

Dissertations and Theses

5-2018

Vision-Aided Navigation using Tracked Lankmarks

Nicodemus Myhre

Follow this and additional works at: <https://commons.erau.edu/edt>



Part of the [Aerospace Engineering Commons](#)

Scholarly Commons Citation

Myhre, Nicodemus, "Vision-Aided Navigation using Tracked Lankmarks" (2018). *Dissertations and Theses*. 390.

<https://commons.erau.edu/edt/390>

This Thesis - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

VISION-AIDED NAVIGATION USING TRACKED LANDMARKS

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Nicodemus Myhre

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

May 2018

Embry-Riddle Aeronautical University

Daytona Beach, Florida

VISION-AIDED NAVIGATION USING TRACKED LANDMARKS

by

Nicodemus Myhre

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Richard J. Prazenica, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

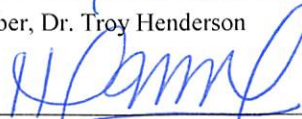
THESIS COMMITTEE




Chairman, Dr. Richard J. Prazenica



Member, Dr. Troy Henderson



Member, Dr. Hever Moncayo



Member, Dr. Ebenezer Gnanamanickam



Graduate Program Coordinator, Dr. Magdy Attia

4.18.2018

Date



Dean of College of Engineering, Dr. Maj Mirmirani

4/18/2018

Date



Vice Chancellor, Academic Support, Dr. Christopher Grant

4/18/2018

Date

ACKNOWLEDGMENTS

No thesis is a one-man project, and I'd like to take a few words to thank those who have helped make this one possible.

First, I'd like to thank my advisor Dr. Prazenica for his excellent teaching and mentorship. He has taught me a great deal about control theory, technical writing, and plain common sense. I also wish to thank my committee members, Dr. Henderson, Dr. Moncayo, and Dr. Gnanamanickam for their service and suggestions.

I'd like to thank my good friend Andres Armijos Chavez who gave considerable aid throughout with the flight testing, camera calibrations, and code debugging. I also want to thank him for his moral support on the long days when I could think of nothing but going home. Andres, I wish you the best in your future endeavors. Thanks for being a great friend.

I also need to thank my parents for their support, and my mom in particular for making sure I stayed sane and healthy while away from my home in Tennessee. I'm very grateful for all that they have poured into raising me and taking care of me even in Daytona. I love you both very much.

God Bless,

Nicodemus Myhre

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
SYMBOLS.....	ix
ABBREVIATIONS	xi
NOMENCLATURE	xii
ABSTRACT.....	xiii
1. Introduction	1
1.1. Motivation.....	1
1.2. Thesis Outline.....	5
2. Literature Review	8
2.1. Non a Priori Methods	8
2.1.1. Non-Mapping Algorithms	8
2.1.2. Simultaneous Localization and Mapping (SLAM).....	9
2.2. A Priori Methods	10
2.2.1. Map-based approaches.....	11
2.2.2. Landing Target	11
2.2.3. Landmark Navigation	12
3. EKF Filtering.....	13
3.1. Overview.....	14
3.2. Process Model.....	16
3.3. Landmark-Based Measurement Models	18
3.3.1. Monocular Measurement	19
3.3.2. Stereo Measurement.....	22
3.4. Underweighting for Nonlinear Instability.....	23
3.5. Observability Analysis.....	27
3.5.1. Monocular Single-Landmark Unobservability.....	29
3.5.2. Additional Landmarks.....	34
3.5.3. Stereo.....	35
3.5.4. Nonlinear Effects.....	36
4. Simulation	39
4.1. Sensor Simulation.....	39
4.2. Test Cases	40
4.2.1. Process and Measurement Noise.....	40
4.2.1. Number of Landmarks	49
4.2.1. Flight Path.....	53
4.2.2. Monocular vs. Stereo.....	57
5. Quadcopter Flight Testing.....	61

5.1.	Flight Test Setup.....	61
5.2.	IMU and Camera Parameters	63
5.3.	Camera Calibration.....	64
5.4.	Color Detection.....	65
5.5.	Post-Processing Results	66
6.	Hybrid Navigation.....	69
6.1.	Homography Measurement	69
6.2.	Hybrid Results	72
7.	Conclusion.....	75
8.	Recommendations	77

LIST OF TABLES

Table 1: IMU and Camera, Noise and Bias	39
---	----

LIST OF FIGURES

Figure 1: EKF Framework	13
Figure 2: Nonlinear Re-convergence Instability	24
Figure 3: Overly Optimistic Covariance Convergence.....	25
Figure 4: Underweighted Re-convergence	27
Figure 5: Simulated IMU Data	39
Figure 6: Simulation flight overview with flight test noise levels.....	41
Figure 7: Position estimation with increasing noise level.	42
Figure 8: Estimation error and covariance bounds with increasing noise.	43
Figure 9: X-position error noise dependence.....	44
Figure 10: Velocity estimation with increasing noise level.....	45
Figure 11: Velocity estimation error with increasing noise level.	46
Figure 12: Attitude estimation with increasing noise level.	47
Figure 13: Attitude estimation error with increasing noise level.....	48
Figure 14: Estimation with two landmarks	50
Figure 15: Side view of hover.....	51
Figure 16: Estimation with a single landmark	52
Figure 17: Hovering and abrupt maneuvers.....	54
Figure 18: Side view; extended hover flightpath.....	55
Figure 19: Hovering and abrupt maneuvers with one landmark.....	56
Figure 20: Stereo setup, hovering flightpath, and one landmark	58
Figure 21: Side View; Stereo Measurement	59
Figure 22: Stereo setup, original spiraling flightpath, and one landmark.....	60
Figure 23: Iris Quadcopter fitted with GoPro	61
Figure 24: Aerial View of Landmark Cluster	62
Figure 25: Sample GPS Track	62
Figure 26: Sample IMU Data.....	62
Figure 27: Flight test snippet for IMU uncertainty estimation	63
Figure 28: Checkerboard pattern for Calibration.....	65
Figure 29: Color Detection Process Overview	66
Figure 30: Post-processing position estimation	67
Figure 31: Post-processing velocity estimation	68
Figure 32: Post-processing attitude estimation	68

Figure 33: Hybrid filter mission overview.....	69
Figure 34: Homography Relationship.....	70
Figure 35: Homography measurement simulation.....	71
Figure 36: Hybrid navigation method simulation.....	73
Figure 37: Hybrid post-processing estimation.....	74

SYMBOLS

t	Time
X_k	State Vector
\hat{X}_k	Estimated State Vector
P_k	State Autocovariance Matrix
P_N	Vehicle Position in Inertial Frame
V_N	Vehicle Velocity in Inertial Frame
ρ	Vehicle Attitude – Roll, Pitch, Yaw
ϕ	Roll Angle
θ	Pitch Angle
ψ	Yaw Angle
b_{acc}	Accelerometer Bias
b_{gyr}	Rate Gyro Bias
g	Gravitation Constant
\mathbf{g}	Gravitation Vector
A_k	EKF Process Jacobian
G_k	EKF Process Noise Jacobian
C_k	EKF Measurement Jacobian
Q_k	EKF Process Noise Covariance
R_k	EKF Measurement Noise Covariance
K_k	Kalman Gain
Δt_s	Time Step
R_N^B	Rotation Matrix from Navigation Frame to Body Frame
R_B^C	Rotation Matrix from Body Frame to Camera Frame
R_N^C	Rotation Matrix from Navigation Frame to Camera Frame
L_B^N	Euler Angle Dynamics Matrix
a^B	Acceleration in Body Frame
ω^B	Angular Rates in Body Frame
p	Roll Rate
q	Pitch Rate
r	Yaw Rate
Y	EKF Measurement
M	Camera Intrinsic Matrix
f_x	Camera X-direction Focal Length
f_y	Camera Y-direction Focal Length
c_x	Camera Principal Point X-position
c_y	Camera Principal Point Y-position
Π_0	Monocular Camera Projection Matrix
\mathbf{x}_i	Position of i-th Landmark in Camera Frame
\mathbf{X}_i	Position of i-th Landmark in Inertial Frame
G	Vehicle Pose
B_k	Nonlinear Kalman Uncertainty

U_k	Underweighting Substitute Matrix
β	Underweighting Factor
σ	Underweighting Threshold
P_o	Observability Matrix
Y_P	Measurement Position Jacobian
Y_ρ	Measurement Attitude Jacobian
x^N	Vector to Landmark from Vehicle CG in Inertial Frame
x^B	Vector to Landmark from Vehicle CG in Body Frame
x^C	Vector to Landmark from Vehicle CG in Camera Frame
λ	Arbitrary Scale Factor
\mathbf{v}_u	Unobservable State-Space Direction
δX	State Displacement
X^N	Position of Landmark in Inertial Frame
σ_{a_x}	Accelerometer X-direction Standard Deviation
σ_{a_y}	Accelerometer Y-direction Standard Deviation
σ_{a_z}	Accelerometer Z-direction Standard Deviation
σ_p	Rate Gyro Roll Rate Standard Deviation
σ_q	Rate Gyro Pitch Rate Standard Deviation
σ_r	Rate Gyro Yaw Rate Standard Deviation
$\sigma_{x,cam}$	Camera Intrinsic Plane X-direction Standard Deviation
$\sigma_{y,cam}$	Camera Intrinsic Plane Y-direction Standard Deviation
$\sigma_{z,cam}$	Camera Stereo Distance Measurement Standard Deviation
b_{a_x}	Accelerometer X-direction Bias
b_{a_y}	Accelerometer Y-direction Bias
b_{a_z}	Accelerometer Z-direction Bias
b_p	Rate Gyro Roll Rate Bias
b_q	Rate Gyro Pitch Rate Bias
b_r	Rate Gyro Yaw Rate Bias
$b_{x,cam}$	Camera Intrinsic Plane X-direction Bias
$b_{y,cam}$	Camera Intrinsic Plane Y-direction Bias
H	Homography Matrix
R	Relative Rotation Between Frames
T	Relative Translation Between Frames
X_1	Projection of Points in Frame 1
X_2	Projection of Points in Frame 2

ABBREVIATIONS

GPS	Global Positioning System
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
SLAM	Simultaneous Localization and Mapping
LIDAR	Light Detection and Ranging
NED	North-East-Down
UKF	Unscented Kalman Filter

NOMENCLATURE

a priori	An algorithm using prior information about its environment
non a prior	An algorithm with no initial information about its environment
Landmark	A uniquely identifiable feature with known position
Feature point	A prominent unidentified feature detected in an image
Pose	The position and rotation of a frame of reference
Ground truth	The true state of a flight vehicle

ABSTRACT

Myhre, Nicodemus MSAE, Embry-Riddle Aeronautical University, May 2018. Vision-Aided Navigation Using Tracked Landmarks.

This thesis presents vision-based state estimation algorithms for autonomous vehicles to navigate within GPS-denied environments. To accomplish this objective, an approach is developed that utilizes a priori information about the environment. In particular, the algorithm leverages recognizable ‘landmarks’ in the environment, the positions of which are known in advance, to stabilize the state estimate. Measurements of the position of one or more landmarks in the image plane of a monocular camera are then filtered using an extended Kalman filter (EKF) with data from a traditional inertial measurement unit (IMU) consisting of accelerometers and rate gyros to produce the state estimate. Additionally, the EKF algorithm is adapted to accommodate a stereo camera configuration to measure the distance to a landmark using parallax. The performances of the state estimation algorithms for both the monocular and stereo camera configurations are tested and compared using simulation studies with a quadcopter UAV model. State estimation results are then presented using flight data from a quadcopter UAV instrumented with an IMU and a GoPro camera. It is shown that the proposed landmark navigation method is capable of preventing IMU drift errors by providing a GPS-like measurement when landmarks can be identified. Additionally, the landmark method pairs well with non a priori measurements for interims when landmarks are not available.

1. Introduction

1.1. Motivation

Interest in unmanned autonomous systems (UAS) has grown rapidly in recent years, and the ability of a vehicle (e.g. a quadcopter) to reliably estimate its state (position, velocity, orientation) is a critical enabling task for UAS missions. A myriad of methods to estimate the state of a UAS have been developed with, by far, the most common being the filtered combination of IMU and GPS data. The IMU-GPS solution has become the widely accepted 'gold standard' for vehicle state estimation; however, it has its limitations. In particular, there are many proposed UAS missions in which GPS signals are unreliable or non-existent (Chavez, et al., 2017). For example, the GPS signal is known to be unavailable or unreliable in urban environments due to interference from large and closely spaced buildings (Prazenica, Hielsberg, Sharpley, & Kurdila, 2013). GPS is also completely unavailable in deep space environments such as asteroid exploration and sampling missions, necessitating some non-GPS state estimation algorithm for these missions (Prazenica, et al., 2016) (Perez, et al., 2016). As a final example, for some missions that do have access to the GPS signal, it may nonetheless be desirable to have alternative measurements available as a backup in the event of signal degradation or loss, for example during landing (Mondragón, Campoy, Martínez, & Olivares-Méndez, 2010). Thus, the development of non-GPS navigation methods has been strongly motivated. In particular, as the size, weight, and price of high-resolution and high-speed cameras are all dropping rapidly, vision-based navigation systems are becoming both the most natural and the most economic attempt at an alternate method of state estimation to replace a GPS measurement when necessary. As a result, vision-based state estimation has received a

great deal of attention in recent years (Kanade, Amidi, & Ke, 2004) (Prazenica, et al., 2006).

Vision-based navigation methods can be partitioned into two categories: a priori and non a priori (Zhao, et al.). An a priori algorithm is one in which information about the mission environment is known prior to the mission. Conversely, a non a priori algorithm does not take advantage of any previously available information about the navigation environment. As a result, non a priori algorithms have an immediately apparent advantage in terms of their applicability in nearly any conceivable environment, which a priori algorithms do not share. Thus, it is no surprise that much research into vision-based navigation has been directed toward non a priori methods (Kehoe, Causey, Arvai, & Lind, 2006) (Caballero, Merino, Ferruz, & Ollero, 2009) (Zhao, et al., 2016). The classic example of non a priori algorithms is the broad category of Simultaneous Localization and Mapping (SLAM) algorithms (Smith, Self, & Cheeseman, 1987) (Caballero, Merino, Ferruz, & Ollero, 2009) (Bailey, Nieto, Guivant, Stevens, & Nebot, 2006). However, as the name implies, SLAM algorithms not only estimate the state of the vehicle, but also the layout of the environment, which often increases the computational cost. An unfortunate conclusion implied by SLAM is that any non a priori algorithm that does not simultaneously ‘map’ some feature(s) in the environment will be subject to an inevitable drift in the state estimate over time following the pattern of a random walk (Bonin-Font, Ortiz, & Oliver, 2008). Indeed, in a very loose sense of the definition, SLAM algorithms are nearly a priori, since, in many cases, some ‘information’ about the environment is known (or rather assumed) beforehand; namely, that the layout of the environment is a constant, that it has identifiable features which will persist throughout the duration of the mission,

and that these features can be returned to and re-identified as many times as needed or desired. Zhao et al. (2016) refer to this as ‘loop closure’. In short, the SLAM algorithm has a memory, or as in (Bonin-Font, Ortiz, & Oliver, 2008), it is a map-building-based navigation method.

Other non a priori algorithms, such as feature tracking, homography-based, etc., make no such assumptions (Zhao, Liny, Pengy, Chenz, & Leez, 2012). These algorithms only assume that the environment layout does not change quickly; i.e., the scene does not change significantly between frames. These algorithms have no memory, or equivalently, no map (Bonin-Font, Ortiz, & Oliver, 2008). For example, a feature tracking algorithm tracks random feature points between a pair of frames and generates an estimate of the displacement between those two frames. Once this is complete, the algorithm forgets the first frame and the estimated displacement and eventually will forget even the feature points it was tracking. Thus, any error in a calculated displacement vector from forgotten features will persist in the estimate forever. This error could, in theory, be corrected later on if there were some form of persistent knowledge about the environment, but this is not the case. All the sensor information available in a non a priori algorithm gives information about change in position, change in velocity, or change in attitude. Thus, there is still an integration of a noisy signal occurring. Hopefully the signal will have significantly reduced noise due to integrated sensor measurements, but it will still be noisy, and a random walk in state estimation error will result.

Thus, we can at last see that a priori algorithms have a significant advantage in that they have a ‘memory’ of fixed points in the environment stored away from the outset, which can stabilize them against the drift of a random walk over time. Additionally, a priori

algorithms in the proper sense of the definition do not need to produce this information in real time, as in the non a priori SLAM approach, potentially reducing computational cost. Bonin-Font, Ortiz, & Oliver (2008) refer to this as a ‘map-based’ algorithm. By providing a ‘GPS-like’ measurement at each frame, as opposed to the between-frame comparison given by many non a priori measurements, a map-based approach, or a landmark-based approach as in this thesis, can give state estimation results comparable to those of the GPS-IMU solution. In many scenarios, the benefits of the drift-free estimates produced at relatively low cost may outweigh the loss of the versatility that non a priori algorithms provide. Additionally, as we shall see later on, the landmark method in particular can be paired with a non a priori algorithm to maintain this versatility while simultaneously reaping the benefits of GPS-like information.

This thesis details the development of an a priori vision-based navigation algorithm for autonomous missions. While the a priori nature of this method limits some of its applicability, it has the potential to be more computationally efficient than non a priori SLAM algorithms -- a critical feature in many applications. Additionally, most real-world missions have some a priori information available, and even if none is available it can sometimes be built-in (Lange, Sunderhauf, & Protzel, 2009). For example, in urban environments, there is a plethora of information available in advance. Targets ranging from the features of a unique building to memorial statues can be recognized, roads can be followed, and buildings can be avoided. As a second example, in asteroid exploration missions, certain reference features may be known in advance, or as in (Chavez, Myhre, & Prazenica, 2017) such features can be built into the mission by planting a landing target at the starting point for the mission.

The estimation filter presented in this paper is far from the first attempt to use a priori information for navigation. Others have attempted various methods including reference terrain navigation using a digital terrain elevation database as in (Lee, Kim, & Bang, 2013) (Lee, Kim, & Bang, 2014) or geo-referencing as in (DeAngelo & Horn, 2016) (DeAngelo & Horn, 2017). Additionally, the known size of a feature in view of the camera can be used to determine the range to that feature (Lange, Sunderhauf, & Protzel, 2009). As a second example, reference terrain navigation takes advantage of a complete terrain elevation map along with, usually, a radar altimeter to keep track of the vehicle's location. The approach developed in this thesis is unique in its simplicity and applicability in a variety of situations. In particular, all that is needed is an identification of at least one unique feature in the environment with known or definable position. Indeed, the ID does not even have to be visual in nature; however, vision-based identifications are likely to be the most common and are the only methods considered in this work.

The filter presented here uses the availability of identifiable 'landmarks' in the environment with known (a priori) position. The algorithm filters the information about the identification with IMU data to generate the pose estimation. This filtering process takes the form of a basic first-order Extended Kalman Filter (EKF). The algorithm has been adapted to use either monocular vision, or a stereo setup in which two cameras work in tandem to provide additional information about the distance to a tracked 'landmark' using parallax. Both cases are developed and tested in this work.

1.2. Thesis Outline

The thesis is organized as follows. In Chapter 2, an overview of the relevant literature on UAV state estimation is presented. This Chapter provides the background for

the landmark-based state estimation approach outlined in this work. The distinction between a priori and non a priori algorithms is discussed and the significance of a general ID-based state estimation algorithm is underlined.

Next, Chapter 3 gives an expository of the landmark filter for both the monocular and stereo variants. The mathematics, geometry, and kinematics forming the process and measurement models used in the prediction and update steps of the EKF are presented. The practical organization and implementation of the filter are outlined, the use of ‘underweighting’ to deal with the nonlinear instabilities associated with landmarks leaving and entering view is detailed, and an observability analysis of the algorithm(s) is given. These analyses demonstrate the theoretical capabilities and limitations of the state estimation filter in providing a substitute for GPS measurements as in traditional state estimation schemes.

Chapter 4 gives an overview of the quadcopter simulation testing of the landmark filter. The data collection and simulated sensor data generation are outlined. The simulation gives an excellent opportunity to test the filter in several situations and support the theoretical results from the previous chapter. These multiple test situations give insight into the filter’s capabilities and performance, and are also enumerated here. Among these are evaluations of sensor noise tolerance, performance on different flight paths, and the number of landmarks required for certain tasks.

Chapter 5 presents an off-line implementation of the landmark filter on actual sensor data from UAV flight tests. A description of the sensor array layout and its elements in each of the various flight tests is given here for documentation purposes. Results are presented from off-line filter testing using the data from these flight tests. These results

demonstrate the potential for the algorithm to be applied on-line in practical situations, as the filter is not computationally costly.

Lastly, Chapter 6 outlines a key application of landmark navigation in a hybrid navigation scheme. While the landmark filter has unique advantages in terms of its GPS-like measurement, it also possesses limitations that traditional navigation schemes such as optical flow, visual odometry, etc. do not. Thus, it makes sense to pair the landmark filter with a robust non a priori frame-to-frame type algorithm to get all the best features of each navigation method: GPS-like measurement from the landmarks and robustness in the absence of landmarks from the non a priori method. The non a priori method used here is a homography-based approach that uses tracked feature points (Chavez, et al., 2017) (Chavez, Myhre, & Prazenica, 2017).

2. Literature Review

This review is broken into two sections discussing a priori and non a priori algorithms in greater depth. It is intended that this chapter will clarify how the methods developed in this thesis fit into the larger picture of vision-based state estimation and vision aided navigation.

2.1. Non a Priori Methods

A non a priori algorithm is one in which no information about the environment is given beforehand to the algorithm (Zhao, et al.). The algorithm expects no mission-specific features or attributes in the vehicle's surroundings. It may assume that features will exist, but it has no hard-coded knowledge of what features may or may not exist in the scene. The algorithm simply uses whatever features are found in the scene. For example, most algorithms based on feature point tracking are non a priori. They do not consider any information about the points they are tracking other than an assumption of fixed location, their prominence, which makes them easy to detect, and their relative motion between frames.

2.1.1. Non-Mapping Algorithms

A typical non a priori algorithm may use the relative motion of individual feature points between frames to give an estimation of the relative motion of the vehicle in the time period between those frames, for example (Webb & Prazenica, 2007). Feature points can be extracted in a number of different ways such as the well-known Lucas-Kanade algorithm, which is a popular method (Lucas & Kanade, 1981). If there are sufficiently many points available, the relative pose can be reconstructed completely up to an unknown

scale factor as in the eight-point algorithm for example, which is based on epipolar geometry (Ma, Kosecka, Soatto, & Sastry, 2001). Integrating the estimated motion between frames beginning at a known initial state can then give an estimate of the current state. This technique is the basis of visual odometry, and homography estimation (Strydom, Denuelle, & Srinivasan, 2016) (Zhao, Liny, Pengy, Chenz, & Leez, 2012). A separate category of non a priori algorithms is the group of optical flow algorithms which uses intensity gradients in an image to extract the same relative information as a point matching algorithm (Aggarwal & Nandhakumar, 1988). All such relative-motion algorithms, however, suffer from the drift in time of a random walk resulting from the integration of a noisy (random) signal as observed in (Bonin-Font, Ortiz, & Oliver, 2008). Unfortunately, this is typical for non a priori algorithms, since only the relative motion between frames is measured. Thus, a non a priori algorithm is akin to a more accurate IMU that does not drift as quickly.

2.1.2. Simultaneous Localization and Mapping (SLAM)

In order to solve the problem of random walk drift in time that is inevitable in a priori algorithms, a map of some sort must be introduced. Ideally, this map would be completely specified across the entire domain of the mission prior to the mission taking place. Such methods are briefly outlined in Chapter 2.2.1 on map-based navigation approaches. However, in many cases any available map is incomplete or even non-existent, such as in planetary or asteroid exploration missions (Chavez, Myhre, & Prazenica, 2017). In cases like these, a frequent idea is to have the vehicle build its own map as the mission progresses. Such approaches are known broadly as Simultaneous Localization and Mapping (SLAM) algorithms. Smith, Self, and Cheeseman (1987)

provided a solution to the simultaneous estimation of vehicle location and relative location of features in the environment through stochastic methods, and numerous other SLAM approaches and applications have been developed since (Lemaire, Lacroix, & Sola, 2005) (Ivey & Johnson, 2006) (Caballero, Merino, Ferruz, & Ollero, 2009). Experience has shown that such a map-building approach can stabilize the vehicle state estimate in time against the random walk errors that plague non-mapping algorithms, while still making no initial assumptions, and thus remaining non a priori. Thus, the non a priori time-drift problem is solved without resorting to a priori knowledge. However, the drift actually does still occur in the spatial domain as observed by Bailey et al. (2006). That is, while the uncertainty buildup in the state does not occur the farther the vehicle travels from the initial *time*, the vehicle state does become more uncertain the farther it travels from the initial *position*. This is nonetheless a significant improvement, and one which has enabled SLAM methods to be used in applications such as the Spirit and Opportunity Mars rovers (Maimone, Cheng, & Matthies, 2007).

2.2. A Priori Methods

While the SLAM approach in its numerous variants provides a viable solution to the visual navigation problem, it still has its shortcomings. First, as already noted, SLAM algorithms do not prevent long-distance drift. Additionally, SLAM algorithms can be computationally intensive, which has the potential to limit applicability. To solve these problems while maintaining the benefit of preventing time-drift, we must resort to a priori methods.

2.2.1. Map-based approaches

The map-based approach to navigation is an obvious approach to navigation with a priori information. The algorithm has access to a map with known (a priori) information about the environment that it compares to its measurements to ascertain the vehicle's position in the environment and to plan its route accordingly. Johnson and Montgomery (2008) give an excellent summary of map-based approaches to autonomous spot-landings on the Moon using available lunar maps; however, the principles at play are applicable in numerous other applications where maps are available. For example, Conte and Doherty (2009) used correlation between camera frames and stored aerial photographs of the flight area to localize the vehicle within the map. Additionally, maps may come in the form of a terrain elevation database, within which a vehicle can locate itself using LIDAR or radar-altimeter measurements (Bingol, Akin, & Koc, 2012).

2.2.2. Landing Target

In a map-based approach, the map may have considerable size, and require a large database for the mission to be viable. The database size can be mitigated by reducing the 'map' to contain knowledge of only a select few key features of interest. This knowledge often does not even come in the form of an image, like in a proper map-based approach. A recognizable landing target is a special case of this 'map reduction' idea in which the landing site has prominent features to enable a precise landing using visual observation of the 'target'. For example, Brockers et al. (2011) used a homography approach to detect and land on an elevated platform. Lange, Sunderhauf, and Protzel (2009) used a designed landing target with distinctive markings and a known size for detection and ranging during a landing. Other studies have followed in the same vein using recognizable landing pads

for stability during the landing (Wenzel, Rosset, & Zell, 2010) (Yang, Scherer, & Zell, 2013) (Li, Garratt, & Lambert, 2014).

2.2.3. Landmark Navigation

In this thesis, we delve into the use of generic landmarks for navigation. As mentioned previously, landmarks can come in a variety of forms. For example, a ‘landmark’ may simply be a feature point or edge in a known map (Atiya & Hager, 1993). Landmarks can also come in the form of known features in a map such as buildings and roads, or even trees and fields (DeAngelo & Horn, 2017). Recently, DeAngelo & Horn (2016) used a relatively sparse map with roads and houses to estimate the horizontal position of a vehicle, but not the velocity or attitude. However, if there are enough landmarks, the position and attitude can be completely recovered even without sensor filtering as, for example, in (Masselli & Zell, 2012) or (Huang & Netravali, 1994). Nevertheless, the method of Masselli & Zell used four landmarks, and Huang & Netravali indicate a minimum of three. As we will show, however, we can use fewer if we incorporate sensor filtering. In particular, the proposed method is capable of completely observing the vehicle position, velocity, and attitude with only two landmarks. The landmark navigation proposed in this thesis considers generic landmarks of unspecified type, to allow generalization to any type of landmarks, as long as they can be uniquely identified.

3. EKF Filtering

As shown in Figure 1, the proposed Extended Kalman Filter (EKF) receives data from two separate sources: first, the IMU serves as the input in the dynamic model, and second the landmark based measurement enters the EKF as a state update. The measurement can come in the form of a projection onto a monocular vision system, or as a full relative position vector to the detected target with both direction and distance information, as might be derived from stereo camera measurements, LIDAR, or other methods. Two EKF methods are developed—one for each case. The vision measurement is assumed to be corrupted by zero-mean Gaussian noise in the pixel frame. This measurement updates the state estimate obtained in the state propagation model from an integration of inertial measurements from the IMU (accelerometers, and gyroscope), which are also assumed to have zero-mean Gaussian noise.

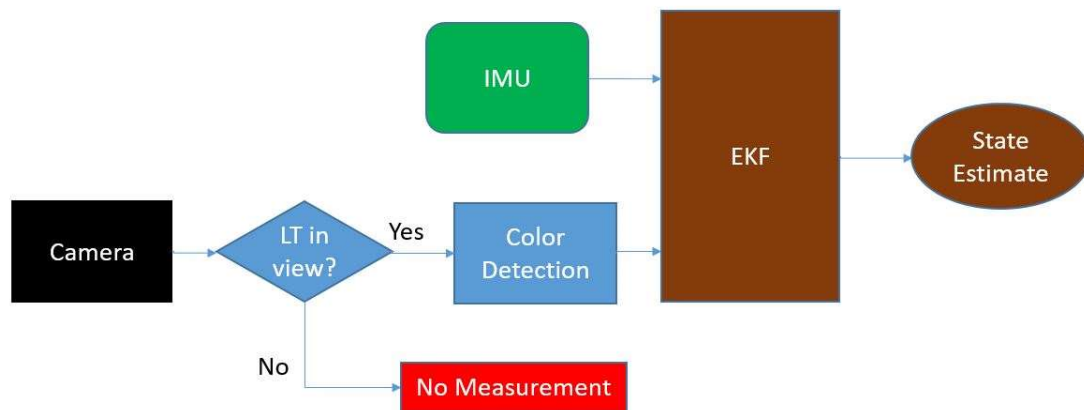


Figure 1: Navigation Algorithm Information Flow Diagram

As an initial note, we define the three separate reference frames considered by the filter. First, the Navigation Frame (N) is defined to represent the fixed world-coordinates

of the environment in which the vehicle operates. Second, the Body Frame (B) is defined, which has the X-axis pointed forward, the Y-axis out the right hand side, and the Z-axis downward with respect to the vehicle, with the origin fixed at the vehicle's center of gravity. Third, we consider the Camera Frame (C). The Camera Frame's orientation is defined by a rotation R_{BC} relative to the Body Frame, and its origin is located at a displacement \mathbf{d} from the center of gravity.

3.1. Overview

The filtering method takes the form of an Extended Kalman Filter, which is an extension of the classic linear Kalman filter (Kalman, 1960) to nonlinear problems via a first order linearization about the current estimate.

The discrete EKF algorithm, as outlined in (Ribeiro, 2004) as well as in numerous other sources, is performed in two steps: prediction and update. The prediction step is given in Equations (1) - (2), and the update step is given in Equations (3) - (5).

$$\hat{X}_k^- = f(\hat{X}_{k-1}, u_{k-1}) \quad (1)$$

$$P_k^- = A_{k-1} P_{k-1} A_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \quad (2)$$

$$K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \quad (3)$$

$$\hat{X}_k = \hat{X}_k^- + K_k (Y_k - h(\hat{X}_k^-)) \quad (4)$$

$$P_k = (I_n - K_k C_k) P_k^- \quad (5)$$

In the EKF equations given above, \hat{X}_k is the state estimate with autocovariance P_k , Y_k is the measurement, and the subscript k designates the current time step. The function f is the (generally nonlinear) discretized state-dynamics equation with forcing u_k , and the (also nonlinear) function h is the measurement model equation. Q_k and R_k are the covariance

matrices representing process noise, w_k , (associated with the dynamics f) and measurement noise (associated with h). The matrices A_{k-1}, G_{k-1}, C_k are the Jacobians representing the partial derivatives given by Equations (6) - (8):

$$A_{k-1} = \left. \frac{\partial f}{\partial X_{k-1}} \right|_{\hat{x}_{k-1}} \quad (6)$$

$$G_{k-1} = \left. \frac{\partial f}{\partial w_{k-1}} \right|_{\hat{x}_{k-1}} \quad (7)$$

$$C_k = \left. \frac{\partial h}{\partial X_k} \right|_{\hat{x}_k^-} \quad (8)$$

Lastly, the matrix K_k as defined in Equation (3) is the well-known Kalman gain, which is optimal in the least-squares sense if both f and h are linear functions. Even if nonlinearities are present, however, the extended Kalman filter is often still effective in practice.

We note for clarity that in some conventions A_{k-1}, C_k are denoted as F_{k-1}, H_k respectively as in the Kalman review given by Hartikainen et al. (2011) for example. The notation A_{k-1}, C_k is used throughout this thesis.

Because the algorithm is separated into two steps, we have the option of performing a prediction step without an update if there is no measurement available. This is usually the case in practice, since IMUs are considerably faster in terms of sampling rate than vision measurements. Additionally, in our landmark method it is possible that measurements may be unavailable for extended periods of time. Chapter 3.4 deals with the instabilities associated with this possibility.

3.2. Process Model

The process model state vector X is defined as a column vector containing the vehicle's position and velocity in the navigation frame, P_N and V_N , the three Euler angles representing the vehicle's orientation, $\rho = \{\phi, \theta, \psi\}$, and the estimated biases on the accelerometers and gyros comprising the IMU, b_{acc} and b_{gyr} . The resulting state vector X is given in Equation (9) below.

$$X = [P_N, V_N, \rho, b_{acc}, b_{gyr}]^T \quad (9)$$

This simultaneous usage of the EKF as both a state estimator and a parameter ID scheme, in our case for the bias, is a well-known one outlined and analyzed for stability by Ljung (1979) which appears to have been utilized in some form as early as 1964 by Cox (Cox, 1964).

The process dynamic model is a basic first-order forward-difference Euler integration. The process model takes the form of Equation (10):

$$X_k = f(X_{k-1}, u_{k-1}, w_{k-1}) \quad (10)$$

where k represents the current time step, the forcing u is the IMU measurement which includes the accelerometers and gyros, and w represents the process noise associated with these (assumed Gaussian with zero mean). We recall that Equation (10) is the general form for any dynamic model mapping the state at the previous time step into the current time step. For our Euler integration model, Equation (10) takes on the form given in Equation (11):

$$\begin{bmatrix} P_{N,k} \\ V_{N,k} \\ \rho_k \\ b_{acc,k} \\ b_{gyr,k} \end{bmatrix} = \begin{bmatrix} P_{N,k-1} + V_{N,k-1}\Delta t_s \\ V_{N,k-1} + (R_{B,k-1}^N a_{k-1}^B - \mathbf{g})\Delta t_s \\ \rho_{k-1} + (L_{B,k-1}^N \omega_{k-1}^B)\Delta t_s \\ b_{acc,k-1} \\ b_{gyr,k-1} \end{bmatrix} \quad (11)$$

Here, R_B^N is the rotation matrix derived from the Euler angles ρ , which is given by the well-known direction cosine matrix:

$$R_B^N = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\psi \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}^T \quad (12)$$

where the shorthand notation s, c denotes \sin, \cos . In Equation (11), a^B and ω^B are the accelerometer and gyro measurements respectively, \mathbf{g} is the gravitation vector, and Δt_s is the time step. We note that the time step Δt_s is set equal to the sampling period of the IMU for practical purposes. Lastly, L_B^N is the well-known Euler angle dynamic matrix relating the gyro measurements in the body frame of roll rate p , pitch rate q , and yaw rate r , to the derivatives of the Euler angles ϕ, θ, ψ representing roll, pitch, and yaw respectively in the navigation frame:

$$L_B^N = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{bmatrix} \quad (13)$$

so that:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi) \tan(\theta) & \cos(\phi) \tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) \sec(\theta) & \cos(\phi) \sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (14)$$

or

$$\dot{\rho} = L_B^N \omega^B \quad (15)$$

Lastly, the IMU measurements are known to be corrupted by both random noise w_{acc} and w_{gyr} (assumed Gaussian) and biases b_{acc} and b_{gyr} (assumed constant). Thus, the measured acceleration and rotational velocity differ from the true values a^B and ω^B as seen in Equation (16):

$$\begin{aligned} a_{acc}^B &= a^B - b_{acc} - w_{acc} \\ \omega_{gyr}^B &= \omega^B - b_{gyr} - w_{gyr} \end{aligned} \quad (16)$$

As in (Zhao, Lin, Peng, Chen, & Lee, 2012), a correction of the IMU measurements can be made using the estimated values of b_{acc} and b_{gyr} from the state vector estimate \hat{X} .

The EFK process Jacobian A is thus given by:

$$A_{k-1} = \left. \frac{\partial f}{\partial X_{k-1}} \right|_{\hat{X}_{k-1}} = \begin{bmatrix} I_3 & I_3 \Delta t_s & [0] & [0] & [0] \\ [0] & I_3 & \frac{\partial V_{Ek}}{\partial \rho_{k-1}} & -R_B^N \Delta t_s & [0] \\ [0] & [0] & \frac{\partial \rho_k}{\partial \rho_{k-1}} & [0] & -L_B^N \Delta t_s \\ [0] & [0] & [0] & I_3 & [0] \\ [0] & [0] & [0] & [0] & I_3 \end{bmatrix} \quad (17)$$

where $[0]$ is the 3x3 zero matrix, and I_3 is the 3x3 identity matrix. We note that there are ways to deal with the bias separately, for example as in (Friedland, 1969), in order to reduce the size of the matrix A ; however, a 15×15 matrix is not particularly unwieldy for modern computers, and such methods are unnecessary.

3.3. Landmark-Based Measurement Models

The measurement Y_i for the i^{th} landmark can be obtained via an algorithm such as the color-detection methods used in this thesis, or a more sophisticated learning algorithm like the Viola-Jones object detection algorithm (Viola & Jones, 2001) (Prazenica, et al., 2016). In any case, measurements are first obtained in terms of the pixel location of an identified landmark in an image and must be converted into calibrated coordinates (Ma, Kosecka, Soatto, & Sastry, 2001). Then, if stereo measurements are available, a full vector representing the position of the landmark in the camera frame can be generated. If not, then only the raw image-calibrated coordinates will be passed to the filter, which effectively provides the unit vector from the camera frame to the landmark. We first consider the case of monocular vision with no stereo capability.

3.3.1. Monocular Measurement

The raw measurement of the landmark from the camera comes in the form:

$$\bar{Y}_{pixels} = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (18)$$

Here, x_p and y_p are the x and y coordinates of the detected landmark in pixels measured from the top-left corner of the image. The bar notation simply indicates homogeneous form. To convert this into calibrated coordinates, we use the camera intrinsic matrix:

$$\bar{Y} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = M \bar{Y}_{pixels} = \begin{bmatrix} 1 & 0 & -c_x \\ f_x & 0 & -f_x \\ 0 & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \quad (19)$$

In Eq. (19), M is the camera intrinsic matrix, $[f_x, f_y]$ are the x and y focal lengths of the camera in pixels (usually the same), and $[c_x, c_y]$ are the x and y coordinates of the camera's

principal point in pixels. Also, while Eq. (19) is applied to every detected landmark individually, the i subscripting has been omitted for simplicity. Once each measurement has been calibrated, the total measurement Y , corresponding to n landmarks, can be passed to the filter as a stacked vector:

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \quad (20)$$

We note that the measurement must be accompanied by a list vector indicating which landmarks were detected, and the order that they appear in Y . This is critical for the EKF to generate an appropriate \hat{Y} for the update step. As a result, the length of the vectors Y and \hat{Y} changes over time as landmarks move in and out of view. For example, at a given time step, the measurement may take the form:

$$Y = \begin{bmatrix} Y_1 \\ Y_3 \\ Y_7 \end{bmatrix} \quad (21)$$

where only the first, third, and seventh landmarks on the map are observed. In this case, the list vector would be:

$$l = \begin{bmatrix} 1 \\ 3 \\ 7 \end{bmatrix} \quad (22)$$

From this list, the measurement estimation vector \hat{Y} can be constructed to match the form of the actual measurement Y .

The estimated measurement \hat{Y} is modeled by:

$$\hat{Y} = \begin{bmatrix} \hat{Y}_1 \\ \vdots \\ \hat{Y}_n \end{bmatrix} \quad (23)$$

where the individual elements \hat{Y}_i are given by the anticipated locations of the identified

landmarks based on the current state estimate:

$$\hat{Y}_i = \frac{1}{z_i} \Pi_0 \mathbf{x}_i = \frac{1}{z_i} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_i/z_i \\ y_i/z_i \end{bmatrix} \quad (24)$$

Here, the vector $\mathbf{x}_i = [x_i \ y_i \ z_i]^T$ is the position of the i^{th} landmark in the camera's reference frame and Π_0 is the projection matrix. Thus, the measurement element \hat{Y}_i is the landmark's position projected into the image plane of an ideal pinhole camera, and so carries information only about the direction to the landmark, but not the distance to the landmark.

As noted, \mathbf{x}_i is the position of a landmark in the camera frame. Thus,

$$\bar{\mathbf{x}}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = G \bar{\mathbf{X}}_i = \begin{bmatrix} R & -T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (25)$$

where $\mathbf{X}_i = [X_i \ Y_i \ Z_i]^T$ is the known a priori position of the i -th landmark in the navigation frame and G represents the relative pose of the vehicle in the navigation frame comprised of the rotation matrix R and translation vector T . The rotation matrix R is formed from the Euler angles representing the vehicle's attitude: $R = R(\rho) = R_N^B$, and T is simply the vehicle position: $T = P_N$. Therefore, the total model for the measurement \hat{Y}_i is given by Eq. (26), which represents the ideal pinhole camera model.

$$\hat{Y}_i = \frac{1}{z_i} \Pi_0 \hat{G} \bar{\mathbf{X}}_i \quad (26)$$

Finally, the measurement Jacobian, C , is given by:

$$C = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \quad (27)$$

where:

$$C_i = \frac{\partial h_i}{\partial X_k} \Big|_{X_k} = \begin{bmatrix} -1/z_i & 0 & x_i/z_i^2 \\ 0 & -1/z_i & y_i/z_i^2 \end{bmatrix} \begin{bmatrix} R_N^C & [0] & -R_B^C \left[\frac{\partial R_N^B}{\partial \phi} x_i^N & \frac{\partial R_N^B}{\partial \theta} x_i^N & \frac{\partial R_N^B}{\partial \psi} x_i^N \right] & [0] & [0] \end{bmatrix} \quad (28)$$

Here, $R_N^C = R_B^C R_N^B$ is the rotation matrix from the navigation frame to the camera frame, and x_i^N is a vector pointing from the vehicle CG to the i-th landmark in the inertial reference frame.

3.3.2. Stereo Measurement

To adapt the filter to make use of depth perception, if it is available, the state estimation vector and process are defined as before in Eqs. (9)-(16). The measurement model is all that requires modification. First we redefine the measurement:

$$Y_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (29)$$

Thus, the full relative position vector (in the camera frame) from the camera to the landmark is measured, instead of only a unit directional vector to the target. This is easily accomplished using basic parallax and the known relative pose between the two cameras, or alternately through knowledge of a landmark's size. The change of measurement vector consequently changes, and indeed greatly simplifies, the Jacobian matrix C_i :

$$C_i = \left. \frac{\partial h_i}{\partial X_k} \right|_{\hat{x}_k^-} = \begin{bmatrix} -R_N^C & [0] & R_B^C \begin{bmatrix} \frac{\partial R_N^B}{\partial \phi} x_i^N & \frac{\partial R_N^B}{\partial \theta} x_i^N & \frac{\partial R_N^B}{\partial \psi} x_i^N \end{bmatrix} & [0] & [0] \end{bmatrix} \quad (30)$$

The total Jacobian matrix C is then constructed as in Eq. (27).

3.4. Underweighting for Nonlinear Instability

A known shortcoming of the EKF is the fact that it represents a linearization about the current state estimate of an otherwise nonlinear system. Thus, there is no guarantee of stability for the EKF algorithm like there is for the linear Kalman Filter. In the context of our landmark filter, the problems associated with the linearization manifest themselves in a tendency for the filter to over/under-correct and generally become unstable as landmarks pass in and out of view. More specifically, when no measurements (i.e. no landmarks) are detected for an extended period of time, the EKF resorts to a simple integration of the IMU to propagate the state estimate forward in time. This leads to the inevitable ‘drift’ problem described previously in Chapter 2. Upon reacquisition of a landmark measurement, the state must converge back toward the ground truth after having strayed. The stability of this re-convergence is often unsatisfactory, as illustrated in Figure 2, which gives the X-position estimate from a simulation in which a cluster of four landmarks was repeatedly lost and then reacquired.

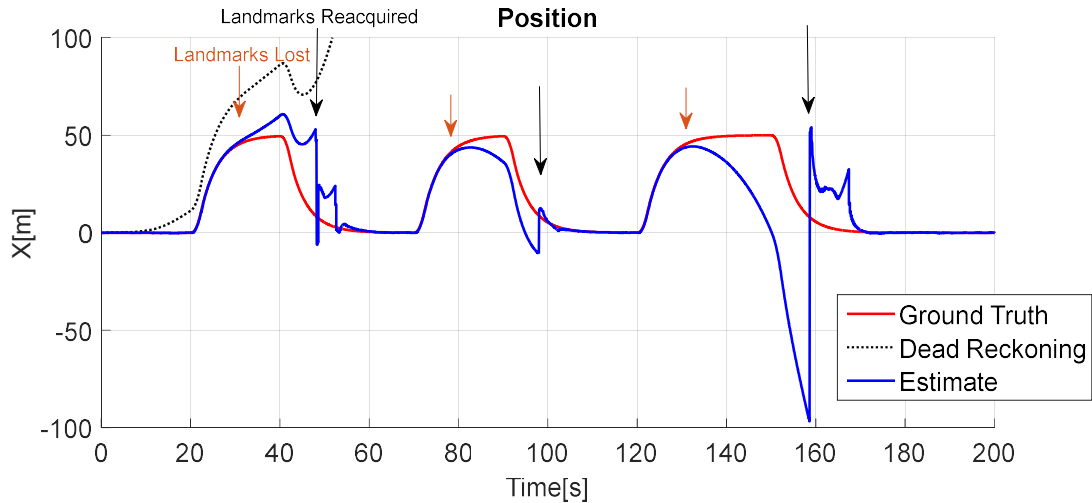


Figure 2: Nonlinear Re-convergence Instability

There are two approaches to solving this stability problem. The first is, quite simply, to acknowledge the nonlinear nature of the problem by using a nonlinear filter. A second-order-EKF, or an Unscented Kalman Filter (UKF) (Julier & Uhlmann, 1997) would likely serve the purpose. However, while the linear nature of the traditional EKF presents an unfortunate weakness, it is also the source of the EKF's great simplicity, efficiency, and near ubiquity as compared with the nonlinear filters mentioned above; therefore we would like to continue using the EKF algorithm. This leads us to the idea of 'underweighting' as developed by Lear (1973) for the Space Shuttle program, which simultaneously resolves the particular instability problem of re-convergence following a long period of drift, while retaining the linear nature (and corresponding efficiency) of the EKF.

The source of the unstable convergence in question is illustrated in Figure 3, which shows the X-position estimation error and its covariance immediately after reacquisition of a cluster of landmarks in the simulation from Figure 2. The problem is that the error covariance converges toward zero faster than the actual error. In fact, once the covariance is sufficiently close to zero, the actual error stops decreasing, since measurement updates

are now being ignored due to the EKF's incorrect perception that the estimation is accurate. The reason for this mis-match of convergence rates is that the covariance update assumes the convergence that would be expected from a linear system, while the actual estimation does not converge this quickly due to nonlinearity. To mitigate this issue, Lear's underweighting approach reduces the rate of convergence of the covariance, preventing it from 'outrunning' the actual estimation.



Figure 3: Overly Optimistic Covariance Convergence

Mathematically, the source of the overly optimistic covariance is in the denominator of the well-known Kalman gain:

$$K_k = P_k^- C_k (C_k P_k^- C_k^T + R_k)^{-1} \quad (31)$$

As mentioned before, the Kalman gain is only optimal for the linear case. As in (Zanetti, DeMars, & Bishop, 2010), which gives an excellent overview of the underweighting topic, a second-order EKF formulation indicates that the denominator of the gain should be given by:

$$C_k P_k^- C_k^T + R_k + B_k \quad (32)$$

where B_k represents the second-order effects on the error covariance. As an extension,

we can think of all the higher-order nonlinear effects as being represented by the matrix B_k . We also note that all the matrices in Equation (32) are positive definite; thus, the denominator of the Kalman gain is too small, and we augment it with a positive definite matrix U_k , intended as a substitute for B_k .

$$K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k + U_k)^{-1} \quad (33)$$

As noted in (Zanetti, DeMars, & Bishop, 2010), there are multiple ways to choose the underweighting matrix U_k ; however, we will restrict ourselves to a discussion of the method used by Lear, as it is the method used in this thesis. Namely, we choose U_k as a constant fraction of the state covariance term in the Kalman denominator: $U_k \equiv \beta(C_k P_k^- C_k^T)$, with $\beta = 0.2$. As a result of this choice, U_k will be large when the estimation covariance, P_k^- , is large (i.e. after long periods of drift), and will be less consequential during cases when estimation error is small. Additionally, to maintain standard EKF behavior under normal (small error) conditions, we can specify a cutoff value, σ , on the trace of P_k^- such that the underweighting factor, β , is set identically to zero when the estimation error is small.

$$\beta = \begin{cases} 0.2 & \text{if } 3\sqrt{\text{tr}(P_k^-)} \geq \sigma \\ 0 & \text{otherwise} \end{cases}$$

In our case, $\sigma = 5$ meters, an arbitrarily chosen value. A substitution of our choice of U_k into the Kalman gain denominator gives us the underweighted gain:

$$K_k = P_k^- C_k^T \left((1 + \beta) C_k P_k^- C_k^T + R_k \right)^{-1} \quad (34)$$

Figure 4 gives the same scenario as in Figure 2 and Figure 3 with underweighting applied, illustrating the improved stability during convergence.

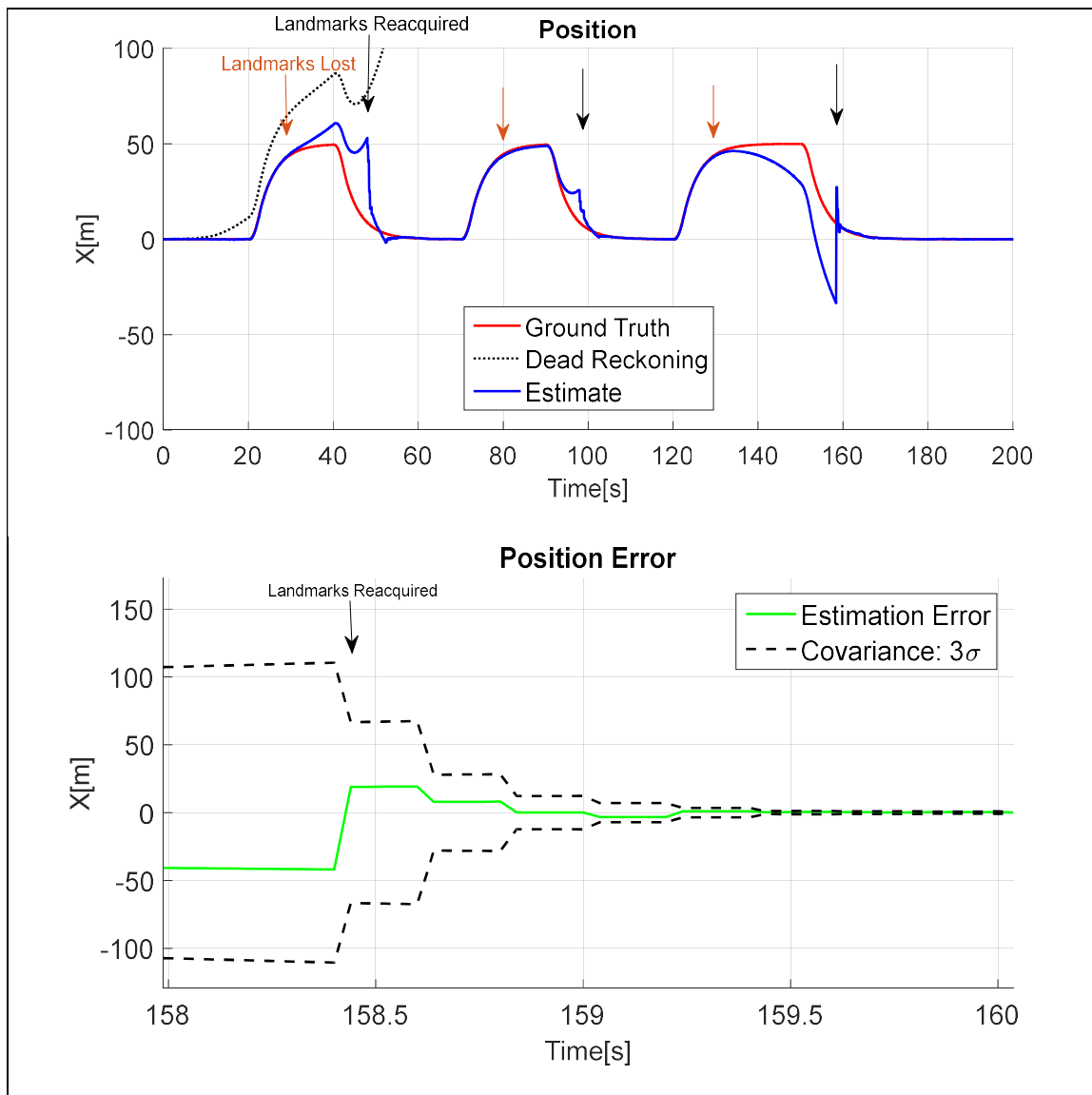


Figure 4: Underweighted Re-convergence

3.5. Observability Analysis

Observability is a key feature in any state estimation endeavor. We thus give an analysis of the observability of our system with consideration for the stereo vs monocular distinction, and the effects of varying the number of landmarks.

For a linear system to be completely observable, we have the well-known result that the observability matrix given by:

$$P_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (35)$$

must be full rank; that is we must have $rank(P_o) = n$ where $n = 15$ is the size of the state vector. We recall that the Jacobians A, C are given by Equations (17) and (28). We note that Equation (35) holds for both the discrete and continuous time cases. In our case, we have a discrete time system. Nevertheless, to simplify the analysis, it will help considerably to perform our derivations using a continuous-time model, eliminating the discrete nature of the problem. This requires us to redefine A . In a discrete-time system, the dynamics take on the following form:

$$X_{k+1} = A_{discrete}X_k = (I + A\Delta t)X_k \quad (36)$$

we can rearrange Equation (36) to obtain:

$$\dot{X}_k \cong \frac{X_{k+1} - X_k}{\Delta t} = AX_k \quad (37)$$

Thus, we have redefined A such that $A = \frac{A_{discrete} - I}{\Delta t}$ where $A_{discrete}$ is the Jacobian given in Equation (17). This redefinition results in an A matrix of the form:

$$A = \begin{bmatrix} [0] & I_3 & [0] & [0] & [0] \\ [0] & [0] & V_\rho & -R_B^N & [0] \\ [0] & [0] & \rho_\rho & [0] & -L_B^N \\ [0] & [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] & [0] \end{bmatrix} \quad (38)$$

Additionally, we will simplify our representation of C to the more compact notation:

$$C = [Y_p \quad [0] \quad Y_\rho \quad [0] \quad [0]] \quad (39)$$

where Y_p and Y_ρ are defined as in Equations (28) and (30).

3.5.1. Monocular Single-Landmark Unobservability

We begin by constructing the observability matrix for the case of a single landmark via a substitution of Equations (38) and (39) into Equation (35):

$$P_o = \begin{bmatrix} Y_p & [0] & Y_\rho & [0] & [0] \\ [0] & Y_p & Y_\rho \rho_\rho & [0] & -Y_\rho L_B^N \\ [0] & [0] & Y_p V_\rho + Y_\rho \rho_\rho^2 & -Y_p R_B^N & -Y_\rho \rho_\rho L_B^N \\ [0] & [0] & Y_p V_\rho \rho_\rho + Y_\rho \rho_\rho^3 & [0] & -(Y_p V_\rho + Y_\rho \rho_\rho^2) L_B^N \\ [0] & [0] & Y_p V_\rho \rho_\rho^2 + Y_\rho \rho_\rho^4 & [0] & -(Y_p V_\rho \rho_\rho + Y_\rho \rho_\rho^3) L_B^N \\ & & & \vdots & \end{bmatrix} \quad (40)$$

In principle, we would now search for right-singular vectors of P_o , which give the unobservable directions in the state space. That is, in general we seek some unobservable vector $\mathbf{u} \neq 0$ such that $P_o \mathbf{u} = 0$. However, Equation (40) is a rather unwieldy matrix to examine in this way. Therefore, we make the simplifying assumptions that the vehicle is facing ‘North’, and in a hover. That is, we assume $V_N, \rho, \omega^B, a^B \rightarrow 0$. This results in the following simplifications:

$$\rho_\rho \rightarrow 0 \quad (41)$$

$$V_\rho \rightarrow -g\Gamma ; \Gamma \equiv \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (42)$$

$$L_B^N, R_B^N \rightarrow I_3 \quad (43)$$

where g is the gravitation constant. This results in Equation (40) simplifying to:

$$P_o = \begin{bmatrix} Y_P & [0] & Y_\rho & [0] & [0] \\ [0] & Y_P & [0] & [0] & -Y_\rho \\ [0] & [0] & -gY_P\Gamma & -Y_P & [0] \\ [0] & [0] & [0] & [0] & gY_P\Gamma \\ [0] & [0] & [0] & [0] & [0] \\ \vdots & & & & \end{bmatrix} \quad (44)$$

In this case, we can see that there will be unobservable state space directions, since each block in P_o has dimensions 2×3 , implying that the maximum possible rank of P_o is 8, since there are only 8 non-zero rows. We note that even if the vehicle is not in a hover or constant translation, the reductions in Equations (41)-(43) are still approximately true since g tends to dominate in the body accelerations ($g \gg \|a^B - \mathbf{g}\|$), and both attitude, ρ , and rotation rates, ω^B tend to be small. Thus, even if the rank of P_o as given in Equation (40) is greater than 8, it is likely to be only marginally so. That is, the non-zero singular values associated with the additional ‘observable’ right-singular vectors will be significantly small.

Now that P_o has been considerably simplified, we can turn our attention to deriving some of the unobservable state space directions. While there are at least $n - 8 = 7$ unobservable directions, we give the derivations for just two of these that are of particular interest. The remaining directions have to do with the biases and their couplings with the other states.

First, we turn our attention to the upper-left block in Equation (44). We recall from Equation (28) that:

$$Y_P = \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_N^C \quad (45)$$

Even this block alone must have at least one unobservable 3D position direction associated with it since it is 2×3 . Thus, we seek a vector $\mathbf{u} \neq 0$ such that $Y_P \mathbf{u} = 0$:

$$Y_P \mathbf{u} = \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_N^C \mathbf{u} = \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} \mathbf{u}^C = 0 \quad (46)$$

where \mathbf{u}^C is our vector expressed in the camera frame. Thus, we have a system of two equations for \mathbf{u}^C :

$$\begin{aligned} -\frac{1}{z} u_1^C + \frac{x}{z^2} u_3^C &= 0 \\ -\frac{1}{z} u_2^C + \frac{y}{z^2} u_3^C &= 0 \end{aligned} \quad (47)$$

Simplifying gives:

$$\left. \begin{aligned} u_1^C &= \frac{x}{z} u_3^C \\ u_2^C &= \frac{y}{z} u_3^C \end{aligned} \right\} \Rightarrow \mathbf{u}^C = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} u_3^C \quad (48)$$

where u_3^C is an arbitrary scaling parameter. The result in Equation (48) is readily recognizable as giving the direction to the landmark, which has position $\mathbf{x} = [x \ y \ z]^T$ as noted in Chapter 3.3.1. Thus, the direction in 3D space toward/from the landmark is unobservable, which is unsurprising given that we are considering a monocular measurement. We also note that the (2,2) block in P_o is identical to the block just considered. This implies that velocity in the 3D spatial direction toward/from the landmark is also unobservable.

We now turn our attention to the observability of attitude by first examining the (1,3) block in P_o . We recall again from Equation (28) that:

$$Y_\rho = - \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_B^C \begin{bmatrix} \frac{\partial R_N^B}{\partial \phi} \mathbf{x}^N & \frac{\partial R_N^B}{\partial \theta} \mathbf{x}^N & \frac{\partial R_N^B}{\partial \psi} \mathbf{x}^N \end{bmatrix} \quad (49)$$

which simplifies to:

$$Y_\rho = - \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_B^C \frac{\partial \mathbf{x}^B}{\partial \rho} \quad (50)$$

recalling our result for the previous unobservable direction, we conclude that we are seeking a vector $\Delta\rho \neq 0$ such that:

$$\frac{\partial \mathbf{x}^B}{\partial \rho} \Delta\rho = \lambda \mathbf{x}^B \quad (51)$$

where λ is an arbitrary scale factor. If this is so, then we will have $R_B^C \frac{\partial \mathbf{x}^B}{\partial \rho} \Delta\rho$ in the direction of the landmark in the camera frame, and as before, we will have $Y_\rho \Delta\rho = 0$ as desired.

However, $\frac{\partial \mathbf{x}^B}{\partial \rho}$ expresses the ability of small changes in attitude, ρ , to produce small changes in the direction to the landmark in the body frame, \mathbf{x}^B . This, of course, is why we chose to name our unobservable attitude-direction vector $\Delta\rho$. Thus, Equation (51) indicates a small change in attitude producing a change in \mathbf{x}^B along its own axis; however, this is not geometrically possible since attitude changes represent a unitary transformation. Therefore, we must have $\lambda = 0$, which implies that $\frac{\partial \mathbf{x}^B}{\partial \rho}$ is itself a singular matrix, and we must find an attitude change $\Delta\rho$ such that \mathbf{x}^B is unaltered. This, of course, indicates that $\Delta\rho$ represents a rotation about the axis defined by \mathbf{x}^B —the direction to the landmark, which is unsurprising given the nature of vision-based perception and the symmetric nature of a point-defined landmark. Thus, rotation about the 3D axis that points in the direction to the landmark would be expected to be unobservable.

However, such a judgement would be premature, because the (1,3) block of P_o is not the only non-zero block in its column. The block: $-gY_p\Gamma$ sits at the (3,3) position as seen in Equation (44). Thus, the gravitation provides some help in the observability of the

attitude. A complete analysis of attitude unobservability must consider the block matrix given by:

$$Y_{P\rho} = \begin{bmatrix} Y_P & Y_\rho \\ 0 & -gY_P\Gamma \end{bmatrix} \quad (52)$$

That is, we seek a vector $\mathbf{u} = \begin{bmatrix} \mathbf{u}_P \\ \mathbf{u}_\rho \end{bmatrix} \neq 0$ such that $Y_{P\rho}\mathbf{u} = 0$, or

$$\begin{cases} Y_P\mathbf{u}_P + Y_\rho\mathbf{u}_\rho = 0 \\ -gY_P\Gamma\mathbf{u}_\rho = 0 \end{cases} \quad (53)$$

We consider first the second equation in (53), which expands after canceling a factor of $-g$ to become:

$$\begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_N^C \Gamma \mathbf{u}_\rho = 0 \quad (54)$$

Thus, as before we need $R_N^C \Gamma \mathbf{u}_\rho$ in the direction of the landmark in the camera frame as given by: $R_N^C \Gamma \mathbf{u}_\rho = \lambda \mathbf{x}^C$ where λ is an arbitrary scale factor. Therefore, $\Gamma \mathbf{u}_\rho = \lambda \mathbf{x}^N$. In general, the Z-coordinate of \mathbf{x}^N is not zero; however, the Z-coordinate of $\Gamma \mathbf{u}_\rho$ is always zero, since the bottom row of Γ contains only zeros. Thus, we must have $\lambda = 0$. This implies that: $\Gamma \mathbf{u}_\rho = 0 \Rightarrow \mathbf{u}_\rho = [0 \quad 0 \quad \Delta\psi]^T$ since the last row of Γ is also all zeros.

We now substitute our result for \mathbf{u}_ρ into the first equation in (53) and expand:

$$\begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_N^C \mathbf{u}_P - \begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_B^C \left[\frac{\partial R_N^B}{\partial \phi} \mathbf{x}^N \quad \frac{\partial R_N^B}{\partial \theta} \mathbf{x}^N \quad \frac{\partial R_N^B}{\partial \psi} \mathbf{x}^N \right] \begin{bmatrix} 0 \\ 0 \\ \Delta\psi \end{bmatrix} = 0 \quad (55)$$

Simplifying, we get:

$$\begin{bmatrix} -1/z & 0 & x/z^2 \\ 0 & -1/z & y/z^2 \end{bmatrix} R_B^C \left(R_N^B \mathbf{u}_P - \frac{\partial \mathbf{x}^B}{\partial \psi} \Delta\psi \right) = 0 \quad (56)$$

Thus, we must have, as before:

$$\left(R_N^B \mathbf{u}_P - \frac{\partial \mathbf{x}^B}{\partial \psi} \Delta\psi \right) = \lambda \mathbf{x}^B \quad (57)$$

Solving for \mathbf{u}_P gives the final result:

$$\mathbf{u}_P = \lambda \mathbf{x}^N + \Delta\psi \frac{\partial \mathbf{x}^N}{\partial \psi} \quad (58)$$

where λ and $\Delta\psi$ are arbitrary scaling factors with \mathbf{x}^N and $\frac{\partial \mathbf{x}^N}{\partial \psi}$ providing the unobservable position directions. The $\lambda \mathbf{x}^N$ term is just our first unobservable mode toward/from the target all over again; however, the second term, $\Delta\psi \frac{\partial \mathbf{x}^N}{\partial \psi}$, tells us that the position is unobservable in a direction defined by the change in \mathbf{x}^N induced by a change in heading. This direction will necessarily be both perpendicular to \mathbf{x}^N , since attitude changes are unitary, and in the X-Y plane of the navigation frame, since we are restricted to yawing motion. Additionally, our ‘arbitrary’ scaling factor, $\Delta\psi$, represents an actual yaw motion, so that the position unobservability in this direction is coupled with the yaw. Physically, this represents the unobservability of the azimuthal location of the vehicle relative to the landmark, which manifests itself in a tendency of the state estimate to drift in a circular pattern in the X-Y plane around the landmark, while always facing the landmark.

We remark that both unobservability ‘directions’ in state space derived here are corroborated by results from simulation testing, as shown in Chapter 4.2.

3.5.2. Additional Landmarks

Next, we consider the case of two landmarks. When two landmarks are present, the observability matrix in Equation (44) is augmented to account for the new landmark:

$$P_o = \begin{bmatrix} P_{o,1} \\ P_{o,2} \end{bmatrix} = \begin{bmatrix} Y_{P,1} & [0] & Y_{\rho,1} & [0] & [0] \\ [0] & Y_{P,1} & [0] & [0] & -Y_{\rho,1} \\ [0] & [0] & -gY_{P,1}\Gamma & -Y_{P,1} & [0] \\ [0] & [0] & [0] & [0] & gY_{P,1}\Gamma \\ [0] & [0] & [0] & [0] & [0] \\ & & & \vdots & \\ Y_{P,2} & [0] & Y_{\rho,2} & [0] & [0] \\ [0] & Y_{P,2} & [0] & [0] & -Y_{\rho,2} \\ [0] & [0] & -gY_{P,2}\Gamma & -Y_{P,2} & [0] \\ [0] & [0] & [0] & [0] & gY_{P,2}\Gamma \\ [0] & [0] & [0] & [0] & [0] \\ & & & \vdots & \end{bmatrix} \quad (59)$$

We now have 16 non-zero rows which is already more than the 15 required for potential full observability. Thus, two or more landmarks give the possibility for a completely observable system, even with the restrictions of monocular vision. As for our previously derived unobservable directions, they may remain unobservable if the landmark positions are in certain degenerate configurations. First, if the landmark positions and the position of the vehicle are all co-linear, in which case, $Y_{P,1}$ and $Y_{P,2}$ differ only by a constant scalar and are thus linearly dependent, the first unobservable direction toward/from the targets will persist. Also, if the second landmark is positioned directly above/below the first in the navigation frame, the second derived unobservable direction will remain since $\frac{\partial x_1^N}{\partial \psi}$ and $\frac{\partial x_2^N}{\partial \psi}$ are identical.

3.5.3. Stereo

We now briefly consider the effect that a stereo measurement can have on the observability of our system, and of the two derived unobservability modes from the previous section. We return to our assumption of a single landmark.

When a stereo measurement of a single landmark is present, the observability matrix

in Equation (44) does not change its form; however, Y_p and Y_ρ now have dimension 3×3 . Thus, there are now 12 non-zero rows. This is not enough to provide complete observability, but it is a significant improvement on the 8 rows in the monocular case.

In particular, the previously unobservable direction toward/from the landmark is eliminated since we now have $Y_p = -R_N^C$ from Equation (30), which has full rank since R_N^C is unitary. This is unsurprising given that the distance to the landmark is now being directly measured.

The azimuthal position, however, remains unobservable since the stereo measurement provides no new information in this regard. Mathematically, this is because Γ remains unchanged when the stereo measurement is introduced, allowing for the unobservable $\Delta\psi$ to remain. Thus, we obtain:

$$-R_N^C \mathbf{u}_p + R_B^C \begin{bmatrix} \frac{\partial R_N^B}{\partial \phi} \mathbf{x}^N & \frac{\partial R_N^B}{\partial \theta} \mathbf{x}^N & \frac{\partial R_N^B}{\partial \psi} \mathbf{x}^N \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \Delta\psi \end{bmatrix} = 0 \quad (60)$$

from the first equation in (53), which reduces to an equivalent result:

$$R_N^C \mathbf{u}_p = R_B^C \frac{\partial R_N^B}{\partial \psi} \mathbf{x}^N \Delta\psi \Rightarrow \mathbf{u}_p = \Delta\psi \frac{\partial \mathbf{x}^N}{\partial \psi} \quad (61)$$

Note that, when stereo measurements of more than one landmark are available in a general configuration, the system is fully observable.

3.5.4. Nonlinear Effects

To this point, we have ignored the possible effects of nonlinearities in the measurement model. Returning to the monocular case with one landmark, we recall that two unobservable directions in the state space were derived. One toward the landmark, and one in an azimuthal circle around it. However, this picture does not tell the whole

story, since it ignores nonlinearity. In particular, the position in the direction toward the landmark (i.e. distance from the landmark) can, surprisingly, be observed provided that the vehicle is in motion. Conceptually, this is because of the parallax between frames at adjacent time steps as we shall see.

To show this result, we must consider a discrete-time analysis of observability, which requires use of the original process Jacobian A as given below in Equation (62):

$$A \equiv \frac{\partial f}{\partial X} = \begin{bmatrix} I_3 & I_3 \Delta t_s & [0] & [0] & [0] \\ [0] & I_3 & V_\rho & -R_B^N \Delta t_s & [0] \\ [0] & [0] & \rho_\rho & [0] & -L_B^N \Delta t_s \\ [0] & [0] & [0] & I_3 & [0] \\ [0] & [0] & [0] & [0] & I_3 \end{bmatrix} \quad (62)$$

where f is defined by the discretized dynamics as before:

$$X_k = f(X_{k-1}) \quad (63)$$

With the Jacobian A redefined, we can now re-evaluate the observability matrix:

$$P_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} = \begin{bmatrix} Y_P & [0] & Y_\rho & [0] & [0] \\ Y_P & Y_P \Delta t_s & Y_\rho \rho_\rho & [0] & -Y_\rho L_B^N \Delta t_s \\ Y_P & 2Y_P \Delta t_s & Y_P V_\rho \Delta t_s + Y_\rho \rho_\rho^2 & -Y_P R_B^N \Delta t_s & \dots \\ Y_P & 3Y_P \Delta t_s & \dots & -3Y_P R_B^N \Delta t_s & \dots \\ Y_P & 4Y_P \Delta t_s & \dots & -6Y_P R_B^N \Delta t_s & \dots \\ \vdots & & & & \end{bmatrix} \quad (64)$$

We recall from Chapter 3.5.1 that landmark-directed unobservability comes from the 2×3 block matrix Y_P in the measurement Jacobian, C , which appeared in the upper right-hand entry in the continuous-time observability matrix and had max rank of 2. Here, we see it repeated throughout the first column. We see a similar situation for the second column, which gave us the unobservability of velocity.

At first glance, this repetition of Y_P seems irrelevant, since Y_P is certainly linearly dependent with itself, and successive multiples in the second column for the velocity do

not change the verdict either. However, since we are dealing with a discrete-time analysis, the rows of the observability matrix represent measurements at a series of n time steps. Thus, since there is a linearization performed by the EKF at each time step, it is important that the Jacobians be evaluated appropriately, such that the observability matrix should be more precisely defined as:

$$P_o = \begin{bmatrix} C_k \\ C_{k+1}A_{k+1} \\ C_{k+2}A_{k+1}A_{k+2} \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} Y_p|_k & [0] & & & \\ Y_p|_{k+1} & Y_p|_{k+1}\Delta t_s & & & \\ Y_p|_{k+2} & 2Y_p|_{k+2}\Delta t_s & \cdots & \cdots & \cdots \\ Y_p|_{k+3} & 3Y_p|_{k+3}\Delta t_s & & & \\ Y_p|_{k+4} & 4Y_p|_{k+4}\Delta t_s & & & \\ & \vdots & & & \end{bmatrix} \quad (65)$$

Equation (65) thus makes clear that the first and second block-columns can be full rank, due to the EKF's linearizations about the current state estimate at each time step. It is entirely possible that the successive re-evaluations of Y_p may introduce the final linearly independent vector to the columns. However, in order for this to occur, we must have a component of motion perpendicular to the landmark direction (previously unobservable), since any other motion simply yields a scaling of the matrix Y_p between time steps. We thus conclude that distance to the single landmark can be observable provided that parallax from motion is present. In certain applications, judicious path-planning could be implemented to take advantage of this result. We leave such considerations to future work.

At its heart, this added observability comes from the fact that the system is nonlinear, requiring the re-evaluation of Y_p at each time step. Thus, the nonlinearities present in the model do enter into the equations and affect the EKF in spite of its linear nature, which makes no explicit attempt to account for them. Lastly, we note that the nonlinear effects described here are corroborated by the simulation results as can be seen in Chapter 4.2.

4. Simulation

The proposed landmark-based navigation method was first tested with a Simulink quadcopter simulation, which allowed testing in a wide variety of circumstances which are explored in this chapter.

4.1. Sensor Simulation

In the simulation, sensors were modeled as having additive, Gaussian, non zero-mean noise, represented by a standard deviation, σ , and a bias term, b . Snippets of the simulated IMU signals are given in Figure 5.

Table 1: IMU and Camera, Noise and Bias

	IMU						Camera	
Sample Rate	25Hz						5Hz	
Unit	m/s ²			rad/s			degrees	
Std. Dev.	σ_{a_x}	σ_{a_y}	σ_{a_z}	σ_p	σ_q	σ_r	$\sigma_{x,cam}$	$\sigma_{y,cam}$
	0.356	0.6498	0.3846	0.022	0.0208	0.029	0.4	0.4
Bias	b_{a_x}	b_{a_y}	b_{a_z}	b_p	b_q	b_r	$b_{x,cam}$	$b_{y,cam}$
	0.044	-0.0022	0.071	-0.0028	0.005	0.00154	0	0

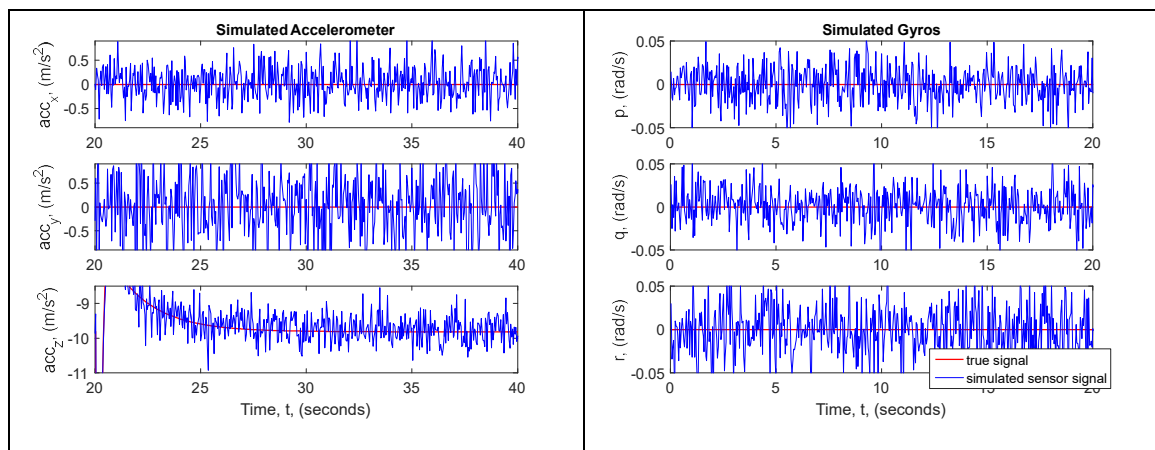


Figure 5: Simulated IMU Data

The IMU noise levels indicated in Table 1 were chosen to match those obtained from flight test data as described in Chapter 5.2, as were the sampling rates for both the

IMU and camera. The camera error is also discussed in Chapter 5.2, though it had to be estimated based on the size of the landmarks rather than simply computed as a standard deviation.

4.2. Test Cases

The simulation allowed for a wide variety of test cases, which differed in four primary aspects: flight path, number of landmarks in the scene, IMU and camera noise levels, and monocular vs stereo setup. Each aspect has the ability to affect the performance of the navigation filter significantly via increases in state uncertainty or even loss of observability. Indeed, the results from our analysis in Chapter 3.5 are entirely confirmed here in the behavior exhibited by the simulation.

4.2.1. Process and Measurement Noise

We begin with an analysis of the overall noise levels on both the IMU and camera, which drive the process and update steps of the EKF respectively. We give, as a first example, Figure 6, which shows the true and estimated flight path of the simulated quadcopter in a circling pattern with two landmarks available in the camera field of view, to ensure full observability. The camera is oriented facing in the forward direction in the quadcopter body frame and angled downward at 60° . The quadcopter maintains a heading of 0° , which corresponds to the positive X-axis.

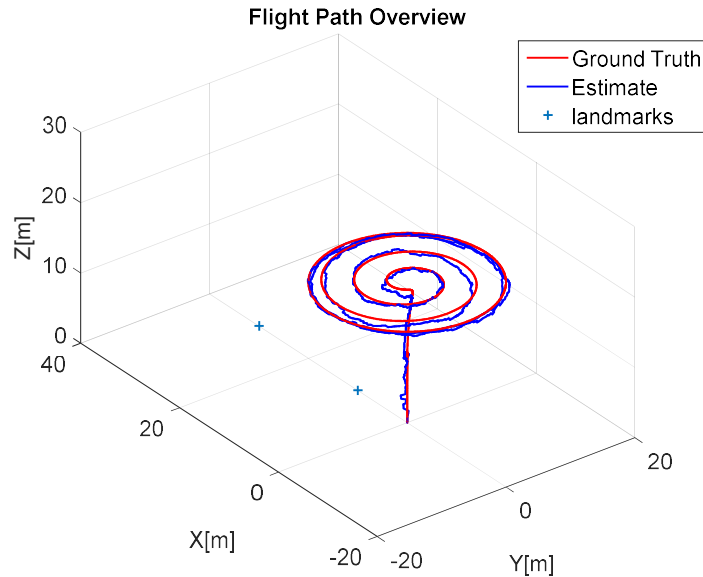


Figure 6: Simulation flight overview with flight test noise levels

In Figure 6, the noise levels on both the IMU and camera are tuned to match those observed in the flight data as given in Table 1. Figure 6 indicates good estimation performance of the landmark filter in a case where the IMU and measurement error are at typical levels. Nevertheless, idealizing assumptions have been made in the course of simulation, and actual flight test performance was often found to be slightly degraded compared to simulation results. Flight test performance will be further discussed in Chapter 5.2, which details how the parameters in Table 1 were obtained, as well as the results from post-processing of flight test data.

To test the capabilities of the algorithm under less amenable conditions, noise parameters were increased in simulation to examine the resulting behavior. Several examples are given in Figure 7, which shows position estimation under increasing levels of noise and bias. In the upper-left, we see the position estimation for the same flight shown in Figure 6. The upper-right gives the result when noise levels are increased by a factor of 5; the lower-left has noise increased by a factor of 10; and the lower-right by a

factor of 25. We observe good stability in our simulation even at some of the higher noise-levels, indicating a certain robustness in the presence of poor IMU data and camera measurements. Upon reaching 25 times the noise levels in the flight test, however, it is safe to say that the algorithm's limits have been surpassed for this flight scenario. Indeed the only reason such a level of noise was attempted at all was to break the limit.

We observe also that the dead-reckoning solution, which uses only the prediction step of the EKF, drifts rapidly even in the case with the lowest noise since it amounts to an integration of the noisy IMU. Moreover, to get a position estimate, the accelerometers must be integrated twice, which produces a quadratic divergence from ground truth.

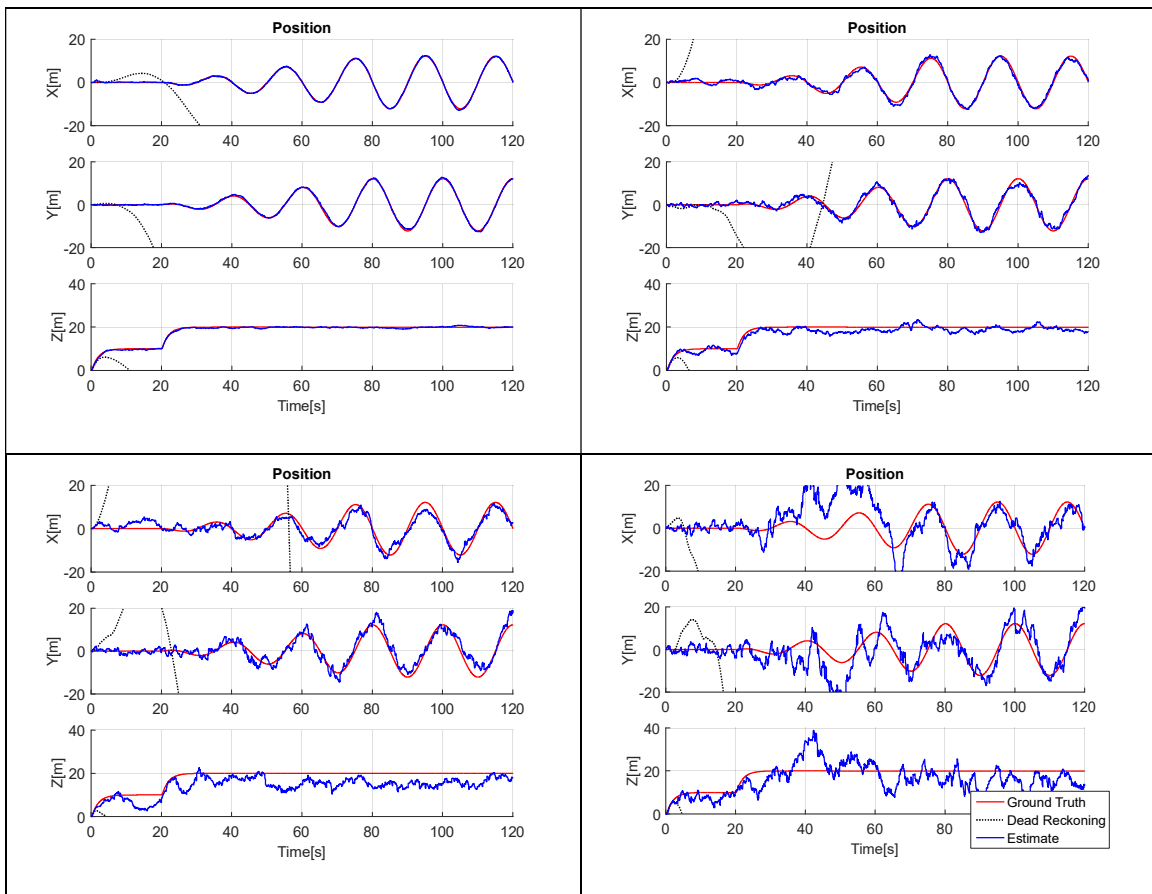


Figure 7: Position estimation with increasing noise level. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

The results in Figure 7 provide a good illustration; however, to quantify the estimation performance in a more concrete way, we consider the 3σ estimation bound(s) taken from the diagonal terms in the state autocovariance matrix P , which represent a 99.7% confidence interval. These bounds are shown in Figure 8, which displays filter performance in terms of the absolute estimation error and the $\pm 3\sigma$ bounds. The estimation error plots more clearly show the steadily increasing estimation uncertainty with increasing noise and bias, as expected. The X-position bounds were then averaged across time and plotted vs noise level in Figure 9, which shows a surprisingly linear behavior. Thus, estimation error scales linearly with overall noise and bias levels.

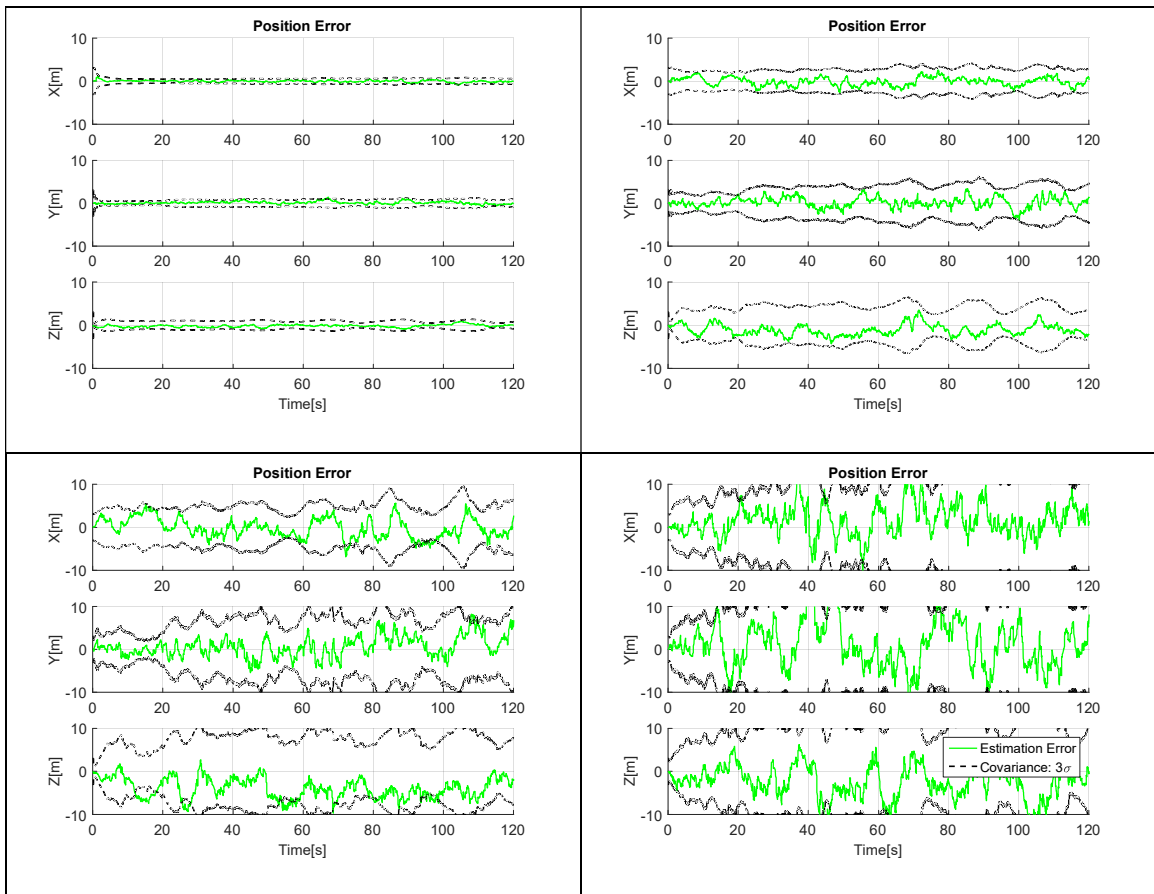


Figure 8: Estimation error and covariance bounds with increasing noise. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

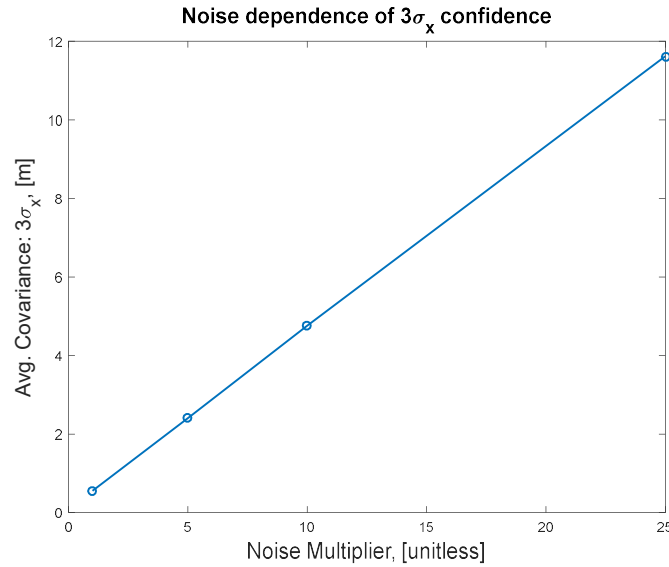


Figure 9: X-position error noise dependence

So far, we have considered only position estimation; however, the developed algorithm estimates velocity and attitude in addition to position. The estimation and error results for these state variables in the same simulated flight scenario are now given.

First, the velocity estimation is given in Figure 10. The velocity estimates are comparable in behavior to the position estimates; however, we observe that the dead reckoning solution does not drift as quickly as in the position estimates. This is due to the fact that velocity estimates come from only a single integration of the IMU accelerometers, which produces a more linear-like growth in dead reckoning error over time. There is also a tendency to underestimate the z-component of velocity at higher noise levels. It is unclear why this occurs.

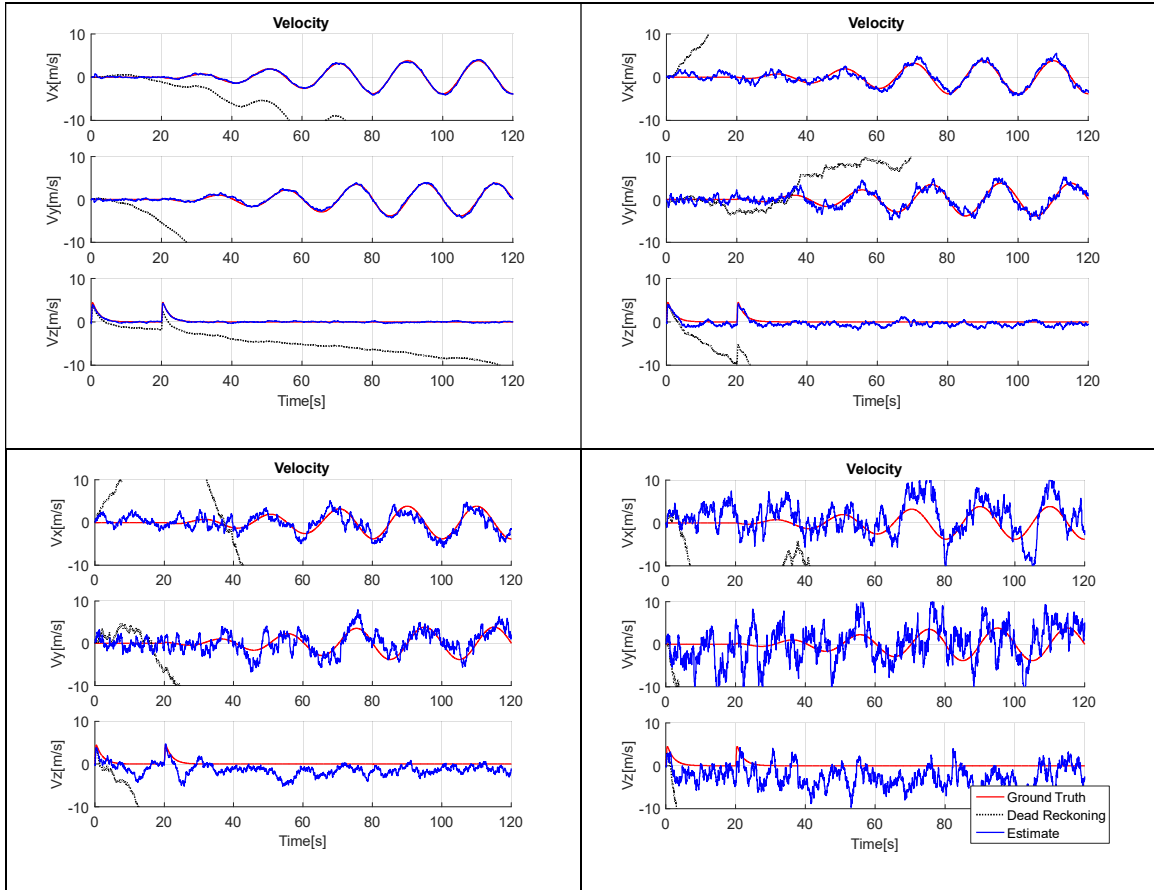


Figure 10: Velocity estimation with increasing noise level. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

Next, Figure 11 gives the velocity estimation error and covariance bounds. As before, the velocity error does not differ significantly in behavior from the position estimates. Again, we see the tendency to underestimate the z-component of velocity, even pushing the boundaries of the covariance. This phenomenon may be worthy of future investigation; however, behavior is still nominal in the more reasonable lower-noise cases.

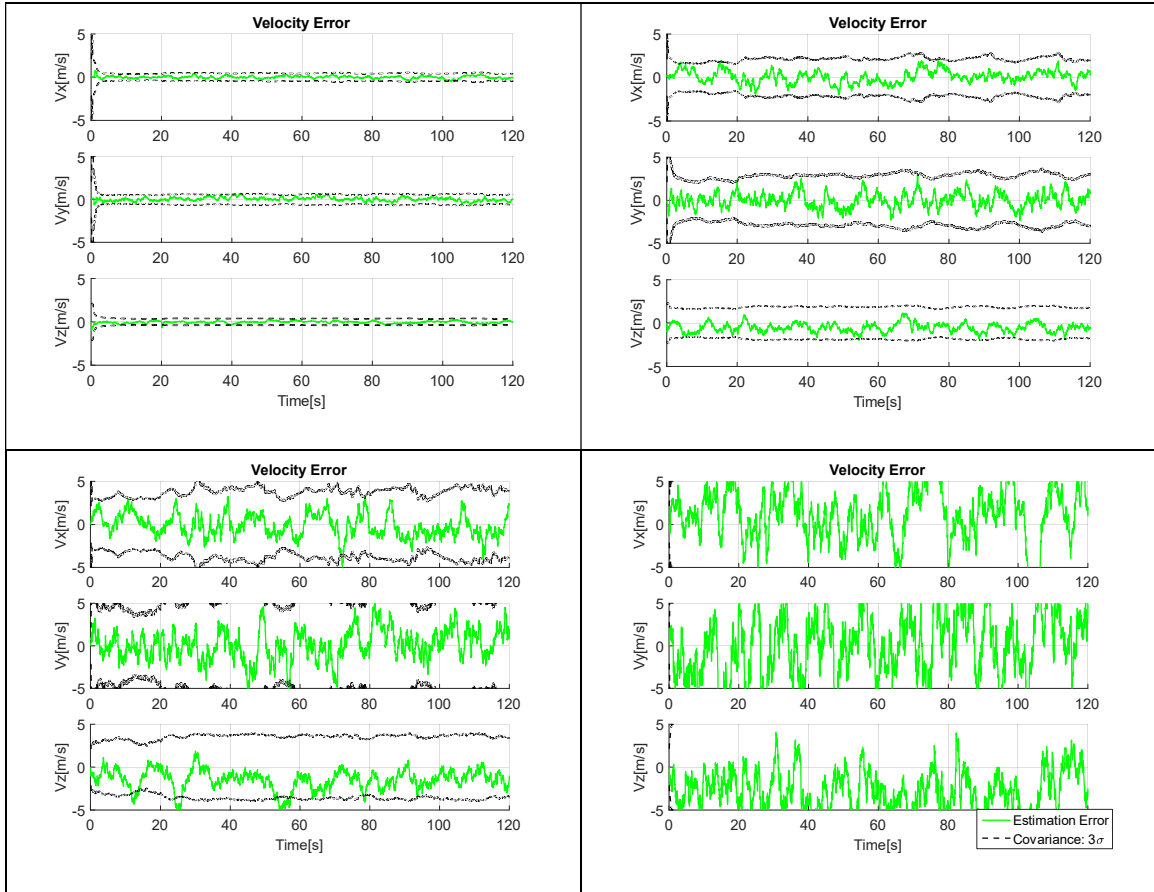


Figure 11: Velocity estimation error with increasing noise level. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

Lastly, we consider estimation performance for the attitude variables $\boldsymbol{\rho} = [\phi, \theta, \psi]$. As before, we give estimation results in Figure 12. Again, the performance is comparable to the position and velocity estimates. The notable exception is the surprisingly good performance of the dead reckoning estimates, suggesting that the landmark filtering almost was not even necessary in the lowest-noise case. This behavior is corroborated by the flight test results given in Chapter 5.5, indicating that the onboard Pixhawk flight computer's rate gyros were more accurate than its accelerometers. Such behavior makes sense on an autonomous aerial vehicle, where attitude stability is of greater importance than accurate position and velocity estimation.

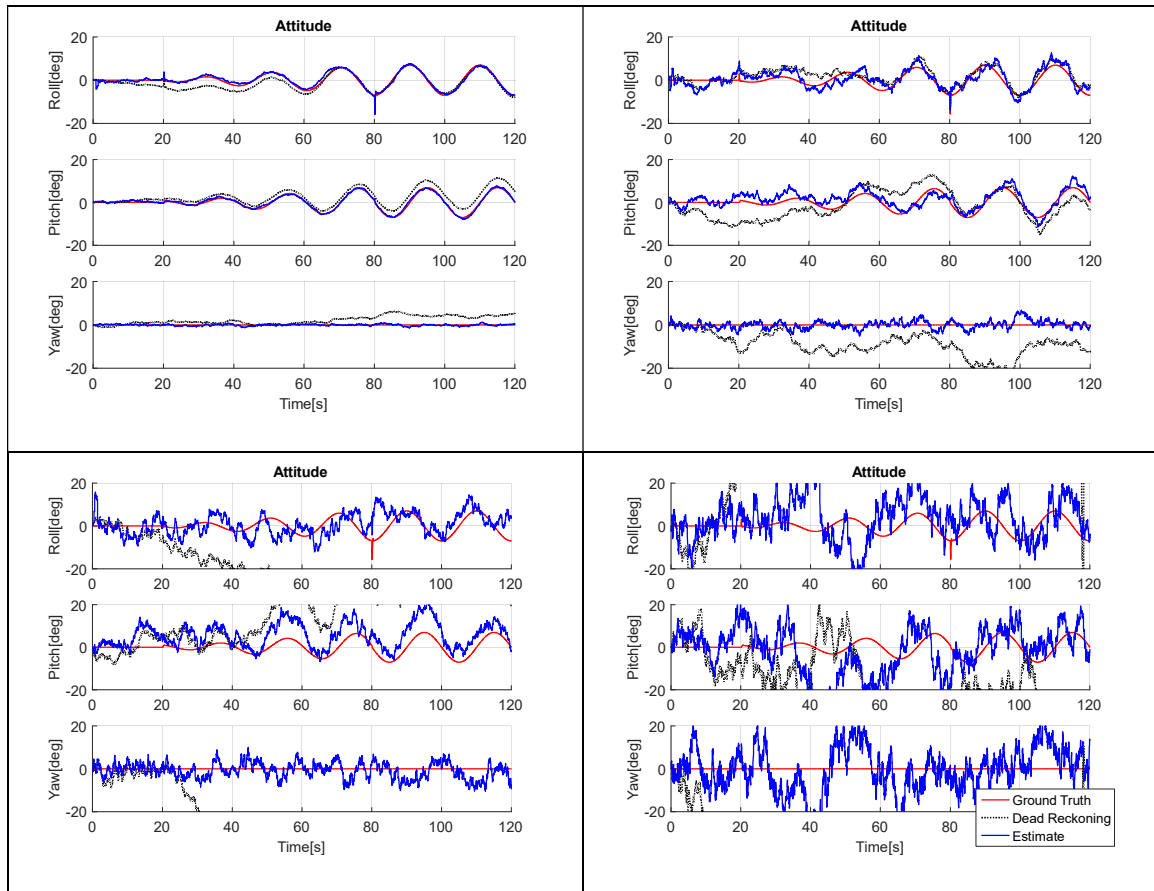


Figure 12: Attitude estimation with increasing noise level. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

Figure 13 gives the error and covariance plots for the estimations in Figure 12. Once again, performance is nominal with the exception of a tendency to overestimate the pitch angle in the presence of high noise, similar to the underestimation of the z-component of velocity. Again, this may warrant future investigation as the cause is unclear. For this study we accept that the performance at lower noise levels is nominal.

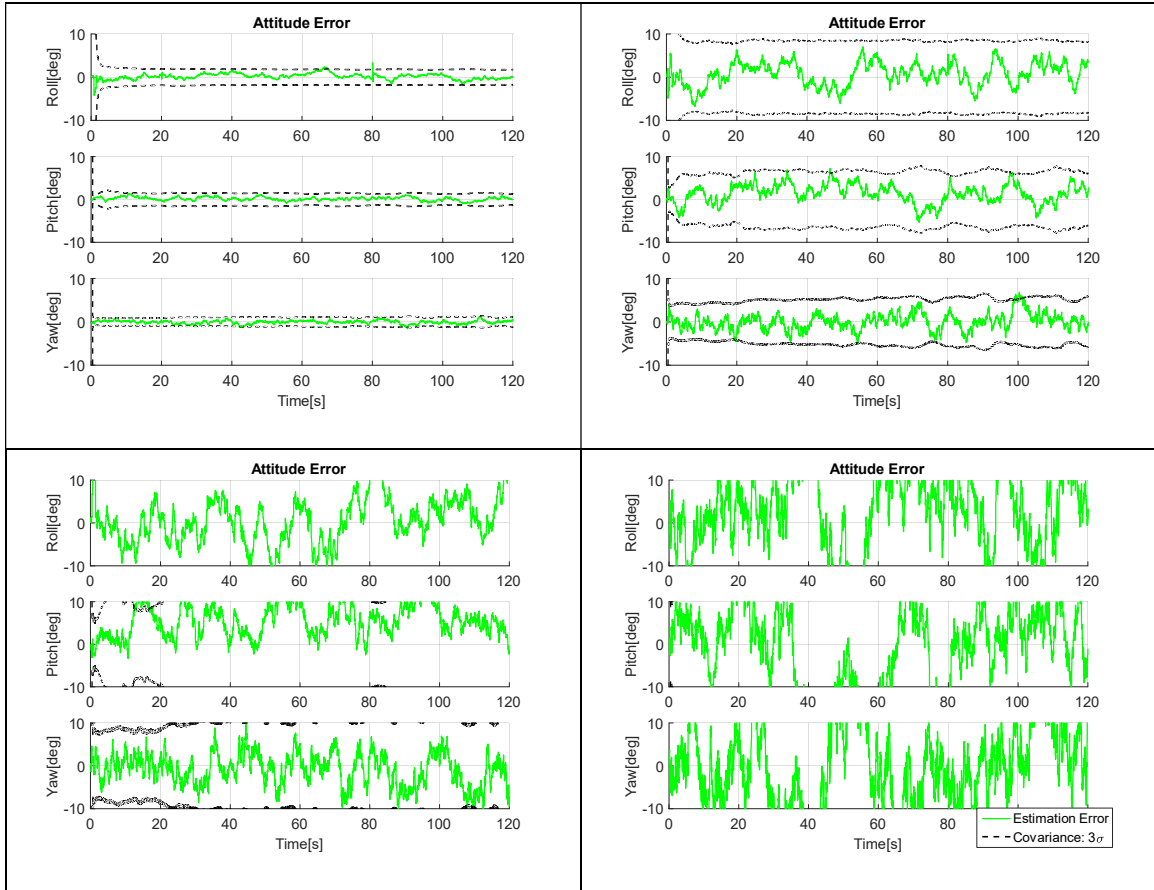


Figure 13: Attitude estimation error with increasing noise level. Upper left: flight test noise levels; upper right: 5x flight test noise levels; lower left: 10x flight test noise level; lower right: 25x flight test noise level.

4.2.1. Number of Landmarks

We now turn our attention to the estimator's dependence on the number of landmarks visible in the scene. In the simulations shown so far, there have always been four landmarks available; however, it is of great interest to know how the algorithm performs when few landmarks are in view, and observability issues creep in.

Figure 14 shows estimation and error for the same flight path scenario as in previous examples, but with only two landmarks available in the scene. Surprisingly, there is no noticeable difference from the case with four landmarks given in Figure 6. This is due in part to the scale of noise levels relative to the distances being covered; however, it is clear nonetheless that two landmarks are enough to provide the GPS-like measurement that landmark navigation is intended to produce, which makes sense given that two landmarks in a general configuration are enough to provide complete observability. In cases with higher noise, more than two landmarks may be desirable, but two landmarks is the minimum for system observability as shown in Chapter 3.5.

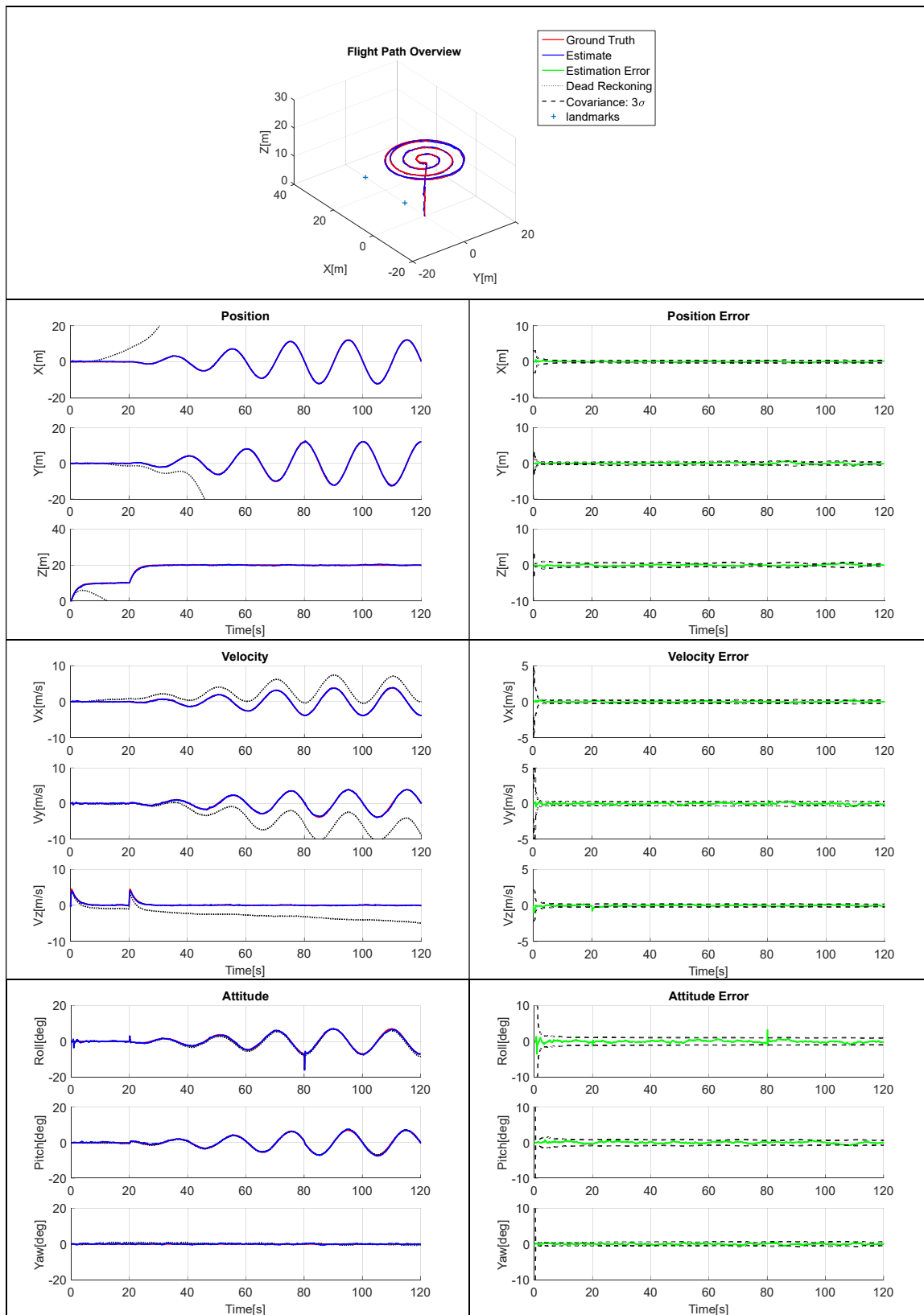


Figure 14: Estimation with two landmarks

Next, Figure 16 shows the estimation and error with just one landmark available in the scene. With one landmark, the state is no longer fully observable and results begin to differ in some interesting ways. First, the estimated flight path exhibits an overall shift in the negative Y-direction. Second, the yaw estimate drifts slowly away from the ground truth with no sign of correction, and even exceeds the 3σ bounds by the end of the simulation. Additionally, there is a large estimation uncertainty prior to approximately 60s, particularly in the X and Z position estimates, which is greatly reduced after this time.

Of course, all of these results are explainable in terms of the observability analysis from Chapter 3.5. First, the shift in Y and the drift in yaw are the coupled attitude-position unobservable state manifesting as an azimuthal rotation about the landmark. Second, the large uncertainty and error prior to 60s comes from the unobservable direction toward the landmark, and it disappears after roughly 60s since this is when the vehicle begins its continuous circling motion and the nonlinear parallax corrections come into play.

The nonlinear effects are particularly visible during the brief period of hover at 10m altitude which occurs between 10 – 20s. A side view of this period in the flight path is given in Figure 15, which shows the estimation drift toward/away from the landmark during this hover, and the subsequent stabilization once the vehicle begins moving.

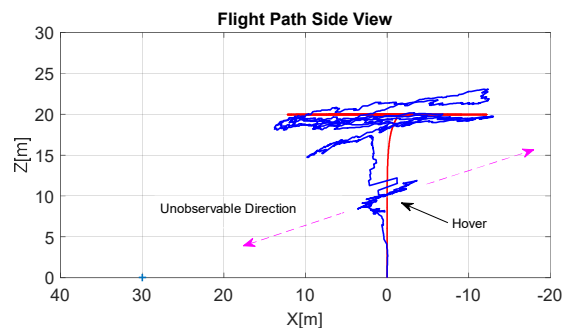


Figure 15: Side view of hover

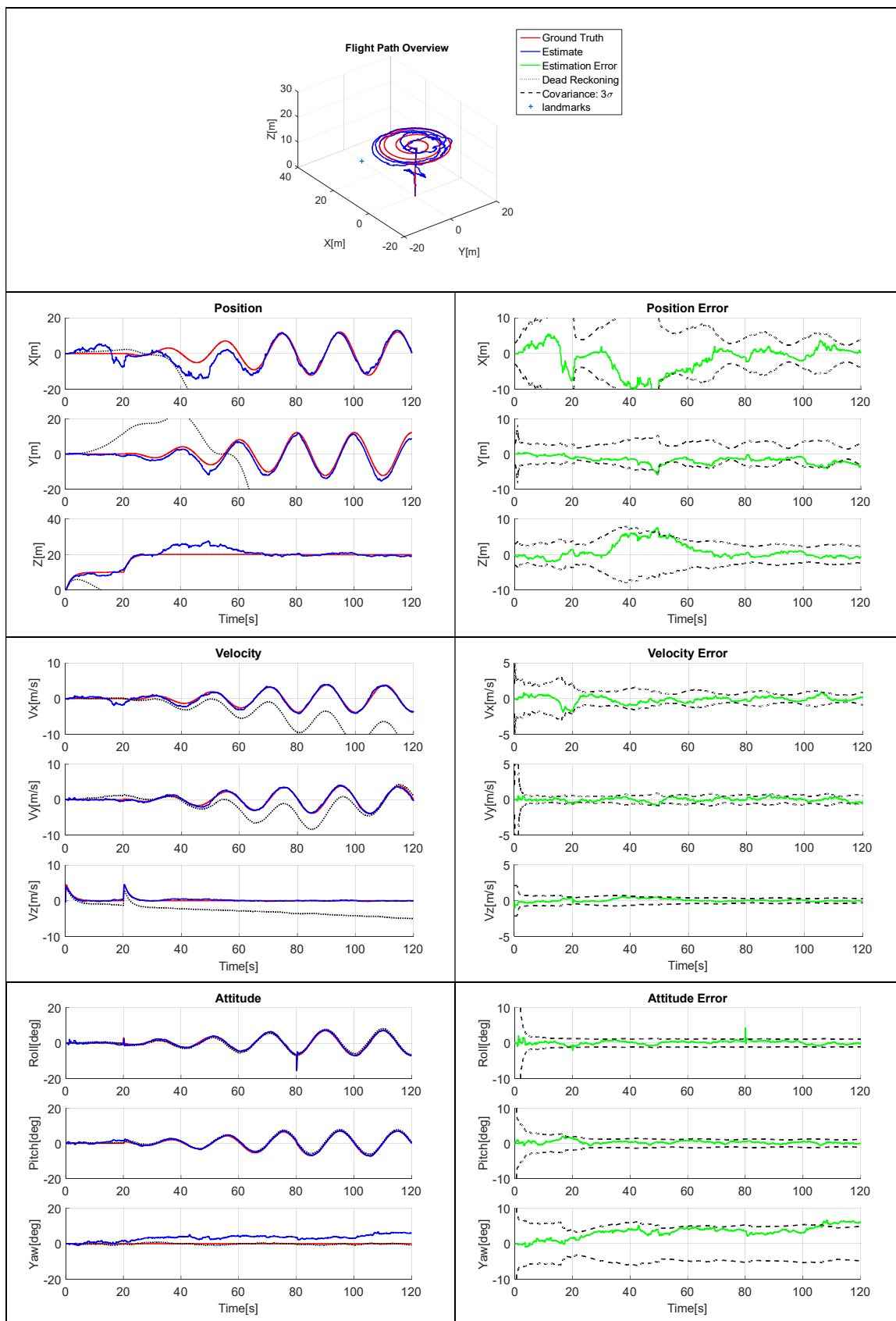


Figure 16: Estimation with a single landmark

4.2.1. Flight Path

We next consider the effect that our choice of flight path has on the estimation performance. As we saw in the previous Chapter (4.2.1) as well as in Chapter 3.5, the flight path does matter due to nonlinearities. In particular, hovering flight gives poorer results than does translating flight. Thus, the algorithm was tested with a flight path on which it would undergo extended periods of hovering flight.

An example flight along a hovering flight plan is shown in Figure 17, in which there are two landmarks available. The vehicle hovers at each corner of the ‘box’ pattern for approximately 40 sec before moving quickly to the next corner. For the moment, we do not see the problems of estimation drift during hover as we did before. This is due to the fact that we have two landmarks so that the system is completely observable.

We do, however, notice some attitude error in Figure 17 during the maneuvers to transition between hover locations. This is due predominantly to integration error since the simulation’s maneuvers were so abrupt as to be comparable in duration to the sampling time step Δt_s , leading to high estimation error at an individual time step for which the simulation has not properly registered the change. The error then persists for a brief time afterward for the same reasons discussed in Chapter 3.4 on underweighting. Unfortunately, the underweighting solution is not applicable here; however, in reality the maneuvers of a typical flight platform will not be nearly as aggressive, so a solution to the problem is not necessary.

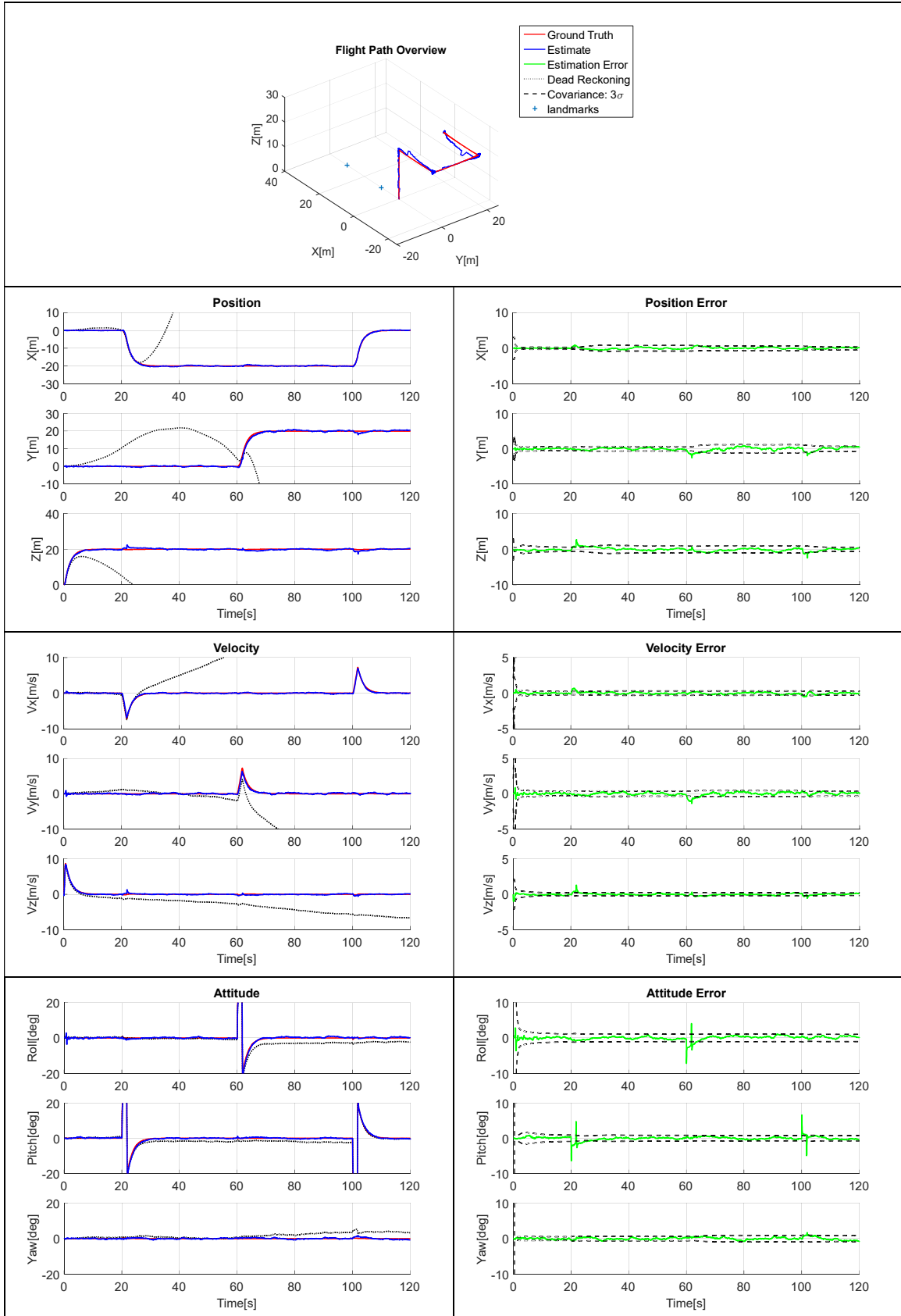


Figure 17: Hovering and abrupt maneuvers

While Figure 17 illustrated the observability of a two-landmark scenario even during hover, the one-landmark case is of interest to us as well. Estimation with a single landmark during hover is given in Figure 19. Here we see all the observability problems outlined previously in Chapter 3.5. In particular, the estimate drifts downwards along the unobservable direction towards the landmark just like before, and then is corrected momentarily when the hover is broken and the vehicle translates to the next hover location as shown in Figure 18, which gives the side-view of the scenario in Figure 19. However, the translation period is not long enough in duration to truly re-stabilize the estimate, and consequently it drifts even faster following the correction. By the finish, position estimation errors are both enormous and completely unaccounted for by the covariance bounds. Recovery is likely impossible.

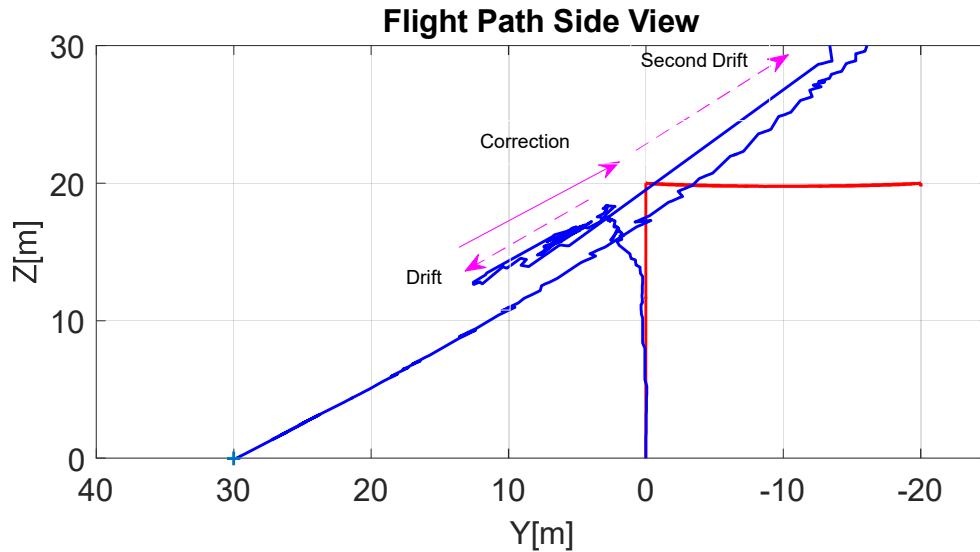


Figure 18: Side view; extended hover flightpath

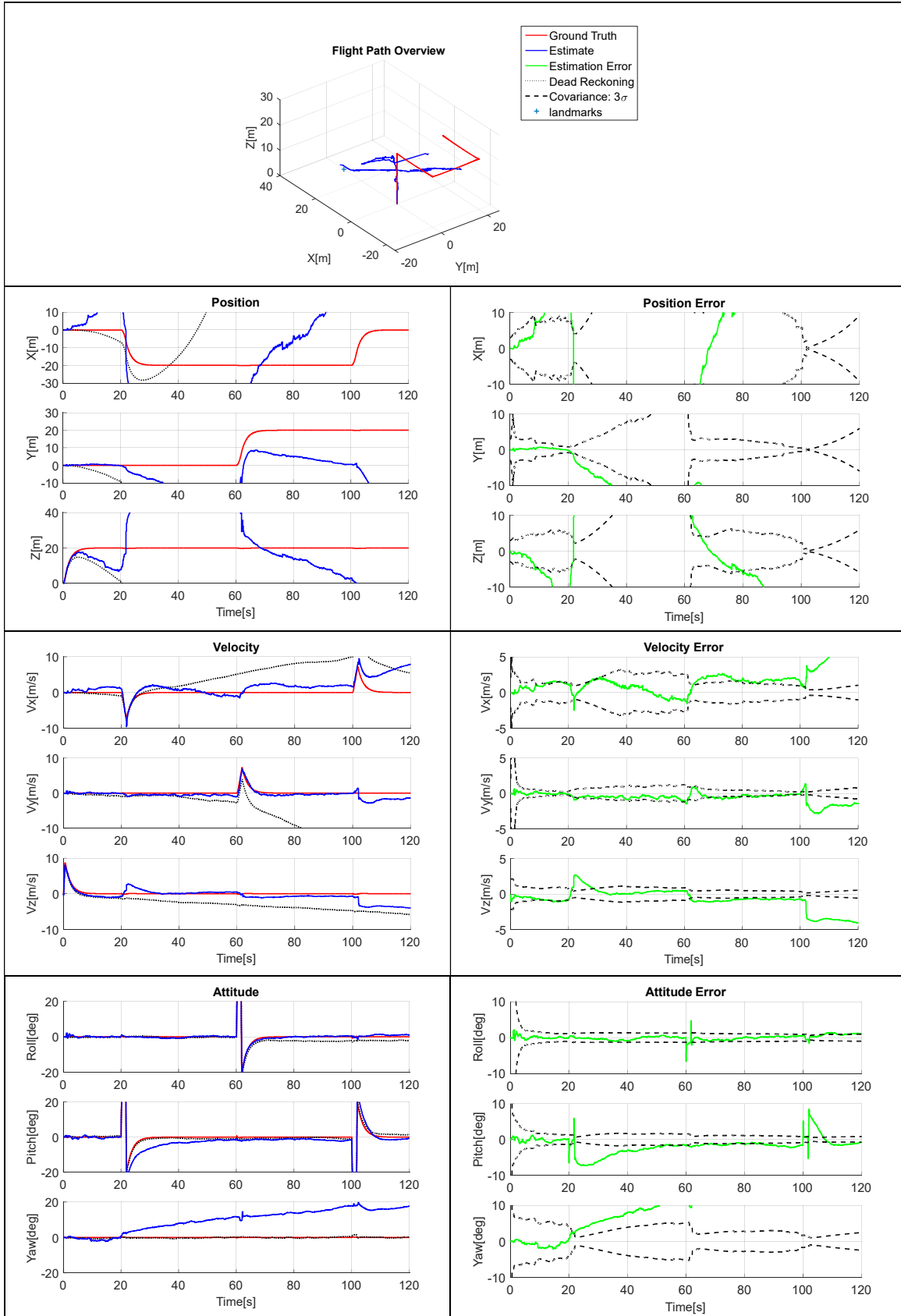


Figure 19: Hovering and abrupt maneuvers with one landmark

4.2.2. Monocular vs. Stereo

Lastly, we consider the estimation performance benefits provided by the additional information entailed in a stereo measurement, or any other measurement such as LIDAR, or bearing capable of measuring the distance to the target. In particular, the measurement of distance to the target increases the rank of the observability matrix, fixing some of the previous observability problems. Most notably, the tendency for the state estimate to drift directly toward or away from the target is removed, since this parameter is now, in fact, being directly measured. We note that the depth-measurement noise is modeled as having a standard deviation error of: $\sigma_{z,cam} = \frac{10\pi}{180} \cdot z$ where z is the distance to the landmark, or ten ‘degrees’ depth error. This number could not be directly calculated, and the value given above was pessimistically chosen as 5 times more uncertain than the directional error.

We observe this performance enhancement in Figure 20, which shows a behavior similar to that of the monocular measurement in Figure 19, but without the instability caused by unobservability in the direction of the landmark. The dominant direction of drift seen here is the circular drift pattern around the landmark, since it is not dealt with in any manner by the stereo measurement. The azimuthal around the landmark can only be remedied by the addition of at least one other landmark, or the introduction of additional measurement information such as, for example, a magnetometer to measure the heading.

The circular drift pattern, as usual, manifests itself predominantly in Y-coordinate and yaw (heading) error, since these two directions are coupled in this unobservability mode. We do note, however, that there is a large jump in error at $t \cong 20$ sec during the first abrupt maneuver, which suggests that a significant portion of the drift in this unobservability mode has been generated by the sharp maneuver itself.

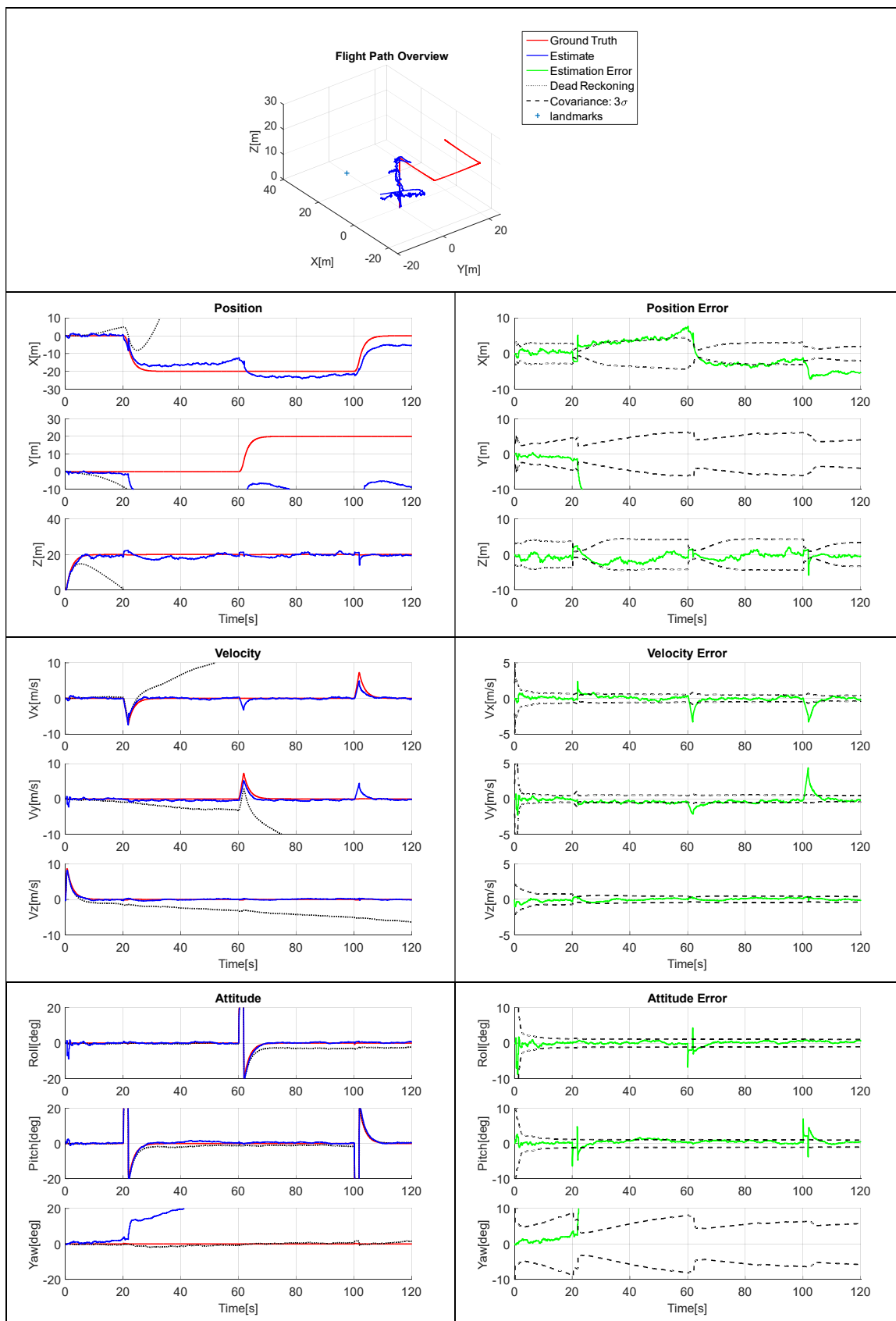


Figure 20: Stereo setup, hovering flightpath, and one landmark

Lastly, we give the performance of the stereo measurement under the original spiraling flight path in Figure 22. The results here differ only slightly from those in the monocular case given in Figure 16. Again, the notable difference is the lack of drift toward the landmark during hover that was observed before. We can see from the side view in Figure 21 that distance from the landmark is, unsurprisingly, accurately gauged even during the brief period of hovering flight at the 10 m altitude location.

We also note that while the nonlinear effects provided an implicit measurement of distance while the vehicle was in motion, the direct measurement of the stereo setup does an overall better job, even during motion, since the distance is measured directly.

Lastly, we reiterate that the circling drift pattern from all the previous cases is still present as observed in the Y-shift and yaw drift in Figure 22 since the stereo measurement of distance does not enable observation of this phenomenon.

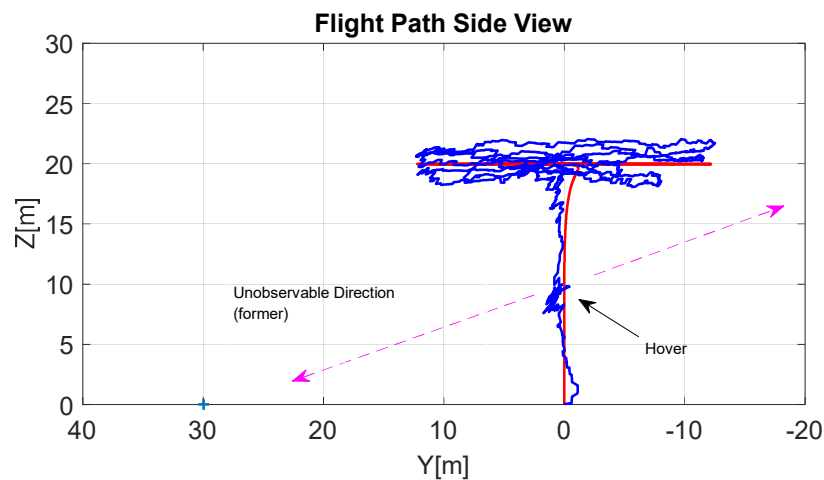


Figure 21: Side View; Stereo Measurement

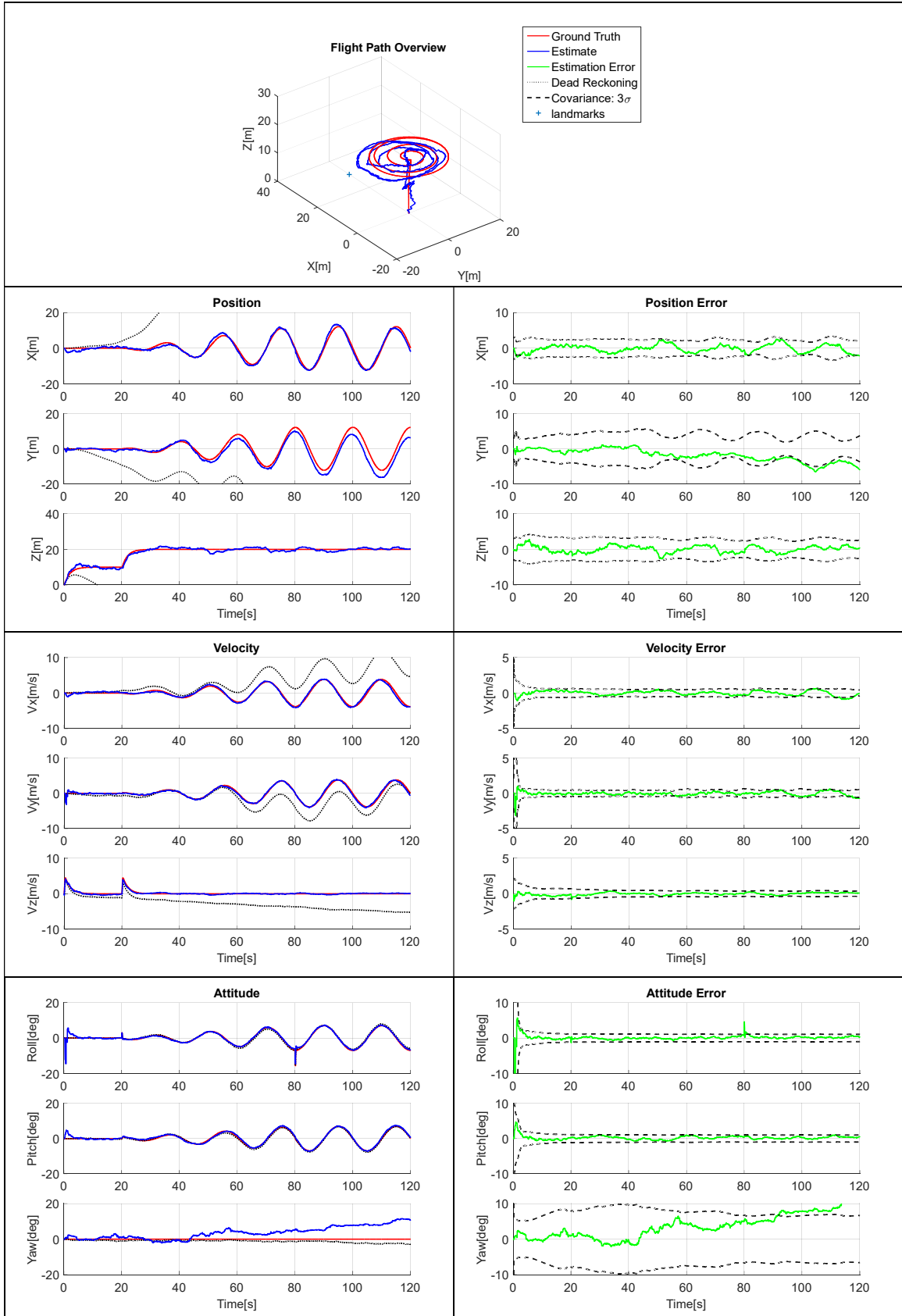


Figure 22: Stereo setup, original spiraling flightpath, and one landmark

5. Quadcopter Flight Testing

The landmark navigation algorithm was used to post-process data from flight tests of a quadcopter capable of carrying the required sensors. The results are compared and contrasted to those from the simulation.

5.1. Flight Test Setup

The Iris quadcopter shown in Figure 23 carries the sensor package, which consists of a GoPro Hero 3+ camera (1080 x 1920 resolution, 60 frames/sec), a Pixhawk flight computer which provides an IMU, and a U-Blox GPS unit. The GPS data are not used in the filter, but are used to validate the filter results. In the flight tests, as shown in Figure 24, a landing target was placed on the ground with 7 colored balls providing the predefined landmarks, where each arm of the white cross is 0.6 m long. Figure 25 gives a representative GPS track from a flight test, and Figure 26 presents sample rate gyro and accelerometer data collected from the onboard IMU.



Figure 23: Iris Quadcopter fitted with GoPro



Figure 24: Aerial View of Landmark Cluster

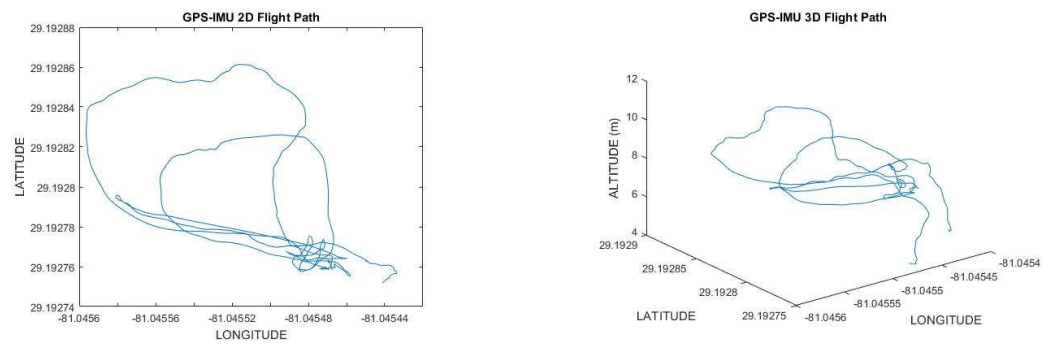


Figure 25: Sample GPS Track

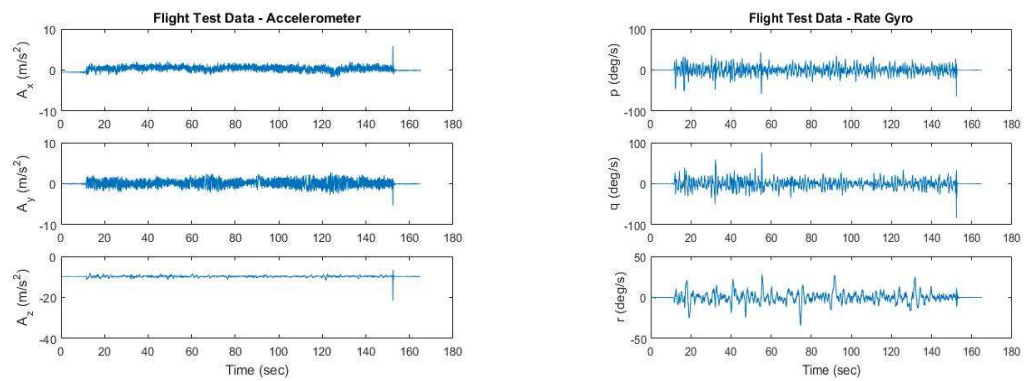


Figure 26: Sample IMU Data

5.2. IMU and Camera Parameters

As mentioned previously, the flight test sensor package consisted of a Pixhawk PX4 flight computer with an onboard IMU, a U-Blox GPS unit, and a GoPro Hero 3+ camera.

First, the Pixhawk flight computer provided accelerometers, rate gyros, a magnetometer, and a barometer. While the magnetometer and barometer were not used by the developed landmark navigation method during the simulation or post-processing, it is easy to incorporate these sensors into the algorithm if available. In particular, a magnetometer could help resolve the azimuthal unobservability problem associated with a single landmark. While the Pixhawk IMU runs at a minimum of 100 Hz (Dronecode), it only logs IMU data at 25 Hz. While the Pixhawk can be made to log at a faster rate, the default rate of 25 Hz was a convenient number for post-processing. Thus, the IMU for the simulation and post-processing effectively runs at 25 Hz.

To estimate the noise and bias levels, we performed a test flight sitting on a level surface with the Iris motors running at takeoff power. The mean and standard deviation of the relevant portion of this ‘flight data’ then provided the bias and noise levels respectively. A snippet of the noise test flight data is shown below in Figure 27.

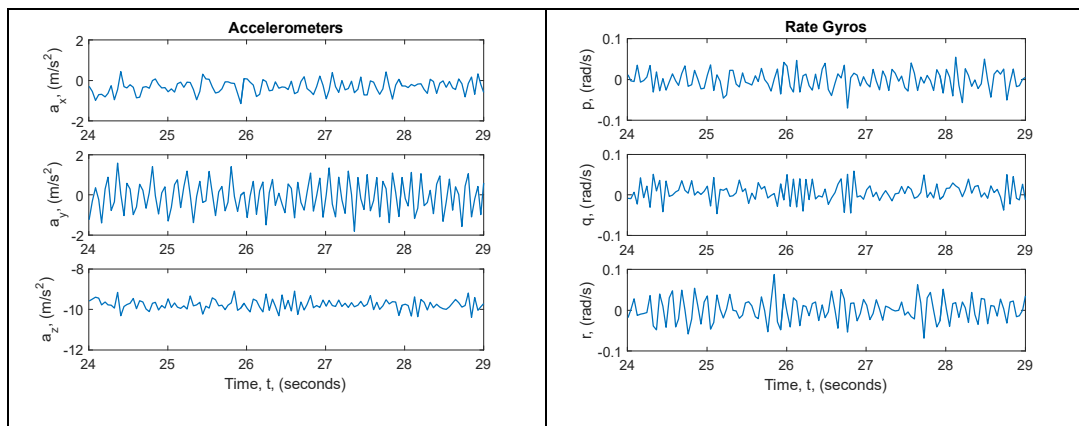


Figure 27: Flight test snippet for IMU uncertainty estimation

The U-Blox GPS unit, like the magnetometer and barometer, is not used by the

landmark algorithm for the reason stated above. However, it does provide the ability to compare our landmark results to the IMU-GPS navigation standard, which is taken to represent the ‘ground truth’ estimate. The U-Blox GPS has a published accuracy of 2.5 m, and a sampling rate of 5Hz (LEA-6 Series).

For the monocular tests, a single GoPro Hero 3+ was used. The GoPro(s) had a resolution of 1960×1080 pix and ran at 60 fps (GoPro). However, it is unreasonable to expect that any vision-detection algorithm could run at a frame-rate as high as 60 fps. Thus, the frame-rate from the GoPros(s) was reduced to the more pragmatic rate of $\frac{60\text{fps}}{12} = 5$ fps by only processing every twelfth frame.

Lastly, the camera error was calculated based on the size of the landmarks. Centroid detection errors were the dominant source of camera measurement error and were at most approximately $\frac{1}{5}$ of a diameter of the detected ball. Given that the largest ball was 10 cm in diameter, and that the quadcopter was, at the closest, ~ 3 m from the target, the small angle approximation gives an error of: $\sigma_{cam} \cong \frac{1}{5} \left(\frac{d}{z} \right) = \frac{10\text{cm}}{5(3\text{m})} = 0.00667 \text{ rad} = 0.38 \text{ deg}$. This number was then rounded up to 0.4 deg to ensure a pessimistic estimate of the camera accuracy. We note also that the landmarks were on the order of 20 pixels in diameter, which indicates that sub-pixel accuracy may be marginally relevant. In particular, aliasing may have subtle effects on the distribution of centroid errors. Such error-biasing can be corrected, but was not deemed necessary in this thesis.

5.3. Camera Calibration

The GoPro cameras used had a considerable amount of image distortion primarily in the form of radial distortion or ‘fish-eye’. Most of this effect was eliminated by the

GoPro image processing software; however, some residual distortion remained. To correct the remaining distortion, and also to obtain the intrinsic calibration matrix for the camera, multiple camera images of a checkerboard from different angles were processed using the MATLAB Camera-Calibrator application. A few example images of the checkerboard, which had squares 22 cm on a side, are given in Figure 28.

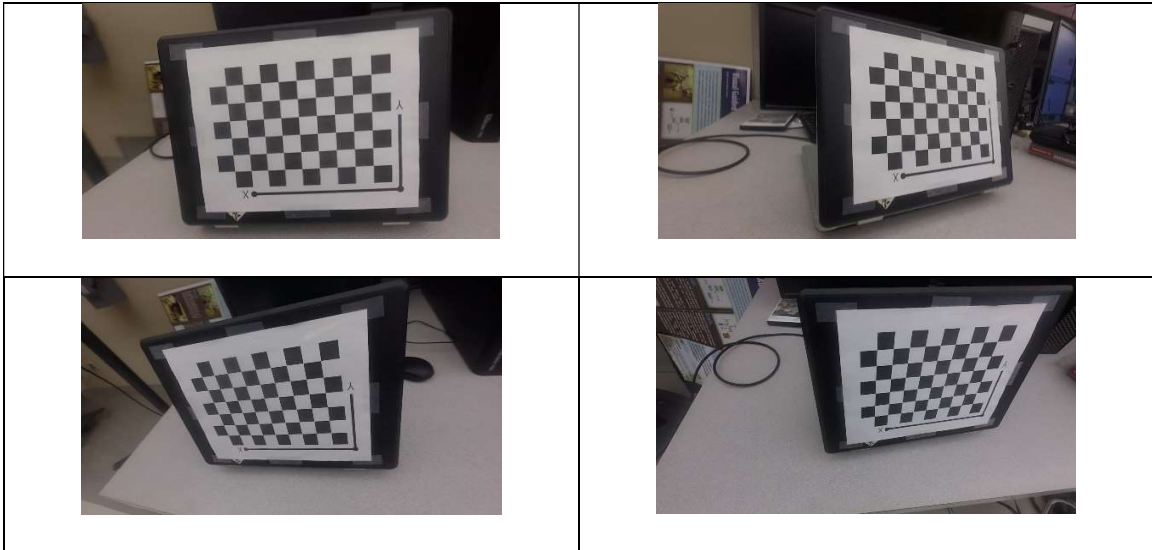


Figure 28: Checkerboard pattern for Calibration

5.4. Color Detection

For the flight tests, landmarks were identified using color recognition. The color-detection filter was obtained using the MATLAB color thresholder application to tune for the desired color(s) in the Hue-Saturation-Value (HSV) color coordinates. The resulting filter(s) produces a sequence of masked images when applied to the flight video frames. When the filter is properly tuned, the landmarks are easily detectable and distinguishable between frames. The detection process using the MATLAB color thresholder is illustrated in Figure 29, which shows, as an example, the tuning for the light green ball toward the lower-right of the landmark cluster, which is intended to simulate a landing target. Not all colors were equally amenable to the tuning process. In particular, the darker-colored balls

were difficult to tune such that the ball was reliably observed while simultaneously minimizing false-positives. The presence of false positives required an additional filter with both area and aspect-ratio thresholds to eliminate colored regions passing through the HSV filter that were too small and/or insufficiently spherical. These cutoffs helped to distinguish between actual detections of the colored landmarks and detections of similarly-colored spots in the grass/rocks. In the example shown in Figure 29, all seven landmarks were detected successfully.

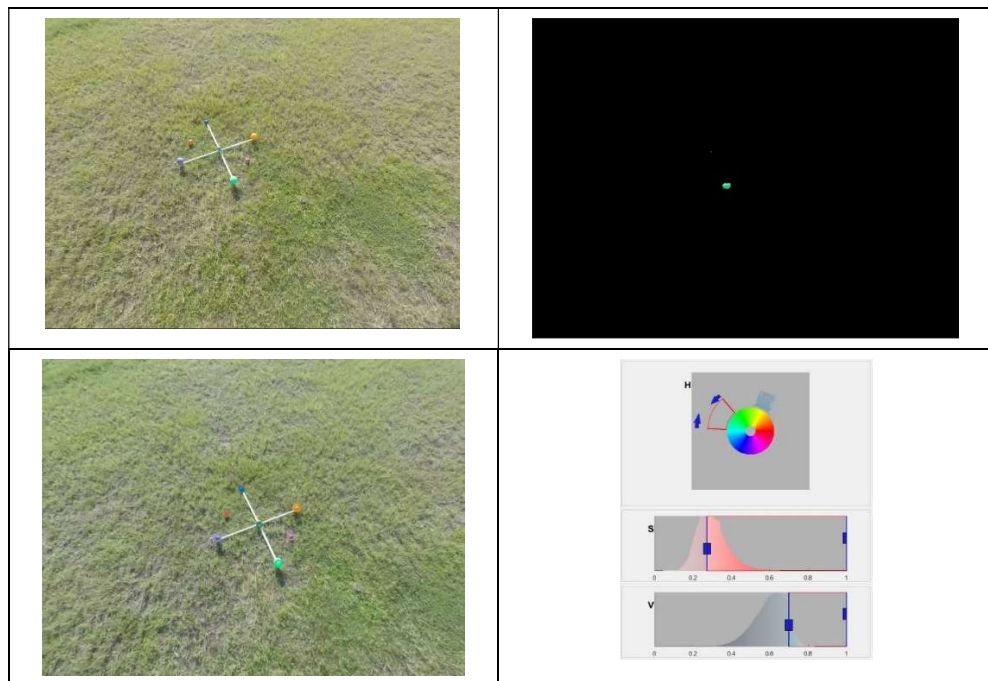


Figure 29: Color Detection Process Overview

5.5. Post-Processing Results

We now give the results from the post-processing of the flight data, starting with the monocular case. Figure 30 gives the monocular measurement position estimation from a portion of the flight data for which the landmark cluster shown in Figure 24 is in view. The landmark cluster enters view at approximately $t \cong 87$ s and leaves view at $t \cong 125$ s.

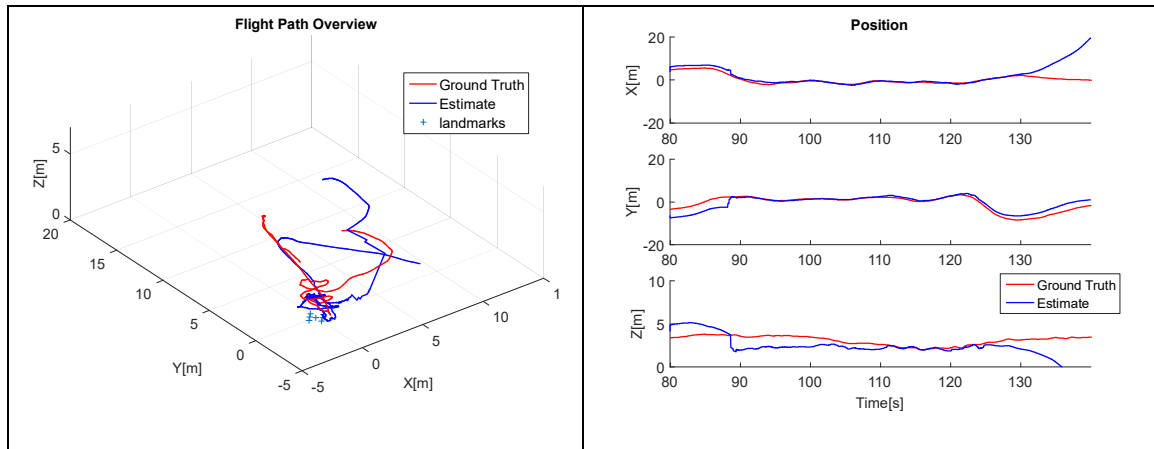


Figure 30: Post-processing position estimation

Prior to the landmarks entering view, there are notable errors in the position estimation. These errors are corrected upon acquisition of the landmarks, with the exception of the Z-estimate which appears to have overcorrected. However, it is not completely correct to interpret the position discrepancy as representing the actual estimation error since the ‘ground truth’ is in fact an IMU-GPS estimate which is prone to errors of its own. Thus, it is possible that the so-called ‘ground truth’ path has error, since the U-Blox GPS has errors on the order of 2.5 m (U-Blox). Though the filtered estimate will be somewhat more accurate, it is difficult to say how much. Finally, upon loss of the landmarks at $t \cong 125$ s, the position estimates begin to drift as expected.

Next, Figure 31 and Figure 32 show the velocity and attitude estimation for the same portion of the flight data as in Figure 30. The velocity estimates show nominal behavior throughout, and drift at the finish as before. There is, however, a brief period immediately following the initial correction phase at $t \cong 87$ s during which the velocity estimates are unstable. This is a possible area for future investigation, as it is unclear why this occurs. Unsurprisingly, the attitude estimates are accurate since the IMU gyros are considerably more precise than the accelerometers.

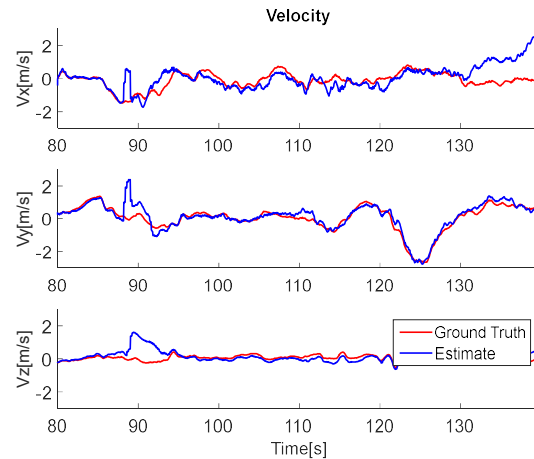


Figure 31: Post-processing velocity estimation

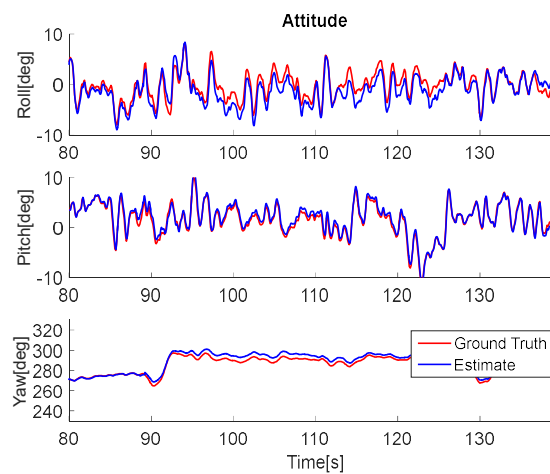


Figure 32: Post-processing attitude estimation

Overall, the landmark algorithm exhibits acceptable performance for a vision-based navigation method considering the low quality of the equipment used. These results are for the basic landmark method and can be expanded upon to attempt better performance in a variety of ways. Known size of landmarks may be incorporated as a form of depth measurement similar to the stereo setup shown in Chapter 4.2.2 to improve observability. Even the shape of a non-symmetric landmark could, hypothetically, be used to correct the azimuthal uncertainty. The following chapter discusses yet another improvement.

6. Hybrid Navigation

A key application of the landmark navigation method is to employ it in conjunction with a more traditional non a priori method such as an optical flow or odometry. In this thesis we consider the landmark filter in conjunction with the homography-based method developed in (Chavez, et al., 2017).

6.1. Homography Measurement

The homography algorithm has some unique advantages for attitude estimation to ensure critical vehicle performance during periods in which no landmarks are visible, and complements our landmark method excellently (Chavez, Myhre, & Prazenica, 2017). The basic idea of the mission layout is given in Figure 33. The vehicle starts at a known location with landmarks available, flies a route over unknown terrain in which it relies on the non a priori homography method, and then corrects any drift which has occurred upon return.

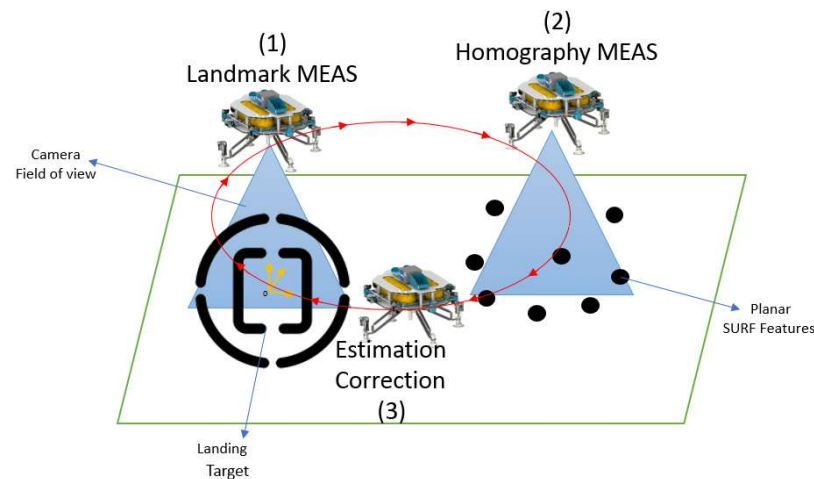


Figure 33: Hybrid filter mission overview

The homography method relies on the tracking of feature points in a planar arrangement. As shown in Figure 34, the projections of the co-planar feature points P are denoted X_1 and X_2 in the two consecutive frames from adjacent time steps. The two frames

are separated by R and T which are the frame-to-frame rotation and translation of the vehicle respectively. Since the points P are co-planar, the so-called ‘homography matrix’, H , can then be derived such that $X_2 = HX_1$. It can be shown that for the homography matrix to be determined uniquely, a minimum of four points are required; however, random feature points can be used, ensuring that the minimum is always amply met. The assumption that these points come from features on the ground ensures their approximately co-planar arrangement. The rows of the 3×3 homography matrix can be computed at each time step and ‘stacked’ into a 9×1 column vector to be used as a measurement in an EKF.

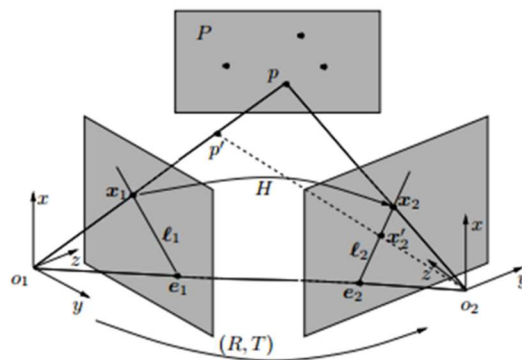


Figure 34: Homography Relationship
(Ma, Kosecka, Soatto, & Sastry, 2001)

We note that the homography matrix contains only information about the relative pose, R and T , between the two frames, and no absolute GPS-like information about the location in the environment. This is typical of non a priori algorithms, but is the price to pay for their versatility and robustness. However, with a hybrid navigation method, we can get the best of both a priori and non a priori algorithms.

On its own, the homography algorithm drifts in both position and heading over time

as shown in Figure 35, which gives the simulation results from the homography measurement used on its own. This is expected since the homography measurement does not provide absolute position or heading information. Nonetheless, the position drift is considerably slower than the dead-reckoning integration of the IMU, and so it represents a significant contribution to the estimation even in terms of position. Additionally, the homography gives reliable estimation of the velocity and attitude (roll and pitch), since these states are observed by the information in the homography matrix. This is a desirable feature since velocity and attitude are important for the stability of a controller. There are noticeable velocity errors due to the abruptness of the simulation maneuvers. However, as mentioned previously, such dramatic maneuvering is not realistic and can be disregarded.

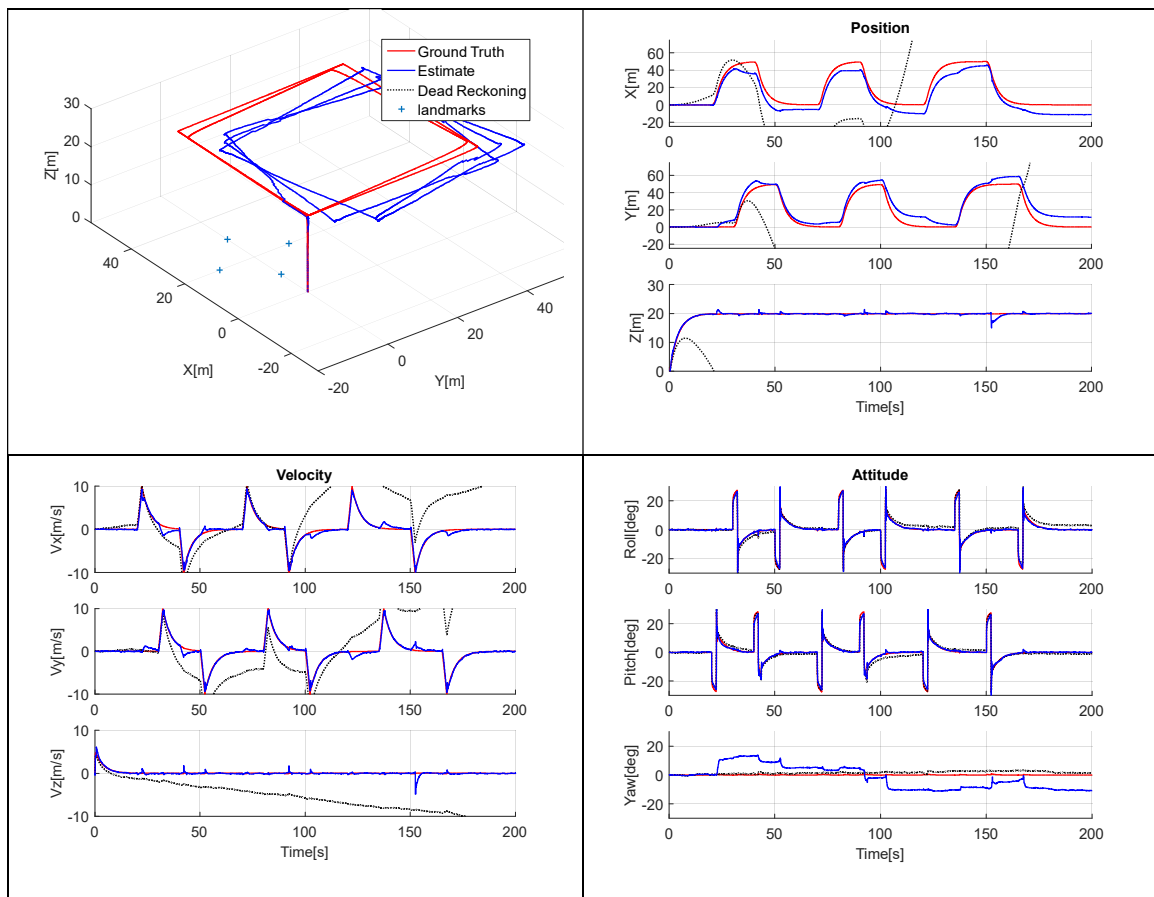


Figure 35: Homography measurement simulation

We note that the four landmarks shown in Figure 35 are not the only feature points observed by the homography, which uses many other randomly distributed points on the ground ($z = 0$) that are not shown, so that the homography matrix can always be computed. Thus, the homography measurement does not suffer from the absence of landmarks since the feature points only need to be detected between frames and no a priori information about them or their locations is needed. As a result, the homography estimation shows no indications of altered behavior when the four landmarks shown are out of view during the box-shaped flight pattern.

6.2. Hybrid Results

From Chapter 3.4, we recall that the landmark navigation method showed potentially unstable behavior for this box-route since that the landmarks upon which it relies are repeatedly popping in and out of view. See Figure 4, for example, which exhibits the more stable underweighted behavior, but still makes repeated corrections each time the landmarks re-enter the field of view. If we use the homography measurement in tandem with the landmark measurement so that the position estimation drifts only slightly while the landmarks are out of view, these corrections will not need to be as dramatic.

The marriage of the landmark and homography methods can be performed in two ways. First, since both algorithms take the form of the EKF, a simple matrix concatenation of the measurements and measurement Jacobians will suffice. This is the method employed here; however, the coincidence of both methods as EKFs is not universal to all algorithms. In such cases, the algorithms can be run separately, and then fused by an additional ‘integration EKF’ as described in (Groves, 2008), for example.

Figure 36 shows a simulation of the hybrid landmark-homography navigation

algorithm, which exhibits better performance than either the landmark or homography alone. The landmark method prevents the position and heading drift by providing the GPS-like measurement during the portions of the flight when the landmarks are visible, while the homography stabilizes the estimation during periods when landmarks are not visible. The overall behavior of the state estimation is both accurate and stable.

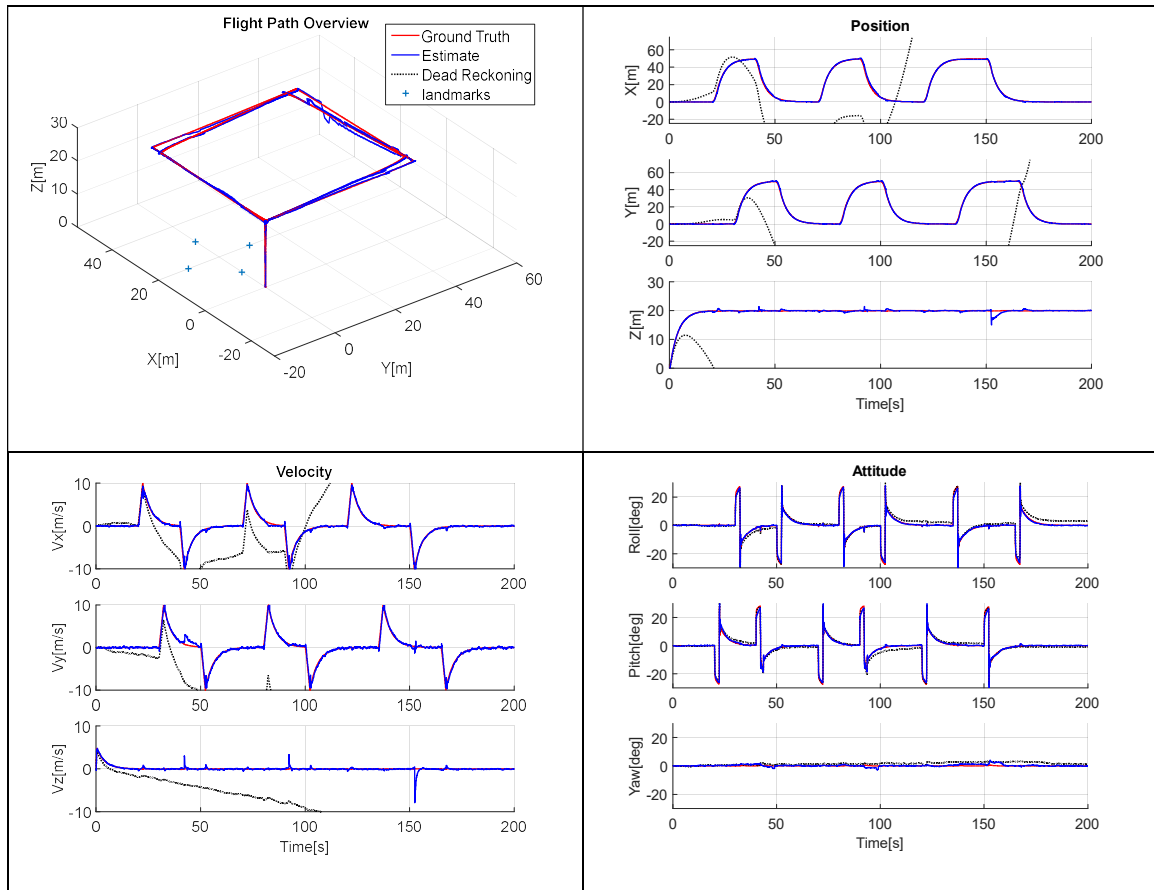


Figure 36: Hybrid navigation method simulation

Due to its robust nature, the hybrid algorithm performs equally well when applied in post-processing to the flight test data. Figure 37 shows the hybrid navigation post-processing results, which are both stable and drift-free for a flight path where landmarks move in and out of view just as in the simulation. Apparent errors do occur during portions of the flight path; however, these occur predominantly during periods in which the landmarks are not visible, and we must also recall as before that the ‘ground truth’ estimate

is in fact an IMU-GPS measurement with errors of its own. The key observation here is that the hybrid estimation performance is comparable to that of the GPS-IMU solution from the onboard Pixhawk flight computer, and is thus viable for online use in future research.

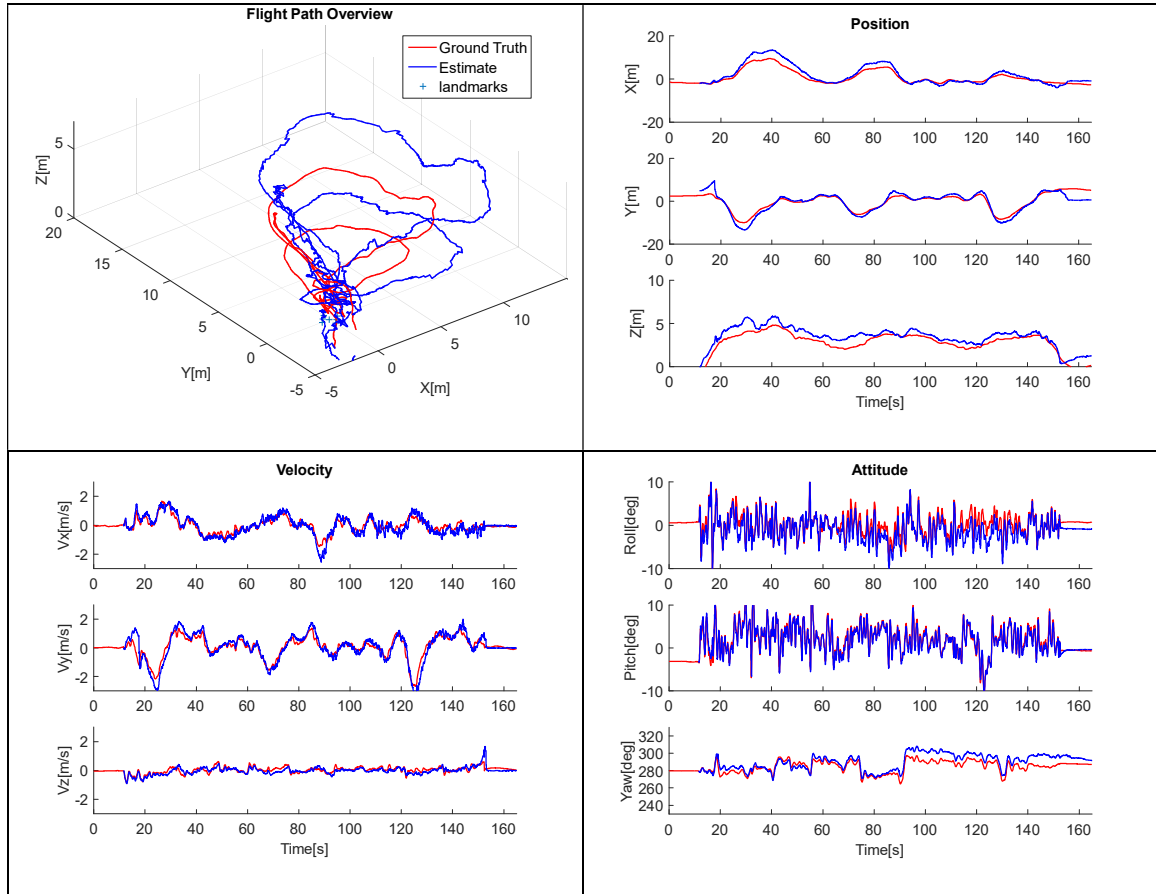


Figure 37: Hybrid post-processing estimation

7. Conclusion

We have developed, analyzed, and tested a landmark-based navigation algorithm for generic landmarks based on the well-known EKF framework. The algorithm utilizes a priori information about a select few ‘landmarks’ in the environment. These landmarks must have known or definable (as in the case of a landing, target for example) position, and be uniquely identifiable. We note again that the detection of the ‘landmarks’ does not need to be visual in nature, though vision measurements are considered exclusively here due to their practicality. The landmark measurement provides a GPS-like estimation correction due to the absolute information it provides through the a priori information.

Loss and reacquisition of landmarks can lead to instability due to measurement nonlinearities, particularly during re-convergence. These instabilities can be dealt with by a nonlinear filter such as a second order EKF, or the UKF algorithm. Underweighting, however, provides a simpler solution to this particular problem, and is used here. If other nonlinear stability issues are present in a particular case, more significant changes to the filter framework may be required.

An analysis of the state observability provided by a landmark measurement shows promise even with a single landmark in view, and improves to full observability with just two landmarks. Stereo vision provides some improvement in terms of removing landmark-directed unobservability, but azimuthal unobservability remains. Likewise, there are some curious nonlinear effects that enable potential landmark-directed observability even with a single landmark and a monocular measurement, provided that the vehicle has a motion component perpendicular to the target direction by effectively using the frame-to-frame parallax. Such effects may be worthy of investigation in future research.

Simulation testing confirms the conclusions from the observability analysis, and demonstrates the GPS-like state correction that the landmark algorithm provides. Flight test post-processing estimation performs similarly, such that further testing with a more stable test setup and more accurate sensors is warranted.

Lastly, the landmark measurement complements the non a priori homography measurement in a hybrid algorithm that is both drift free due to the GPS-like landmark information and stable during periods with no landmarks due to the non a priori homography. Research into applications of the hybrid algorithm is ongoing.

8. Recommendations

There is ample of room for future work on landmark navigation. First, a proof of the stabilizing ‘observability’ provided by the nonlinear nature of the measurement during translation could prove valuable. In Chapter 3.5.4, we showed that the nonlinear correction was driven by the component of translation perpendicular to the direction to the landmark, but we did not formally show that the ‘correction’ is necessarily a stabilizing one as observed in the simulation. This will be investigated in future work.

Additionally, the simple color detection centroiding algorithm used here is inefficient and introduces unnecessary errors to the flight data. A more refined approach to the detection process could enable testing of some of the more interesting simulation scenarios such as single landmark or stereo. The detector can even potentially be trained to recognize certain, more complex objects in the scene using approaches such as the Viola-Jones learning algorithm (Viola & Jones, 2001).

While both the Monocular and Stereo measurements were analyzed for observability, and tested in simulation, flight tests thus far have included only monocular measurements. Such tests are left to future work since they will require more precise equipment and detection algorithms.

Lastly, landmark navigation has promising applications alongside a non a priori algorithm in a hybrid setup. Examples include performing a landing at a predefined location, path following by using landmarks as waypoints, or even exploration missions with post-route mapping corrections upon return to a starting location.

References

- Aggarwal, J., & Nandhakumar, N. (1988). On the Computation of Motion from Sequences of Images - A Review. *Proceedings of the IEEE*, 917-935.
- Atiya, S., & Hager, G. D. (1993). Real-Time Vision Based-Robot Localization. *IEEE Transactions on Robotics and Automation*, 785-800.
- Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM Algorithm. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3562-3568).
- Bingol, H. E., Akin, B., & Koc, O. (2012). Radar altimeter as a navigation aid using hierarchical elevation map clustering. *IEEE/ION Position, Location and Navigation Symposium*, (pp. 377-381). Myrtle Beach.
- Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual Navigation for Mobile Robots: a Survey. *Journal of Intelligent and Robotic Systems*, 263–296.
- Brockers, R., Bouffard, P., Ma, J., Matthies, L., & Tomlin, C. (2011). Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision. *Micro- and Nanotechnology Sensors, Systems, and Applications III*, 12.
- Caballero, F., Merino, L., Ferruz, J., & Ollero, A. (2009). Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *Journal of Intelligent & Robotic Systems*.
- Chavez, A. S., Myhre, N. J., & Prazenica, R. J. (2017). Vision-Based State Estimation for Asteroid Exploration. *AIAA SPACE and Astronautics Forum and Exposition*. Orlando: AIAA SPACE Forum.

- Chavez, A. S., Prazenica, R., Myhre, N., Clark, M., L'Heureux, D., Prabhakar, N., & Law, W.-L. (2017). Homography-Based State Estimation for Autonomous UAV Landing. *AIAA Information Systems*. Grapevine.
- Conte, G., & Doherty, P. (2009). Vision-Based Unmanned Aerial Vehicle Navigation Using Geo-Referenced Information. *EURASIP Journal on Advances in Signal Processing*.
- Cox, H. (1964). On the Estimation of State Variables and Parameters for Noisy Dynamic Systems. *IEEE Transactions on Automatic Control*, (pp. 5-12).
- DeAngelo, M. P., & Horn, J. F. (2016). Landmark-Aided Localization for Air Vehicles Using Learned Object Detectors. *AIAA Guidance, Navigation, and Control Conference*. San Diego: AIAA.
- DeAngelo, M. P., & Horn, J. F. (2017). Aerial Vehicle Location Using Generic Landmarks. *AIAA Guidance, Navigation, and Control Conference*. Grapevine, TX: AIAA.
- Dronecode. (n.d.). *PX4 Developers Guide*. Retrieved from dronecode.org:
https://dev.px4.io/en/tutorials/tuning_the_ecl_ekf.html
- Friedland, B. (1969). Treatment of Bias in Recursive Filtering. *IEEE Transactions on Automatic Control*, 359-367.
- GoPro. (n.d.). *Product Manuals*. Retrieved from gopro.com:
<https://gopro.com/help/productmanuals>
- Groves, P. D. (2008). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston: Artechhouse.
- Hartikainen, J., Solin, A., & Sarkka, S. (2011). Optimal Filtering with Kalman Filters and

Smoothers: a Manual for the Matlab toolbox EKF/UKF. Aalto University Dept. of Biomedical Engineering and Computational Science.

Huang, T. S., & Netravali, A. N. (1994). Motion and Structure from Feature Correspondences: A Review. *Proceedings of the IEEE*, 252-268.

Ivey, G. F., & Johnson, E. (2006). Investigation of Methods for Simultaneous Localization and Mapping Using Vision Sensors. *AIAA Guidance, Navigation, and Control*. Keystone.

Johnson, A. E., & Montgomery, J. F. (2008). Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing. *IEEE Aerospace Conference*, 1-8.

Julier, S. J., & Uhlmann, J. K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems. *Proceedings of SPIE 3068*.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 35-45.

Kanade, T., Amidi, O., & Ke, Q. (2004). Real-Time and 3D Vision for Autonomous Small and Micro Air Vehicles. *IEEE Conference on Decision and Control*. Nassau.

Kehoe, J. J., Causey, R. S., Arvai, A., & Lind, R. (2006). Partial Aircraft State Estimation from Optical Flow Using Non-Model-Based Optimization. *American Control Conference*. Minneapolis.

Lange, S., Sunderhauf, N., & Protzel, P. (2009). A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments. *International Conference on Advanced Robotics*. Munich.

Lear, W. M. (1973). *Multi-Phase Navigation Program for the Space Shuttle Orbiter*.

NASA Internal Note 73-FM-132.

- Lee, D., Kim, Y., & Bang, H. (2014). Vision-aided terrain referenced navigation for unmanned aerial vehicles using ground features. *Journal of Aerospace Engineering*, 228(13), 2399-2413. doi:10.1177/0954410013517804
- Lee, D., Kim, Y., & Bang, H. (2013). Vision-based Terrain Referenced Navigation for Unmanned Aerial Vehicles using Homography Relationship. *Journal of Intelligent Robot Systems*, 69, 489-497. doi:10.1007/s10846-012-9750-1
- Lemaire, T., Lacroix, S., & Sola, J. (2005). A practical 3D Bearing-Only SLAM algorithm. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 2449-2454).
- Li, P., Garratt, M., & Lambert, A. (2014). Monocular Snapshot-Based Sensing and Control of Hover, Takeoff, and Landing for a Low-Cost Quadrotor. *Journal of Field Robotics*, 984-1003.
- Ljung, L. (1979). Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems. *IEEE Transactions on Automatic Control*, 36-50.
- Lucas, B. D., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of Imaging Understanding Workshop*, 121-130.
- Ma, Y., Kosecka, J., Soatto, S., & Sastry, S. (2001). *An Invitation to 3D Vision: From Images to Models*. New York, NY: Springer.
- Maimone, M., Cheng, Y., & Matthies, L. (2007). Two Years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 169-186.
- Masselli, A., & Zell, A. (2012). A Novel Marker Based Tracking Method for Position

and Attitude Control of MAVs. *International Micro Air Vehicle Conference*. Braunschweig, Germany.

Mondragón, I. F., Campoy, P., Martínez, C., & Olivares-Méndez, M. A. (2010). 3D pose estimation based on planar object tracking for UAVs control. *Robotics and Automation (ICRA)*. Anchorage: IEEE. doi:10.1109/ROBOT.2010.5509287

Perez, A. E., Moncayo, H., Prazenica, R. J., Zacny, C., Mueller, R. P., Ebert, T., & DuPuis, M. (2016). Control Laws Development for a Free-Flying Unmanned Robotic System to Support Interplanetary Bodies Prospecting and Characterization Missions. *AIAA Guidance, Navigation, and Control Conference*. San Diego, CA.

Prazenica, R. J., Kurdila, R. C., Binev, P., Hielsberg, M. H., Lane, J., & Evers, J. (2006, June 14). Vision-Based Receding Horizon Control for Micro Air Vehicles in Urban Environments. *Proceedings of the 2006 American Control Conference*, 6.

Prazenica, R. J., Hielsberg, M., Sharpley, R. C., & Kurdila, A. J. (2013). 3-D Implicit Terrain Mapping and Path Planning for Autonomous MAV Flight in Urban Environments. *AIAA Guidance, Navigation and Control Conference*. Boston.

Prazenica, R. J., Kern, Z., John, T. S., Moncayo, H., Zacny, K., Mueller, R. P., . . . DuPuis, M. (2016). Vision-Aided Navigation for a Free-Flying Unmanned Robotic System to Support Interplanetary Bodies Prospecting and Characterization Missions. *AIAA Guidance, Navigation, and Control Conference*. San Diego, CA.

Ribeiro, M. I. (2004). *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*. Lisboa, PORTUGAL: Institute for Systems and Robotics.

- Smith, R., Self, M., & Cheeseman, P. (1987). A Stochastic Map For Uncertain Spatial Relationships. *International Symposium of Robotics Research*, (pp. 467-474).
- Strydom, R., Denuelle, A., & Srinivasan, M. V. (2016). Bio-Inspired Principles Applied to the Guidance, Navigation and Control of UAS. *Aerospace*.
- U-Blox. (2009, July 31). *LEA-6 Series*. Retrieved from [www.u-blox.com: https://www.u-blox.com/en/product/lea-6-series](https://www.u-blox.com/en/product/lea-6-series)
- Viola, P., & Jones, M. J. (2001). *Robust Real-Time Object Detection*. Cambridge Research Laboratory Technical Report Series.
- Webb, T. P., & Prazhenica, R. J. (2007). Vision-Based State Estimation for Autonomous Micro Air Vehicles. *Journal of Guidance, Control, and Dynamics*, 816-826.
- Wenzel, C. E., Rosset, P., & Zell, A. (2010). Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller. *Journal of Intelligent and Robotic Systems*, 297-311.
- Yang, S., Scherer, S. A., & Zell, A. (2013). An Onboard Monocular Vision System for Autonomous Takeoff, Hovering, and Landing of a Micro-Air Vehicle. *Journal of Intelligent and Robotic Systems*, 499-515.
- Zanetti, R., DeMars, K. J., & Bishop, R. H. (2010). Underweighting Nonlinear Measurements. *Journal of Guidance, Control, and Dynamics*, 1670-1675.
- Zhao, S., Dong, X., Cui, J., Ang, Z. Y., Fend, L., Peng, K., . . . Lee, T. H. (n.d.). Design and Implementation of Homography-based Vision-aided Inertial Navigation of UAVs. *Proceedings of the 32nd Chinese Control Conference*.
- Zhao, S., Lin, F., Peng, K., Chen, B. M., & Lee, T. H. (2012, August). Homography-based vision-aided inertial navigation of UAVs in unknown environments. *AIAA*

Guidance, Navigation, and Control Conference.

Zhao, S., Liny, F., Pengy, K., Chenz, B. M., & Leez, T. H. (2012). Homography-based

Vision-aided Inertial Navigation of UAVs in Unknown Environments. *AIAA*

Guidance, Navigation, and Control Conference. Minneapolis, Minnesota.

Zhao, S., Liny, F., Pengy, K., Dong, X., Chen, B. M., & Lee, T. H. (2016). Vision-aided

Estimation of Attitude, Velocity, and Inertial Measurement Bias for UAV

Stabilization. *Journal of Intelligent & Robotic Systems*, 531-549.