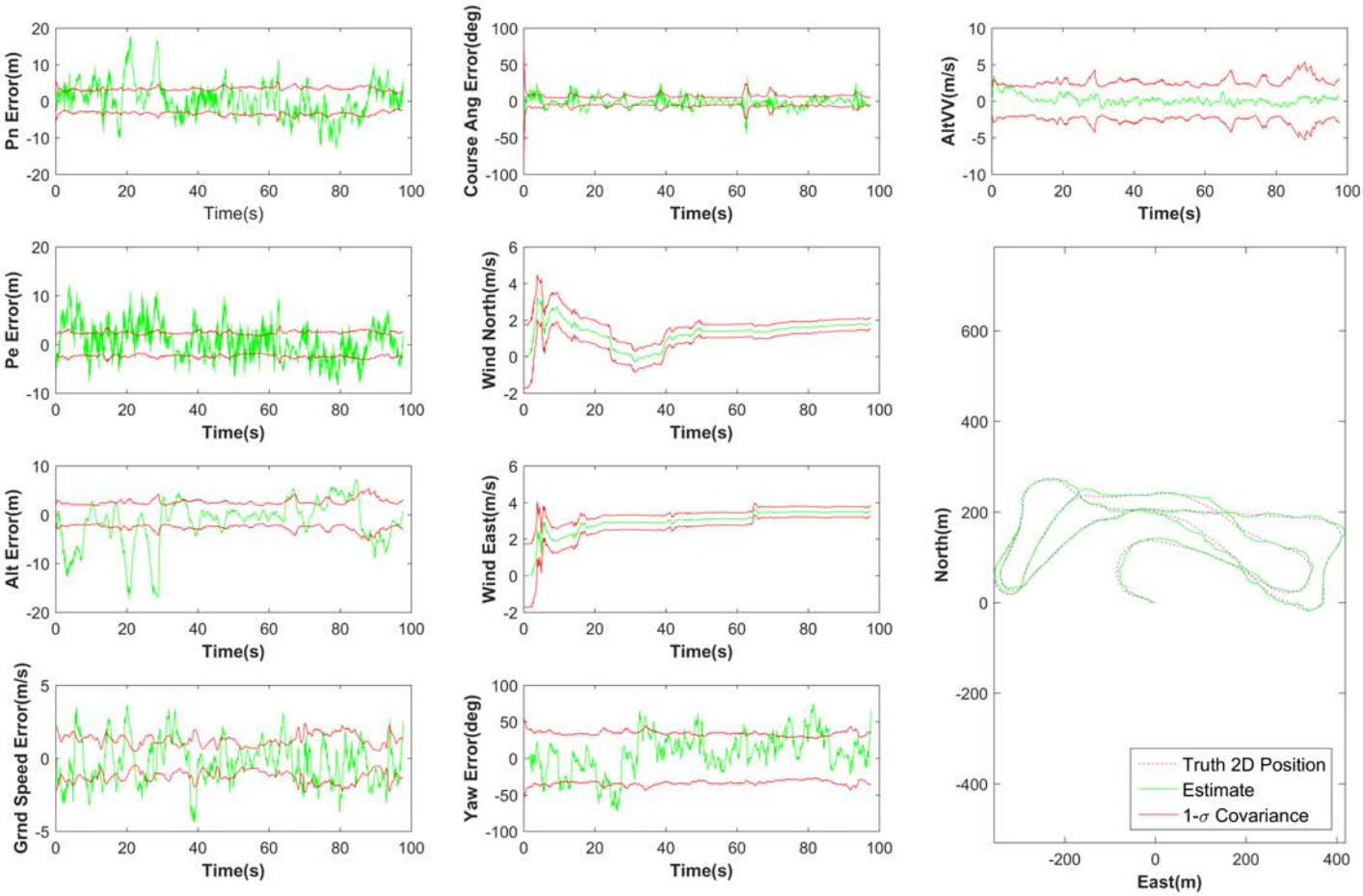



**Figure 49. Longloop With $VO_R$ and Two Ranges (longloopmm$VO_R$RR)**

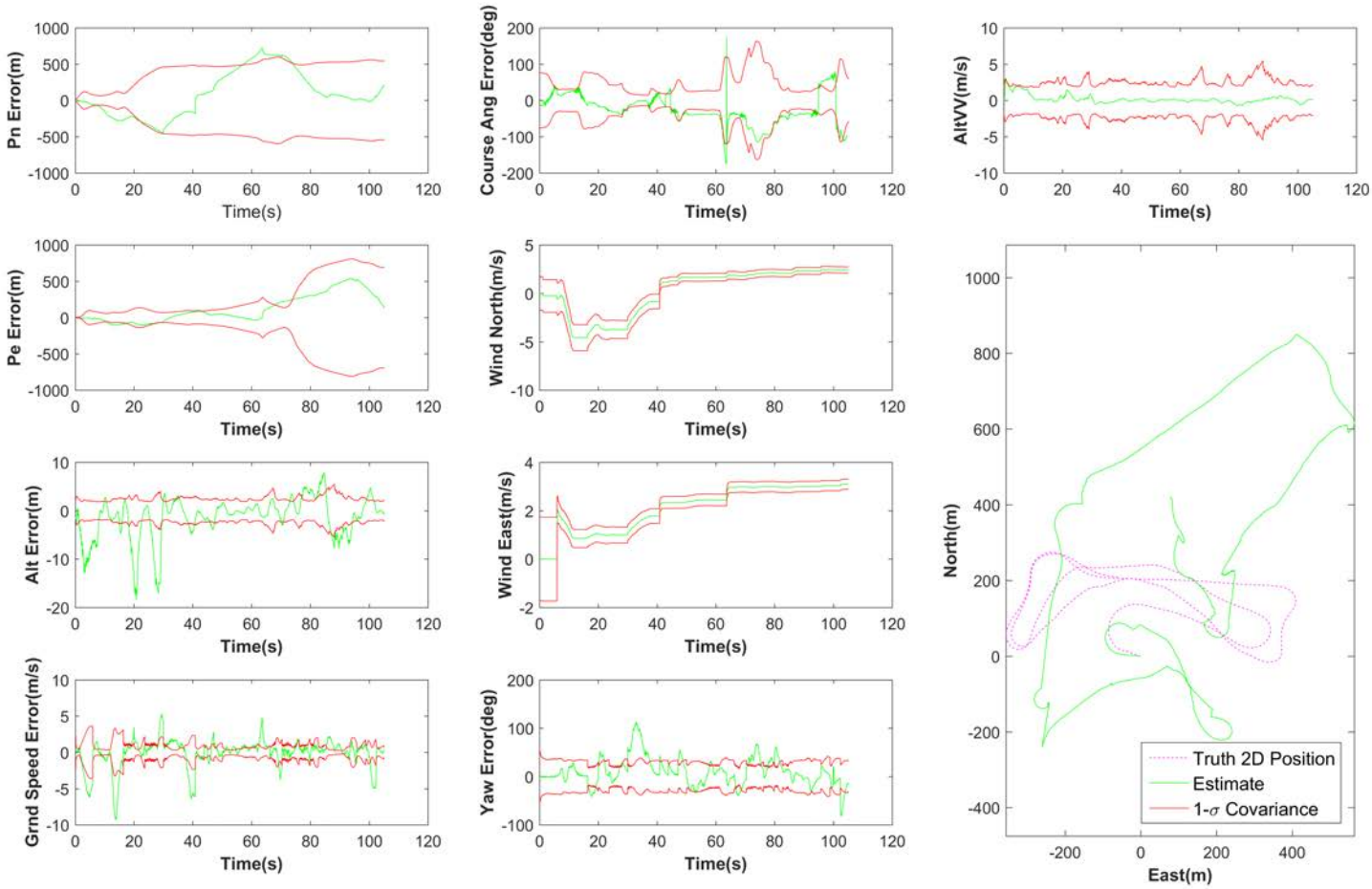



**Figure 50. Longloop SimVO Only (longloopmmSimVO)**

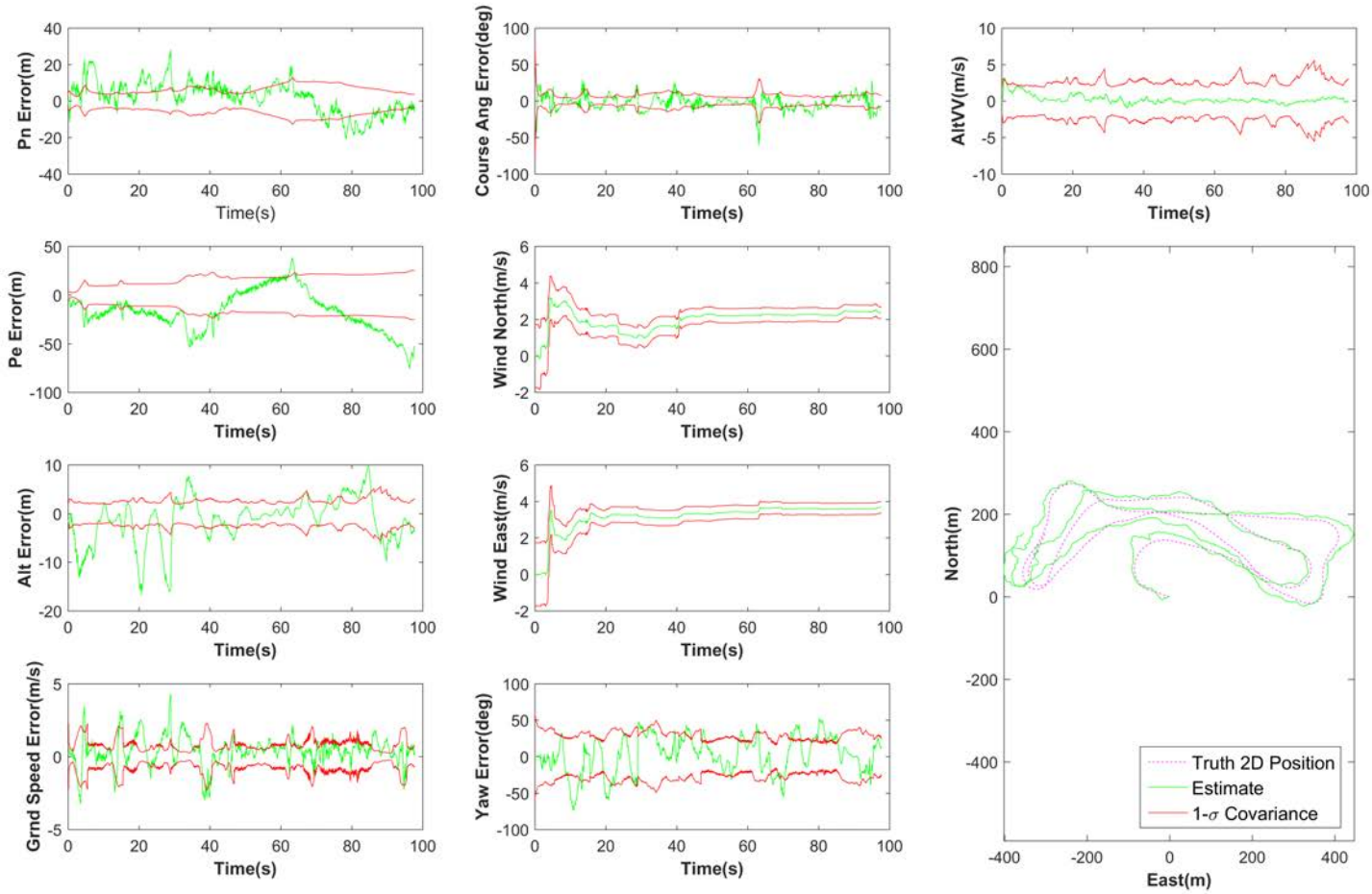



**Figure 51. Longloop With SimVO and One Range (longloopmmSimVOR)**

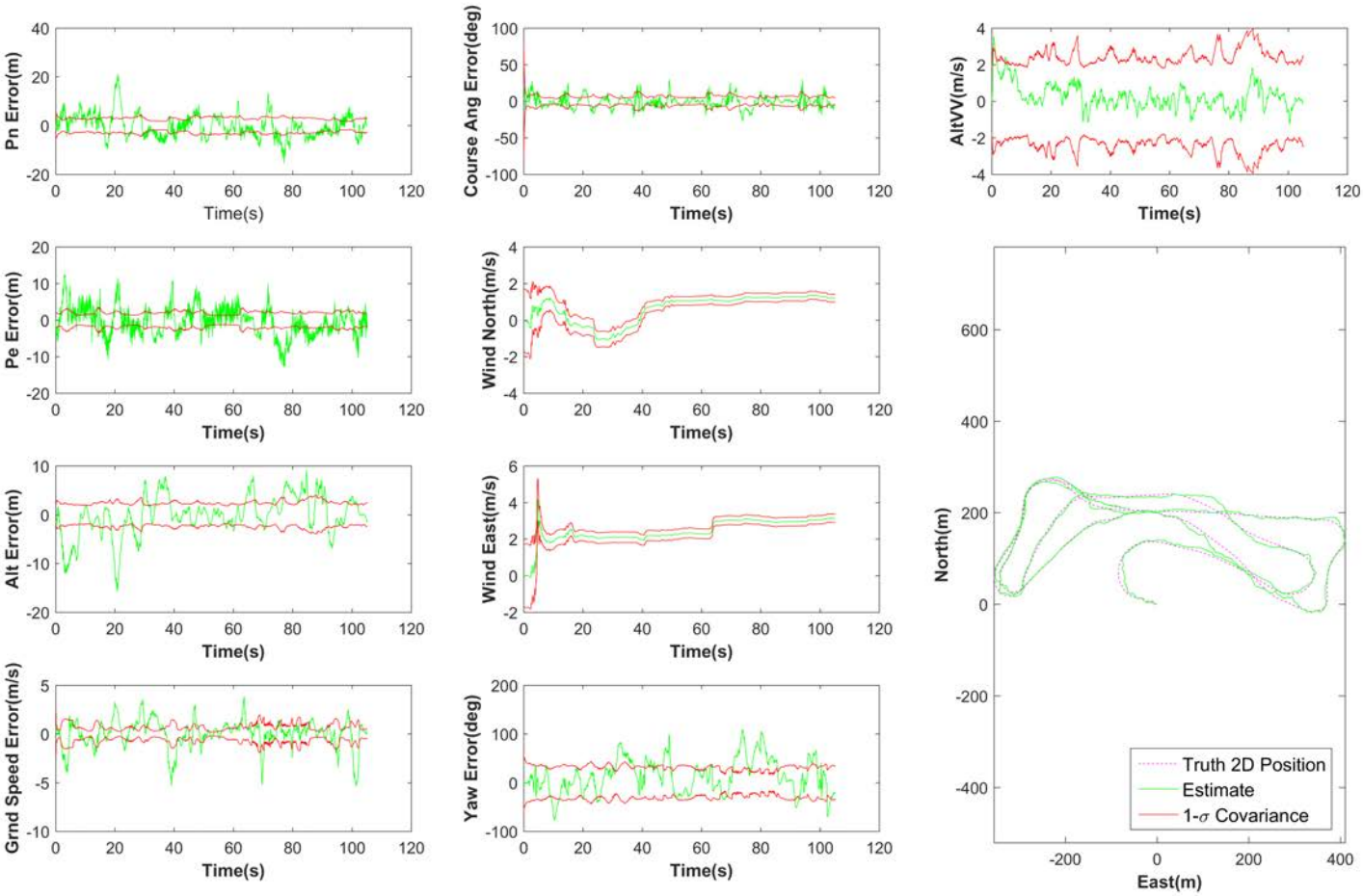



**Figure 52. Oneloop With SimVO and Two Ranges (longloopmmSimVORR)**

# Bibliography


1. M. Planck, "Nerve cells in optic flow," Max Planck Institute of Neurobiology, Tech. Rep., 2010. [Online]. Available: https://www.mpg.de/research/nerve-cells-in-optic-flow

2. G. Conte and P. Doherty, "An integrated UAV navigation system based on aerial image matching," in *IEEE Aerospace Conference Proceedings*, 2008.

3. HobbyExpress, "12 Foot Telemaster ARF w/Drop Box (OVERSIZE)." [Online]. Available: http://www.hobbyexpress.com/12{_}foot{_}telemaster{_}arf.htm

4. 3DR, "Pixhawk Autopilot." [Online]. Available: https://pixhawk.org/modules/pixhawk

5. SECO USA INC, "Udoo Quad." [Online]. Available: http://www.udoo.org/udoo-dual-and-quad/

6. USR IOT, "UART Serial Ethernet Modules." [Online]. Available: http://www.usriot.com/p/uart-serial-ethernet-modules/

7. Allied Vision, "Prosilica GT1290." [Online]. Available: https://www.alliedvision.com/en/products/cameras/detail/ProsilicaGT/1290.html

8. Intel, "NUC Board D53427RKE." [Online]. Available: http://ark.intel.com/products/74484/Intel-NUC-Board-D53427RKE{#}@productimages

9. ENSCO, "ENSCO Pointer 4 . 0 Evaluation Kit," pp. 0–1, 2015.

10. B. W. Randal and T. W. McLain, *Small Unmanned Aircraft Theory and Practice*. Princeton, 2012.

11. Department of Defense, "Unmanned Systems Integrated Roadmap," pp. 2–153, 2013.

12. M. J. Veth, "Fusion of Imaging and Inertial Sensors for Navigation," PhD Dissertation, Air Force Institute of Technology, 2006.

13. J. Rohde, J. Png, J. Raquet, and D. Jacques, "Vehicle Navigation in Dense Urban Environments Using GPS , Image , and IMU Measurements from Commercial Off-The-Shelf Sensors," *ION Pacific PNT*, 2015.

14. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): part I The Essential Algorithms," *Robotics & Automation Magazine*, vol. 2, pp. 99–110, 2006.

15. M. J. Milford and G. F. Wyeth, "Single camera vision-only SLAM on a suburban road network," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2008, pp. 3684–3689.

16. D. Nistér, O. Naroditsky, and J. Bergen, "Visual Odometry for Ground Vehicle Applications," *Journal of Field Robotics*, vol. 23, pp. 3–20, 2006.

17. USAF, *Air Force Future Operating Concept*, 2015.

18. NIMA, "Department of Defense World Geodetic System 1984," Tech. Rep., 2004.

19. P. S. Maybeck, *Stochastic Models: Estimation and Control: Volume 2*, 1982.

20. P. Maybeck, *Stochastic Models: Estimation and Control: Volume 1*, 1979.

21. *Scorpion - Air Force Institute of Technology - ANT Center*. [Online]. Available: http://github.com/ANTCenter

22. T. R. Penn, "All Source Sensor Integration Using an Extended Kalman Filter," Masters thesis, Air Force Institute of Technology, 2012.

23. R. G. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed. Wiley New York, 1992.

24. D. Scaramuzza and F. Fraundorfer, "Tutorial: Visual odometry," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

25. D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC," *Robotics and Automation*, pp. 4293–4299, 2009.

26. C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," *Robotics and Automation*, 2014.

27. A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3946–3952, 2008.

28. K. Konolige and M. Agrawal, "Large Scale Visual Odometry for Rough Terrain," *Robotics Research*, 2010.

29. C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Procedings of the Alvey Vision Conference 1988*, pp. 147–151, 1988. [Online]. Available: http://www.bmva.org/bmvc/1988/avc-88-023.html

30. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision*, pp. 430–443, 2006.

31. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," pp. 117–156, 1998.

32. R. Maini and H. Aggarwal, "Study and Comparison of Various Image Edge Detection Techniques," *International Journal of Image Processing*, no. 31, 2009.

33. D. G. Lowe, "Distinctive image features from scale invariant keypoints," *Int'l Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

34. T. Machin, "Real-time implementation of vision-aided monocular navigation for small fixed-wing unmanned aerial systems," Masters thesis, Air Force Institute of Technology, 2016.

35. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Computer Science*, pp. 404–417, 2006.

36. "OpenCV." [Online]. Available: http://opencv.org/

37. P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," *Computer Science*, vol. 7577 LNCS, no. PART 6, pp. 214–227, 2012.

38. P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces," *British Machine Vision Conference*, pp. 13.1–13.11, 2013.

39. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, 1992.

40. D. Carson, "Aerial Visual-Inertial Odometry Preformance Evaluation," Masters thesis, Air Force Institute of Technology, 2017.

41. D. T. Venable, "Improving Real World Performance for Vision Navigation in a Flight Environment," Ph.D. dissertation, Air Force Institute of Technology, 2016.

42. R. H. Wu and S. A. Davidson, "An Alternative Positioning Navigation and Timing Concept based on Diverse Ranging," *IEEE/ION*, 2014.

43. J. N. Gross, Y. Gu, and M. B. Rhudy, "Robust UAV Relative Navigation With DGPS, INS, and Peer-to-Peer Radio Ranging," *Ieee Transactions On Automation Science And Engineering*, vol. 12, no. 3, 2015.

44. P. Systems, "Wave Relay." [Online]. Available: http://www.persistentsystems.com/integration-board-gen4/

45. SIG MFG, "SIG Rascal 110." [Online]. Available: http://www.sigmfg.com/IndexText/SIGRC84ARFB.html

46. RFDESIGN PTY LTD, "RFD900+ Modem." [Online]. Available: http://store.rfdesign.com.au/rfd-900p-modem/

47. Michael Oborne, "Mission Planner." [Online]. Available: http://ardupilot.org/planner/docs/mission-planner-overview.html

48. Dualcomm, "USB Powered 10/100Base-T Network Tap." [Online]. Available: http://www.dual-comm.com/port-mirroring-LAN{_}switch.htm

49. A. S. Huang, E. Olson, and D. C. Moore, "LCM: Lightweight Communications and Marshalling," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, no. Lcm, pp. 4057–4062, 2010.

50. T. Committee, "IEEE Std 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *Society*, vol. 2008, no. July, pp. c1 –269, 2008.

51. 3DR, "Dronekit." [Online]. Available: http://dronekit.io

52. "Pymba." [Online]. Available: https://github.com/morefigs/pymba

53. A. Chakraborty, "Cooperative Localization for Fixed Wing Unmanned Aerial Vehicles," *Position, Location and Navigation Symposium (PLANS), 2016 IEEE/ION*, 2016.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704–0188*



| 1. REPORT DATE *(DD–MM–YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From — To)* |
|---|---|---|
| 23–03–2017 | Master's Thesis | Sept 2015 — Mar 2017 |

**4. TITLE AND SUBTITLE**

Small Fixed-wing Aerial Positioning Using Inter-vehicle Ranging Combined With Visual Odometry

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Fain, Benjamin M., Capt USAF

**5d. PROJECT NUMBER**

16G328

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

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-ENG-MS-17-M-027

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

Adam Rutkowski (adam.rutkowski@us.af.mil)
Air Force Research Lab Munitions Directorate
101 W. Eglin Blvd
Eglin AFB, FL 32542
COMM 850-882-4600

**10. SPONSOR/MONITOR'S ACRONYM(S)**

AFRL/RWWI

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

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**



**14. ABSTRACT**

There has been increasing interest in developing the ability for small unmanned aerial systems (SUAS) to be able to operate in environments where GPS is not available. This research considers the case of a larger aircraft loitering above a smaller GPS-denied SUAS. This larger aircraft is assumed to have greater resources which can overcome the GPS jamming and provide range information to the SUAS flying a mission below. This research demonstrates that using a ranging update combined with an aircraft motion model and visual odometry can greatly improve the accuracy of a SUASs estimated position in a GPS-denied environment.

**15. SUBJECT TERMS**

Ranging, Visual Odometry, SUAS, DRMS, Inter-vehicle, Motion model, Range to Aircraft, Absolute Position

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Dr. John F. Raquet, AFIT/ENG |
| U | U | U | UU | 120 | 19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4580; John.Raquet@afit.edu |