

**Deep Integration of a Flight Vehicle Dynamic Model in a Vector Tracking Software
Defined Receiver**

by

Noah S. Miller

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 2024

Keywords: Simulation, GNSS/INS Integration, Aircraft Modeling

Copyright 2024 by Noah S. Miller

Approved by

Scott M. Martin, Assistant Professor of Mechanical Engineering
David M. Bevly, Bill and Lana McNair Distinguished Professor of Mechanical Engineering
Chad G. Rose, Assistant Professor of Mechanical Engineering

Abstract

The greater occurrence of signal interference on Global Navigation Satellite Systems (GNSS) requires additional alternative navigation solutions to provide robust, reliable localization of flight vehicles when measurements from GNSS are unavailable. This thesis proposes a Flight Vehicle Dynamic Model (FVDM) that is deeply integrated with Global Positioning System (GPS) correlator measurements for aircraft state estimates in GPS-challenged environments.

It is well-documented that large, modern aircraft feature an array of sensors that work in tandem to provide robust positioning performance. However, the sizing limitations of low Size, Weight, Power, and Cost (SWaP-C) do not allow for such redundant sensor suites, meaning an alternative navigation solution is required. Furthermore, low cost flight vehicles typically feature lower quality sensors that are subject to vibrations and subsequently faulty, unreliable measurements that provide no benefit to the flight vehicle when GNSS measurements are also considered unreliable.

The FVDM is a high-fidelity flight vehicle model based on the Diamond DA-40 single-propeller fixed wing aircraft. The aircraft model features a piston engine model that generates thrust power through a shaft that spins a numerically modeled 3-blade propeller. The speed of the propeller is controlled through a electric governor, and the pitch is controlled to maintain efficient propeller action onto the incident airflow. The aerodynamics of the aircraft are modeled using a discretized aerodynamic coefficient technique, also known as strip theory. Although not used in this work, a landing gear model incorporates the three landing gear on the Diamond DA-40 and evaluates the forces and moments applied during landing as a second-order spring-mass-damper system. An International Standard Atmosphere (ISA) model is used to calculate the density, temperature, and ambient pressure based on aircraft altitude. To close the loop of the FVDM, a set of controllers in collaboration with a waypoint manager are used to actuate the control surfaces on the aircraft. Multiple planned paths are demonstrated during this work and are presented in their respective sections.

The proposed navigation filter presented in this work provides a closed loop solution fusion of the vector tracking loop algorithms and the FVDM process model via a Vector Delay and Frequency Lock Loop (VDFLL). Vector tracking loops are able to maintain channel lock on satellite signals when either signal interference is present, or the dynamics of the collection platform are too high for scalar loops to track the signal consistently.

The results within this work showcase the improvements in flight vehicle state estimates when compared to a standard VDFLL zero-mean acceleration kinematic model. In simulation, two trajectories are flown under varying levels of signal degradation. The first trajectory is a steady-level, un-accelerated flight path where the aircraft is maintaining a constant altitude and heading for the duration of the 60 second simulation. For this trajectory, it is expected that the standard VDFLL implementation performs comparably to the proposed navigation filter. The second trajectory features a more dramatic flight bath – full of oscillatory turns and a constant climb segments. For the second trajectory, the proposed navigation filter out performs the standard VDFLL due to its capability to predict the behavior of the aircraft given a set of control inputs. Each of the trajectories and subsequent cases of signal degradation are tested in 100-run Monte-Carlo sims to further test the robustness of the proposed navigation filter.

Acknowledgments

Graduate school is not a walk in the park and can be more closely compared to a game of chutes and ladders. Often times, one encounters chutes or setbacks that forces them strengthen their foundations, locate the nearest ladder, and stumble closer to the finish line. Luckily, this windy path is not traversed alone. During my time in graduate school, several ladders come to mind, helping me to become the scholar and man I am today. The first one is David Bevly, who offered me an opportunity to join the GPS and Vehicle Dynamics Lab despite my low-undergraduate GPA (2.96) and no GRE score. The Zoom call we had on 12/16/2021 changed my life for the better by giving me the direction and purpose I was wanting as I completed my Master's degree. The second ladder that comes to mind is Scott Martin, whose guidance and hands-off approach allowed me to hone my leadership abilities and grow into my own space as an academic. My time and accomplishments in the GPS and Vehicle Dynamics Lab could not be possible without the two biggest ladders in my life, my parents. My father, whose 25 years of service in the Army taught me leadership, courage, and love for country. My mother, whose empathetic spirit taught me how to care for others and the value of education. Not all ladders can hold steady when climbing the most precarious of paths, but these two, I know I can count on. Also influential in my completion of my graduate degree are the students in GAVLAB. Thank you for the paper revisions and ping pong matches and basketball games after long days in the lab. Lastly, I would like to thank God for providing me with guidance in life.

Table of Contents

Abstract	ii
Acknowledgments	iv
List of Abbreviations	xiii
1 Introduction and Background	1
1.1 Prior Art	3
1.1.1 Sensor Fusion Overview and Variation	3
1.1.2 Vehicle Dynamic Model Sensor Fusion	4
1.1.3 Other Types of Flight Vehicle Sensor Fusion	5
1.2 Vector Tracking	5
1.3 Field Contributions	6
1.4 Thesis Outline	7
2 Flight Vehicle Dynamics Model	8
2.1 Reference Frames	8
2.1.1 Flight Vehicle Reference Frame	9
2.1.2 Propeller Reference Frame	11
2.1.3 Local Wind Reference Frame	11
2.1.4 Local Navigation Reference Frame	12
2.1.5 Global Reference Frames	12
2.2 Reference Frame Transformations	14

2.2.1	ECEF to LLA	14
2.2.2	LLA to Local	15
2.2.3	Local to Body	16
2.3	Atmosphere Model	16
2.4	Aerodynamic Principles	17
2.5	Aero-Propulsive Forces	22
2.5.1	Piston Engine Model	23
2.5.2	Propeller Modeling	25
2.6	Landing Gear Model	30
2.7	Forces and Moments Calculations	32
2.8	Measures of Fidelity	32
2.9	Conclusions	33
3	GPS Software Defined Receiver Architecture	34
3.1	Front End	34
3.2	Acquisition	35
3.3	Tracking	37
3.3.1	Correlators	37
3.3.2	Discriminators and Loop Filters	39
3.4	Navigation Algorithms	41
3.4.1	GPS Measurements	42
3.4.2	Open-Loop Architectures	45
3.5	Conclusions	47
4	Proposed Navigation Filter Architecture	49
4.1	Vector Delay and Frequency Lock Loop	51
4.2	Deeply Coupled GPS and FVDM Navigation Filter	52

4.2.1	Update <i>a posteriori</i>	58
4.3	Conclusions	60
5	Simulation Environment	61
5.1	Guidance System	61
5.1.1	Waypoint Generation	61
5.2	Control Scheme	63
5.3	Disturbance Modeling	65
5.4	Conclusions	65
6	Scenario Implementation and Results	67
6.1	Monte-Carlo Analyses	67
6.2	First Trajectory	68
6.2.1	Monte-Carlo Analyses	71
6.3	Second Trajectory	77
6.3.1	Monte-Carlo Analyses	78
6.4	Conclusions	85
7	Conclusions and Future Work	86
7.1	Concerns of Observability	87
7.2	Future Work	88
Appendices		99
Appendix A – Extra Results from Trajectory Two		99

List of Figures

1.1	Aircraft with a larger Max Takeoff Operating Weight (MTOW) typically have more space for more complex sensor suites compared to their General Aviation counterparts [5].	1
2.1	Pilot flying Diamond DA-40 single propeller aircraft – the focus of the collection platform for this thesis [6].	9
2.2	Standard flight vehicle reference frame used in this work. [24]	10
2.3	The reference frame used to model the dynamic of the propeller in this work [25].	11
2.4	Flight vehicle reference frame and local wind reference frame (dotted lines) (Adapted from [26]).	12
2.5	ECEF and geodetic reference frames as used in this thesis.	13
2.6	Absolute temperature, ambient pressure, air density, and speed of sound using the ISA model.	18
2.7	Subsonic wind tunnel in Auburn University’s Aerodynamics Laboratory.	18
2.8	Pressure gradient and subsequent lift produced onto an airfoil. The continuous white line in the middle signifies the freestream incident wind [29].	19
2.9	Horizontal and vertical strips that are analyzed for aerodynamic coefficients in FlightStream [32].	22
2.10	Power output for modeled engine as a function of aircraft altitude.	24
2.11	Power factor look up table for modeled engine.	24
2.12	Brake specific fuel consumption look up table for modeled engine.	25
2.13	Collection of historic wood and steel propellers [37].	26
2.14	Illustration of the variable pitch propeller [39].	26
2.15	Static propeller thrust for the modelled propeller (Adapted from [41]).	28
2.16	Mass-spring damper system, representing the components of landing gear on the aircraft (adapted from [42]).	31

2.17	Identification of the landing gear components on the Diamond DA40.	31
3.1	Block diagram of a software defined receiver front end.	35
3.2	Block diagram of the PCPS acquisition algorithm applied to GPS L1 C/A signal modulation.	36
3.3	Successful acquisition of GPS SV:7 using the PCPS algorithm.	37
3.4	Flow diagram of scalar tracking algorithms.	38
3.5	Discrete block diagram of the PLL-FLL loop filters used in this work [44].	40
3.6	Discrete block diagram of the DLL loop filter used in this work [44].	41
3.7	Tracked navigation message for GPS SV:7 using the ascribed methods.	42
3.8	Static position of receiver using a WNLS navigation algorithm.	48
4.1	Loosely-coupled architecture between GPS and INS systems [52].	50
4.2	Tightly-coupled architecture between GPS and INS systems [52].	51
4.3	Block diagram of the VDFLL used in this work (Adapted from [9]).	52
5.1	A custom trajectory created in Google Earth for the guidance system discussed in this work.	62
5.2	Block diagram of lateral stick control through two closed-loop, PID controllers.	63
5.3	Block diagram of longitudinal stick control through two closed-loop, PID controllers.	64
6.1	(Left) Top-view of simulated flight path for trajectory one. (Right) Altitude of flight path for trajectory one.	69
6.2	Orange dots signify satellite locations at the start of the simulations given the date of broadcast ephemeris. Black dots signify satellites that are in-view but are discarded due to the 10 degree mask angle used. The green triangle represents the initial receiver position.	70
6.3	Code phase and carrier frequency RMSE as a function of signal power, specified in dB-Hz.	72
6.4	The probability that each navigation filter is able to maintain channel lock throughout the simulation across different levels of signal interference.	73

6.5	Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	73
6.6	Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory when subject to a degraded signal power of 16 dB-Hz.	74
6.7	Average altitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	74
6.8	Average speed estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	75
6.9	Average clock bias estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	76
6.10	Average clock drift estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	76
6.11	(Left) Top-view of simulated flight path for the second trajectory. (Right) Altitude of second flight path where aircraft is commanded to climb to 1150 meters and then maintain the altitude for the remainder of the simulation.	78
6.12	Code phase and carrier frequency RMSE as a function of signal power, specified in dB-Hz.	79
6.13	The probability that each navigation filter is able to maintain channel lock throughout the simulation across different levels of signal interference.	79
6.14	Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	80
6.15	Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory when subject to a degraded signal power of 16 dB-Hz.	80
6.16	Average altitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	81

6.17	Average speed estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	82
6.18	Average angular rate estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 15 dB-Hz. The right figure is with a signal power of 20 dB-Hz.	82
6.19	Average Euler attitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 15 dB-Hz. The right figure is with a signal power of 20 dB-Hz.	83
6.20	Average clock bias estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	84
6.21	Average clock drift estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.	84
7.1	Cascaded architecture for an improved FVDM coupling.	89
2	Latitude and longitude of the proposed navigation filter compared to the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.	99
3	Altitude errors of the proposed navigation filter and the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.	100
4	Aircraft speed estimate from the proposed navigation filter compared to the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.	100
5	Velocity components of the proposed navigation filter compared with the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.	101
6	Angular rate estimates of the proposed navigation filter when receiving no GPS measurements for trajectory two.	102
7	Euler attitude estimates of the proposed navigation filter when receiving no GPS measurements for trajectory two.	103

List of Tables

1.1	Inexhaustive list of sensors available to commercial and general aviation aircraft. (Adapted from [6, 7])	2
1.2	Common sensor fusion frameworks used for GPS and INS collection platforms	2
2.1	Properties describing the WGS84 ellipsoid	13
2.2	Characteristics of the propeller modeled for this work.	27
2.3	List of spring and damper coefficients for nose and rear landing gear.	31
5.1	List of available, controllable inputs to the Diamond DA40 modeled in this thesis.	63
5.2	Gains for the throttle and propeller lever closed-loop PID controllers.	64
6.1	Signal power for each case applied to each trajectory.	69
6.2	Initial conditions for simulated trajectory from Figure 6.1.	70
6.3	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the deeply-coupled FVDM is subject to a degraded signal power level of 35 dB-Hz.	72
6.4	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the standard VT receiver is subject to a degraded signal power level of 35 dB-Hz.	72
6.5	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.	77
6.6	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.	77
6.7	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.	85
6.8	RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.	85

List of Abbreviations

ADC	Analog-to-Digital Converter
BPF	Band Pass Filters
CFD	Computational Fluid Dynamics
DCM	Direction Cosine Matrix
DOF	Degrees of Freedom
ECEF	Earth-Centered, Earth-Fixed
EKF	Extended Kalman Filter
FFT	Fast-Fourier Transform
FLL	Frequency Lock Loop
FVDM	Flight Vehicle Dynamic Model
GA	General Aviation
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GVDM	Ground Vehicle Dynamic Model
ISA	International Standard Atmosphere
LLA	Longitude, Latitude, and Altitude

LNA Low Noise Amplifiers

MSL Mean Sea Level

MTOW Max Takeoff Operating Weight

OEM Original Equipment Manufacturer

PCPS Parallel Code Phase Search

PLL Phase Lock Loops

PRN Pseudo-Random Number

RHCP Right-Hand Circularly Polarized

TOW Time of Week

UKF Unscented Kalman Filter

UWB Ultra-Wide Band

VDLL Vector Delay Lock Loop

VFDDL Vector Frequency and Delay Lock Loop

WGS84 World Geodetic System 1984

WNLS Weighted Non-Linear Least Squares

Chapter 1

Introduction and Background

Redundant, accurate flight vehicle localization has been well-documented over several decades [1, 2, 3, 4]. However, as the threat to GPS signals continues to rise, the need for robust pose estimation increases. While robust systems already exist to combat incoming interference, these systems are either proprietary or government controlled, making them infeasible for widespread use. Cheaper, robust systems are critical for the safe future of civilian and military flight vehicles. Commercial aircraft feature a wide sensor suite that works in tandem to provide redundancy and safety critical features to maintain safe flight. These aircraft have the space to install robust sensor suites and the power to run them consistently in all phases of flight (Figure 1.1), more importantly, the companies that design and build these aircraft also have the budget to afford such expensive sensor suites.

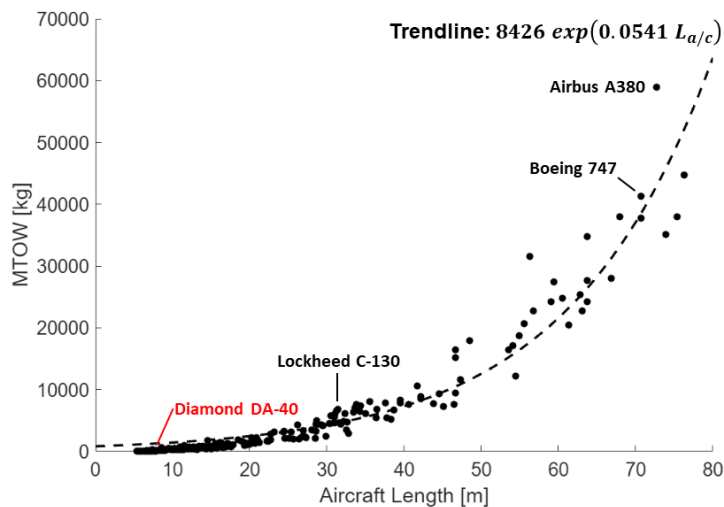


Figure 1.1: Aircraft with a larger Max Takeoff Operating Weight (MTOW) typically have more space for more complex sensor suites compared to their General Aviation counterparts [5].

Smaller aircraft are not allowed these luxuries, being manufactured with fewer sensors, overall making them less safe. Table 1.1 compares the different sensors onboard a commercial airliner and a civilian general aviation aircraft.

Table 1.1: Inexhaustive list of sensors available to commercial and general aviation aircraft. (Adapted from [6, 7])

Sensor	Commercial	General Aviation
Pitot Tubes	5	2
Distance Measuring Equipment	2	1
Ultra-High Frequency Sensors	2	1
Very-High Frequency Sensors	3	2
Communication Channels	2	2
Outside Air Temperature Sensors	4	1
Fuel Flow Gauge	4	1
GPS receivers	2	1
Inertial Measurement Units	3	1
Satellite Communications	1	–
Specific Impulse Sensors	6	–
Weather Radar	1	–
Traffic Collision Avoidance System	4	–
Radio Altimeter	3	–

Regardless of components, the sensor suite aboard any flight vehicle is able to provide measurements more often than that of a GPS receiver, so a sensor fusion navigation algorithm is optimal. Because sensor measurements have inherit errors due to a variety of factors, they can wander or drift over time. GPS measurements, although slower, provide measurements that do not drift at the cost of being noisier. Table 1.2 describes the most common sensor fusion frameworks for GPS and INS platforms. More details on each these frameworks can be found in Chapter 4.

Table 1.2: Common sensor fusion frameworks used for GPS and INS collection platforms

Name	Level of Measurement	Advantages
<i>Loosely-Coupled</i>	Position and Velocity	Simplicity, Redundancy
<i>Tightly-Coupled</i>	Pseudorange and Doppler	Observability, Robustness
<i>Deeply-Coupled</i>	Inphase and Quadrature Correlators	Robustness, Optimal Architecture

This work presents a deeply-coupled sensor fusion algorithm tied into a Vector Tracking (VT) Software Defined Receiver (SDR). Of the three types of GPS and INS sensor fusion

frameworks, it is well-documented that deep-coupling provides higher resistance to signal interference and can operate with less than four satellites for a limited time [8]. The benefits of the closed-loop navigation framework (vector tracking) enables a coupling of measurements between satellites. This provides stronger tracking performance in GPS-challenged environments where the signal to noise ratio is lower due to interference [9].

The proposed navigation filter is not considered a full-replacement of existing sensor coupling solutions, but instead can be utilized as a short-term, reliable source of pose and attitude estimation in scenarios of GPS degradation, or when the aforementioned sensors are providing faulty measurements.

1.1 Prior Art

The objective of this thesis is to investigate the efficacy of flight vehicle state estimation in GPS-degraded conditions by utilizing a deeply-coupled FVDM. This thesis will analyze the performance through use of a deeply-coupled sensor fusion algorithm utilizing the state propagation of a FVDM and GPS correlator measurements. The idea of fusing a vehicle dynamic model and GPS measurements together on a flight vehicle is a recent development, tied with increased computer performance. The following subsections highlight the work performed by other authors and their contributions to the field.

1.1.1 Sensor Fusion Overview and Variation

Sensor fusion between GPS and other sensors aboard flight vehicles has existed for years and continues to be developed. A precise, accurate, and robust navigation solution is achievable when redundant, expensive, and high-quality sensor are installed. More information about the sensors found aboard commercial aircraft and smaller aircraft can be found in [10] and [6], respectively. Salmon [11] provides an exploratory analysis using ground vehicles and their sensors complimented with a Ground Vehicle Dynamic Model (GVDM) in a tightly-coupled GPS/INS/GVDM sensor fusion framework. The work of Rhudy [12] is similar to the research presented in this thesis as they present use of known controller inputs into a propriety nonlinear flight vehicle model to perform GPS sensor fusion with low-cost IMUs. The following section

explores the work of previous authors and the advances of sensor fusion between GNSS measurements and vehicle dynamic models.

1.1.2 Vehicle Dynamic Model Sensor Fusion

One of the more common ways to simulate and model a flight vehicle is to use known aerodynamic coefficients tied in with classical flight dynamics to propagate the states in time. The NAVION aircraft model is a nonlinear mathematical aircraft model that uses known aerodynamic coefficients from a single propeller fixed wing Navion aircraft [13]. Rhudy [12] uses the NAVION model included in a loosely-coupled sensor fusion framework with IMU and GPS measurements. Rhudy [12] focuses on using a known aircraft model with incoming piloted control inputs to predict the attitude of the aircraft in time. Khaghani and Skaloud [14] simulate a UAV given, known initial aerodynamic coefficients but feature the coefficients in their loosely-coupled framework to continuously estimate these variables in their vehicle model. Online estimation of the aerodynamic coefficients allows the model to be generalizable, further allowing it to be used on a wide array of aircraft with a similar configuration. Both papers fuse the dynamic model with GPS and IMU measurements at the position and velocity level (loosely-coupled). [12] uses an Unscented Kalman Filter (UKF) in the navigation filter for *ease of implementation* while Khaghani [14, 15] uses a 47-state Extended Kalman Filter (EKF). Both Rhudy and Khaghani use simulation environments to perform a Monte-Carlo analysis. Each IMU measurement is modeled with a constant bias and integrated white noise as a first-order Gauss-Markov process with numbers that mirror a MEMs quality IMU. Both [14] and [12] models GPS measurement errors as white noise with a variance of 1 meter in the North, East, and Down directions. Three years later, Khaghani and Skaloud [15] improve the original filter implementation by adding a barometer sensor to the measurement update along with the already standing IMU and GPS measurements. From their 2016 publication [14], the Monte-Carlo analysis showed unbounded error growth once the GPS outage started, leading to growing errors in aerodynamic coefficient estimates. The additional measurements provided by the barometer aid the overall position solution by providing observability in the altitude estimates. In 2020, [16] innovated upon the 2016 work of Khaghani by coupling the FVDM, IMU, and

GPS measurements within a tightly coupled navigation filter. The focus of this work was the performance of the state estimates in scenarios where the receiver had sparse satellite visibility of two and three satellites for a short period. [16] further proved the efficacy of utilizing a FVDM within a sensor fusion framework by conduction live-sky experiments via a modified remote controlled aircraft.

1.1.3 Other Types of Flight Vehicle Sensor Fusion

Other research to localize flight vehicles in GPS-degraded environments revolves around fusing together multiple sensors including Ultra Wide Band (UWB) radios, LiDAR and vision-based algorithms. Dong [17] integrates IMU, UWB, and computer vision for the autonomous approach and landing of a small UAV onto a moving platform. The vision framework operates on identifying and tracking independent ArUco markers to calculate the size of the landing zone. The IMU and UWBs work in tandem in estimating the states of the UAV and distance to the landing platform. The UWB radio provide updates at a high frequency but suffer from noise and dropouts when the UAV is farther away from the landing platform [17]. Gróf [18] develops a similar algorithm for detecting a runway using a down-view monocular camera, barometer readings, IMU, and previously recorded air data. Because of the camera, both of the aforementioned papers provide bounded results for pose and attitude estimation. Also because of the camera, the system is more complex and expensive, two things avoided in this work.

1.2 Vector Tracking

Signal tracking is the process of forming navigation observables from a received signal. Typically, signal tracking revolves around feedback from the cross-correlation of a local replicated signal and the incoming, received signal. Based on the power of the cross-correlation, parameters of the locally replicated signal are adapted to match with the received signal, and subsequently can be decoded for information. For GNSS signals, this process is performed through scalar tracking algorithms (discussed in Chapter 3). Scalar tracking replicates each received signal independent of each other to extrude measurements without sharing any information to the other channels. From here, scalar tracking algorithms pass the navigation observables to

a navigation module to propagate satellite positions and, through trilateration, will determine the pose of the receiver. Given benign signal conditions, scalar tracking algorithms are efficient and can produce accurate PVT solutions; however, in scenarios of signal interference, scalar tracking loops can become unstable and lose lock on the signal they are receiving. Because of this, navigation observables are unable to be obtained, and no PVT solution can be found. Facing this problem head-on, Parkinson and Spilker [19] proposed a new tracking loop in which an EKF would be utilized instead of standard scalar tracking loops. This was the onset of the Vector Delay Lock Loop (VDLL) filter. For the VDLL, an EKF would be utilized to predict receiver position to replicate the estimated code delay on the replicated signal. This was proposed in 1996, and computers were still in their infancy compared to today's computers and as such, no extensive performance analysis of the VDLL was cited. Eight years later, Benson from MITRE [20], published one of the first key analyses of the VDLL compared to the scalar Delay Lock Loop (DLL) filter. As expected, it showed significant improvements in degraded signal environments. A few years later, the work of Lashley and Bevilacqua [21, 22, 23], focused heavily on the comparison of vector tracking loops and scalar tracking loops in different scenarios – mainly collection platforms subject to high-dynamic stress and weak satellite visibility. Overall, their work showed that vector tracking loops can provide a 2 to 6 dB improvement over standard scalar tracking architectures. Regardless of the great benefits discussed, vector tracking has caveats that discourage its use on commercial products. For one, it is still extremely computationally expensive to run on even today's modern computers. Furthermore, it relies on the receiver to know its position, which cannot always be guaranteed. Because of the fundamental nature of using an EKF, vector tracking also suffers from noise sharing, in which degraded channels can have a negative impact on benign channels. Despite these drawbacks, vector tracking is the best possible signal processing algorithm to use in scenarios of signal degradation, which is the focus of this thesis.

1.3 Field Contributions

The focus of the research presented in this thesis is the development of a deeply-integrated FVDM with GPS L1 C/A correlator-level measurements. An evaluation of pose and attitude

estimation performance of the proposed navigation filter in GPS-degraded and GPS denied environments is also provided. The following contributions to the field are made:

- Development of a flight vehicle model to use in the navigation algorithm considering complexity and computational performance.
- Comparative analysis of multiple flight scenarios reflecting realistic flight plans and GPS degradation.
- Analysis of deeply-coupled sensor fusion algorithm using the flight vehicle model and simulated GPS measurements to determine the efficacy of safe localization.

1.4 Thesis Outline

For several decades, GPS has served as the backbone for attaining a global position solution on a variety of collection platforms. This ubiquity is well-documented, making GPS a practical and realistic signal to simulate for the focus of this thesis. Mainly used as medium for United States government entities, including the military services, encourages foreign threats to develop the technology needed to disrupt the GPS signal via jamming or spoofing. These imminent threats catalyze the greater need for more research in this area on all collection platforms, including flight vehicles. This thesis investigates the performance of pose and attitude estimation of simulated flight vehicle in scenarios of GPS-degradation and GPS-denied environments. Chapter 1 divulges prior art related to the investigation, while chapters 2, 3, 4, and 5 describe the FVDM, scalar and vector tracking receiver architectures, along with adaptations made by the proposed navigation filter, respectively. Chapter 6 presents the results, including a Monte-Carlo analysis of the performance improvements compared to a standard zero-mean acceleration kinematic process model for a VDFLL. Chapter 7 concludes the work followed by a section of future work for interested parties.

Chapter 2

Flight Vehicle Dynamics Model

The proposed navigation filter can be divided into two main modules, the flight vehicle dynamics model and the vector tracking algorithm. This chapter covers the flight vehicle dynamic model and how it predicts the forces and moments acting onto the airframe. The aircraft modeled in this thesis is the Diamond DA-40 single propeller fixed wing aircraft (Figure 2.1). The simulation is a full six Degrees of Freedom (DOF) and features 12 control inputs – ranging from throttle and control surface inputs to propeller pitch and mixture levers. The sections below detail the flight mechanics modules that propagate the aerodynamics, engine and propeller, landing gear, and gravitational forces and moments during the simulation. Small General Aviation (GA) aircraft are easier to model compared to commercial and military aircraft as performance information for the mechanical components of the plane are not hidden through government or proprietary documents. Using publicly available data, numerical tables can be compiled before simulations to lower the computational burden. The equations of motions and stochastic elements of the model are covered in a later chapter. Heavily important to modeling a flight vehicle, this chapter begins with a discussion of reference frames and how they interact with the proceeding flight mechanics.

2.1 Reference Frames

Reference frames and their transformations are a critical component to any navigation algorithm. Reference frames describe the orientation of the flight vehicle with respect to a global or local reference point. Because different components of the FVDM are physically



Figure 2.1: Pilot flying Diamond DA-40 single propeller aircraft – the focus of the collection platform for this thesis [6].

located and oriented at unique locations of the Diamond DA-40, multiple reference frames are used to describe the forces and moments these components generate due to their position with respect to the Center of Gravity (CG) of the aircraft. Reference frame transformations are used to transform local forces and moments into a congruent reference frame such that they can be summed together correctly. This section describes the different reference frames used in this work for the development of the flight mechanics modules within the FVDM.

2.1.1 Flight Vehicle Reference Frame

The flight vehicle reference frame or body frame describes the reference frame with origin at the center of mass of the modeled Diamond DA-40. The body frame maintains alignment shown in Figure 2.2 and remains fixed to the flight vehicle at all times. The flight vehicle reference frame is essential to the FVDM as all forces and moments are added together in this frame before integration into global and local pose states. In the flight vehicle reference frame, x always points through the nose of the aircraft, y points to the right, or *starboard* side of the aircraft, and z points through the bottom of the aircraft.

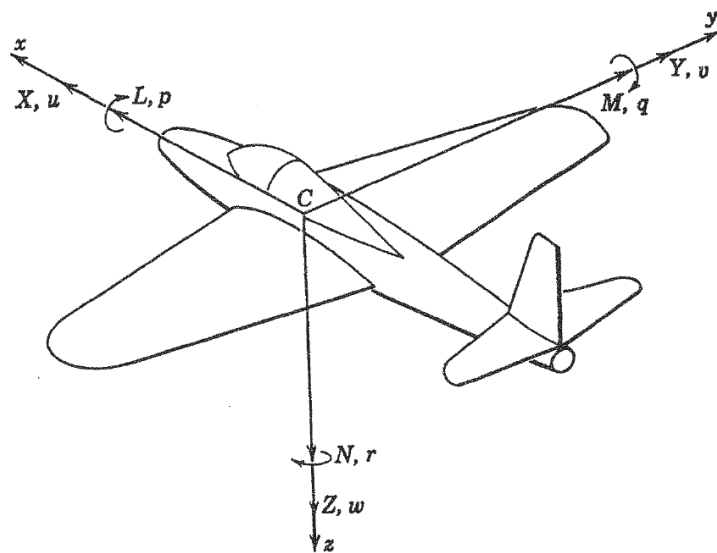


Figure 2.2: Standard flight vehicle reference frame used in this work. [24]

2.1.2 Propeller Reference Frame

The propeller reference frame describes the axis about which the propeller rotates with origin to the propeller nacelle, as seen in Figure 2.3. Propeller *pitch*, as it is referenced in this thesis is a rotation about the y axis, while the blades of the propeller rotate about the x axis. This axis remains fixed in orientation and position similarly to the flight vehicle reference frame. ϕ from Figure 2.3 describes the twist of the blade and is the rotation of the blades about the x axis. The twist of the blade helps the propeller produce more thrust, but is not used in any reference frame calculations.

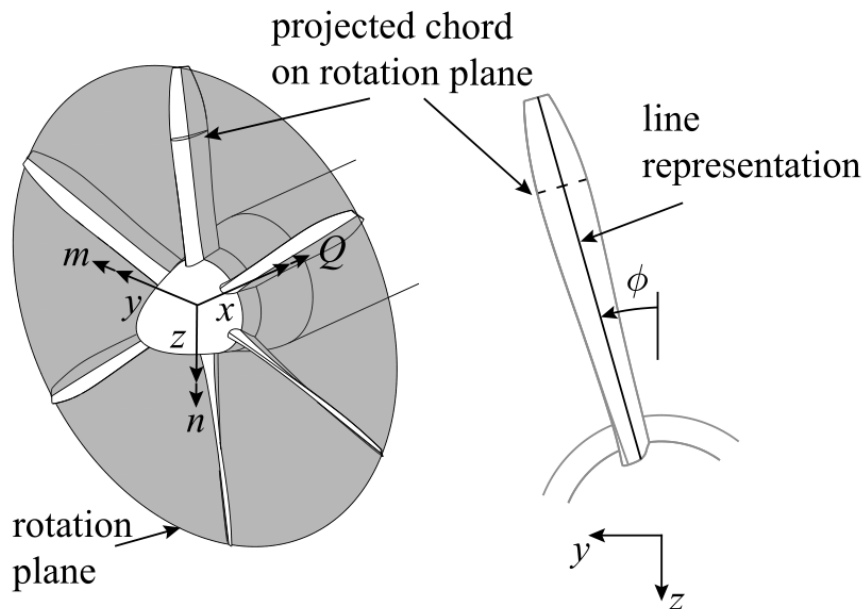


Figure 2.3: The reference frame used to model the dynamic of the propeller in this work [25].

2.1.3 Local Wind Reference Frame

The local wind reference frame is used to calculate the lift and drag generated from their respective components with origin at the center of mass of the modeled Diamond DA-40. The orientation of the reference frame changes such that β remains parallel with the trajectory of the aircraft and α remains parallel to the ground below. In literature, β and α are referred to as the *sideslip* and *angle of attack* of the aircraft during flight. Sideslip can be thought of as the left or right direction of the aircraft with respect to the incident free stream and angle of attack is the up or down direction of aircraft with respect to the incident freestream.

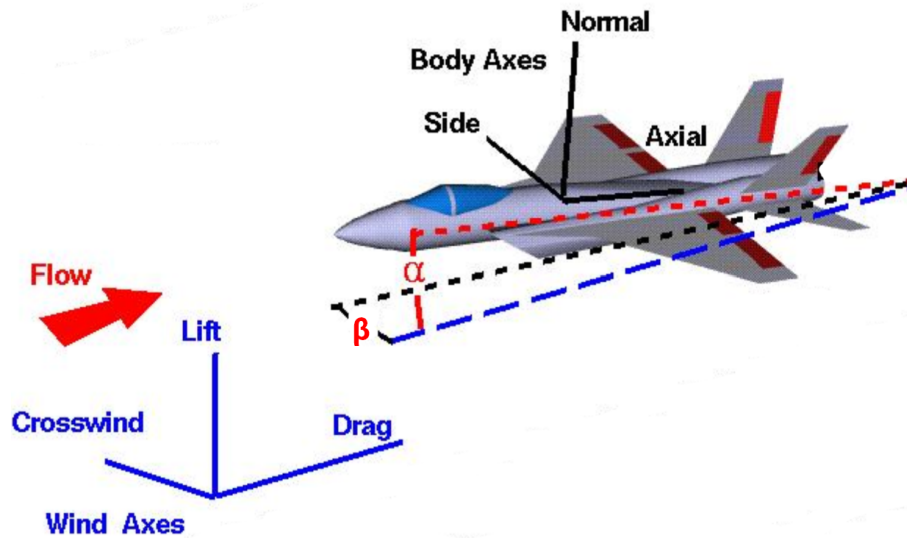


Figure 2.4: Flight vehicle reference frame and local wind reference frame (dotted lines) (Adapted from [26]).

2.1.4 Local Navigation Reference Frame

A local navigation frame is necessary to define the orientation of the flight vehicle. This work implements a North-East-Down (NED) reference frame in which the axes are aligned with the topographic directions as seen in Figure 2.5 by X_n , Y_n , and Z_n . The origin of the local navigation reference frame is defined at the initial position in the FVDM and navigation filter. The Euler attitude angles are used to describe the orientation of the Diamond DA-40 relative to the local navigation frame. When the aircraft has no roll, pitch, or yaw, this equates to the aircraft flying perfectly tangential to Earth's ellipsoid, with the nose of the aircraft pointing towards the North.

2.1.5 Global Reference Frames

Global Reference Frames are critical for GPS receivers to calculate position and velocity. In summary, a position from a receiver is calculated based on distances between the receiver and visible satellites (details about positioning using GPS satellites is provided in Chapter 3). As all GPS satellites operate in orbit around the Earth, the global reference frame provides a suitable solution that encompasses both satellite and receiver positions and velocities. In this thesis, two global reference frames are used. The Earth-Centered, Earth-Fixed (ECEF) and the geodetic

reference system (LLA). The ECEF reference frame is considered the standard global reference frame as it works well for defining both positions and velocities for satellites and receivers, but falls short when it comes to visualization as the origin of the ECEF frame is the center of the Earth. The geodetic reference frame is an alternative frame that allows a visualization advantage by defining positions based on the surface of the Earth. Both LLA and ECEF reference frames utilize the World Geodetic System 1984 (WGS84) to describe Earth's geoid and gravitational field as function of parameters in Table 2.1. Visualization of both reference frames can be seen in Figure 2.5.

Table 2.1: Properties describing the WGS84 ellipsoid

Property	WGS84 Value	Units
Equatorial Radius, R_0	6,378,137.0	meters
Polar Radius, R_P	6,356,752.31425	meters
Flattening, f	1/298.257223563	
Eccentricity, e	0.0818181808425	

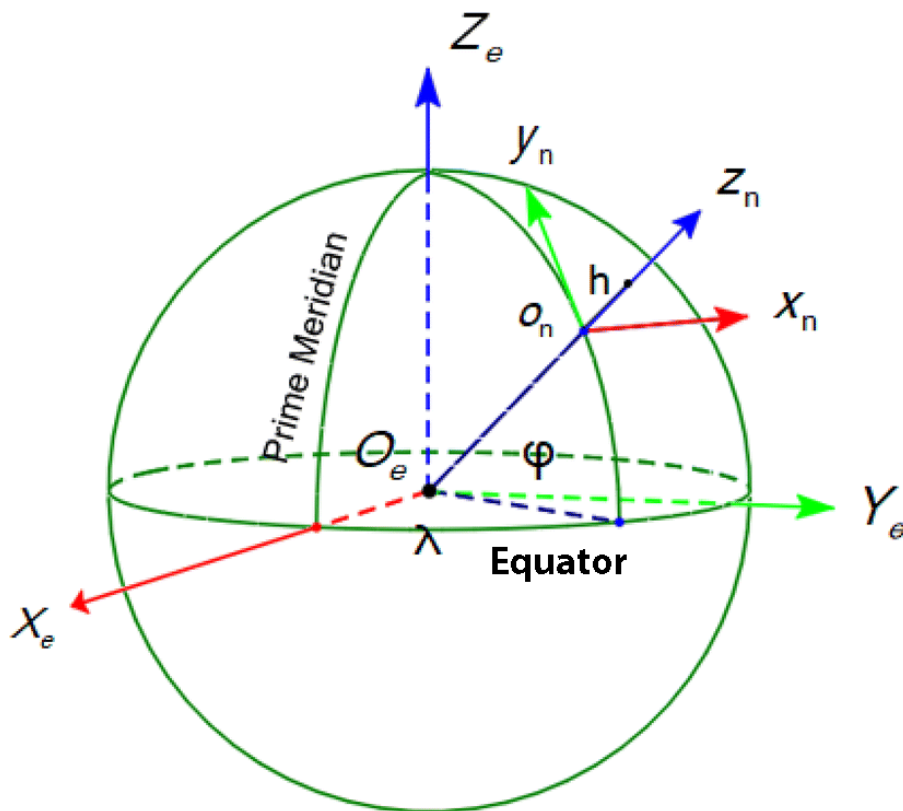


Figure 2.5: ECEF and geodetic reference frames as used in this thesis.

2.2 Reference Frame Transformations

Reference frame transformations are critical to any sensor fusion algorithm in order to properly add vectors of different frames together. Reference frame transformations range in complexity, the simplest being a transformation of the NED frame into the East-North-Up reference frame. A more complex transformation could be the global ECEF frame into the body frame. The reference frame transformations used in this thesis are described in the following subsections.

2.2.1 ECEF to LLA

One of the more complicated reference frame transformations is between the two global frames, ECEF and LLA. The conversion from LLA to ECEF is provided first in Equations 2.1, 2.2, 2.3, 2.4, and 2.5. The variables within these equations are defined in Table 2.1.

$$R_N(L) = \frac{R_0 (1 - e^2)}{(1 - e^2 \sin^2(L))^{3/2}} \quad (2.1)$$

Equation 2.1, $R_N(L)$ describes the one the two radii of curvature. In this case, the meridian radius of curvature describes the north to south radius. L is the estimated user longitude as described by the geodetic reference system. The second radii of curvature (Equation 2.2) describes the east to west radius and is known as the transverse radius of curvature ($R_E(L)$).

$$R_E(L) = \frac{R_0}{\sqrt{1 - e^2 \sin^2(L)}} \quad (2.2)$$

After the transverse radius of curvature is calculated, it is used with the LLA positions to calculate the ECEF X , Y , and Z positions as shown in Equations 2.3, 2.4, and 2.5.

$$X_{ECEF} = (R_E(L) + h) \cos(L) \cos(\lambda) \quad (2.3)$$

$$Y_{ECEF} = (R_E(L) + h) \cos(L) \sin(\lambda) \quad (2.4)$$

$$Z_{ECEF} = [(1 - e^2) R_E(L) + h] \sin(L) \quad (2.5)$$

The ECEF to LLA conversion is more complicated. In order to calculate the longitude, L , in Equation 2.6, the altitude, h , (Equation 2.8) must be calculated – and vice versa. Lastly, λ describes the latitude of the user position as described by the geodetic reference frame. The correct way to solve for positions in the LLA reference frame is to iterate until the difference between positions from iteration to iteration is miniscule.

$$L = \text{atan2} \left(\frac{Z_{ECEF} [R_E(L) + h]}{\sqrt{X_{ECEF}^2 + Y_{ECEF}^2} [(1 - e^2) R_E(L) + h]} \right) \quad (2.6)$$

$$\lambda = \text{atan} \left(\frac{Y_{ECEF}}{X_{ECEF}} \right) \quad (2.7)$$

$$h = \frac{\sqrt{X_{ECEF}^2 + Y_{ECEF}^2}}{\cos(L)} - R_E(L) \quad (2.8)$$

2.2.2 LLA to Local

The conversion from the LLA reference frame to the local navigation reference frame can be done by forming a Direction Cosine Matrix (DCM),

$$C_e^n = \begin{bmatrix} -\sin(L) \cos(\lambda) & -\sin(L) \sin(\lambda) & \cos(L) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(L) \cos(\lambda) & -\cos(L) \sin(\lambda) & -\sin(L) \end{bmatrix}, \quad (2.9)$$

and then multiplying the ECEF position vector to produce a position in the NED frame as discussed previously. In Equation 2.9, C_e^n describes the DCM used for the transformation. The notation follows such that the subscript (e) is the position in the original reference frame and the superscript (n) is the reference frame the position is being rotated into. This thesis will always follow this notation to avoid any confusion.

2.2.3 Local to Body

Similar to the conversion of LLA to the local navigation frame, the conversion from the local navigation to the flight vehicle reference frame can be done by forming the DCM (Equation 2.10).

$$C_n^b = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix} \quad (2.10)$$

In this case, the DCM comprises of the three Euler angles – roll (ϕ), pitch (θ), and yaw (ψ). To allow the matrix to fit the width of the paper, c and s denote the *cosine* and *sine* trigonometric functions, respectively.

If one wanted to convert position from the body to ECEF reference frame, multiplying Equations 2.10 and 2.9 provides the user with a DCM to do this (Equation 2.11).

$$C_e^b = C_n^b C_e^n \quad (2.11)$$

It should be noted that any of these DCM reference frame transformations can be inverted by simply transposing the final transformation matrix.

2.3 Atmosphere Model

In order to more accurately calculate a handful of dynamics modeled in this work, a model of Earth's atmosphere is needed to provide ambient temperature, pressure, and density. This thesis uses the ISA model to approximate ambient temperature, ambient pressure, and ambient density given a certain height above Mean Sea Level (MSL) [27]. Using an assumed linear distribution for temperature as a function of altitude, the ISA model assumes hydrostatic equilibrium as seen by Equation 2.12,

$$\frac{dP}{dh} = -\rho g, \quad (2.12)$$

where $\frac{dP}{dh}$ is the vertical pressure gradient as a function of air density, ρ , and acceleration due to gravity, g . After integrating Equation 2.12, the ISA model uses the ideal gas law (Equation 2.13)

$$P = \rho R_{\text{air}} T \quad (2.13)$$

to solve for the ambient pressure P , and density, ρ . A complete form of the ISA model is seen in Equations 2.14 and 2.15.

$$P = P_0 \exp\left(\frac{-g \Delta h}{R_{\text{air}} T}\right) \quad (2.14)$$

$$\rho = \rho_0 \exp\left(\frac{-g \Delta h}{R_{\text{air}} T}\right) \quad (2.15)$$

where P_0 and ρ_0 are atmospheric layer values for pressure and density, respectively; R_{air} is the specific gas constant for air and Δh is difference between the current altitude of the flight vehicle and altitude of the current atmospheric layer. Figure 2.6 describes these atmospheric parameters from MSL to 85,000 meters above sea level.

2.4 Aerodynamic Principles

In all phases of flight, the surfaces on any aircraft generate two aerodynamic forces – *lift* and *drag* along with an aerodynamic moment vector. Prior to the modern computer, these forces and moments were derived from swaths of wind tunnel data (Figure 2.7) under various conditions. Today, during the research and development phase of modern production aircraft, these aerodynamics components can be determined using Computational Fluid Dynamic (CFD) programs such as *FlightStream* [28]. This subsection provides the modeling equations for the surfaces of the airplane that generate lift, drag and moments during simulation.

A wing produces lift by allowing the air to move faster over the top of the airfoil. Bernoulli's principle equation (Equation 2.16)

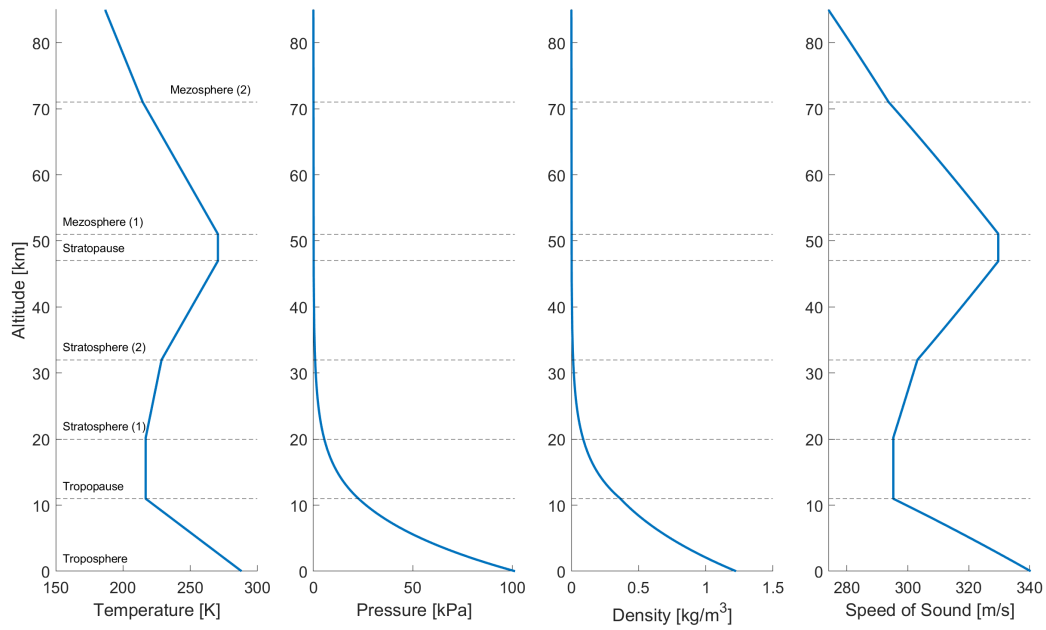


Figure 2.6: Absolute temperature, ambient pressure, air density, and speed of sound using the ISA model.



Figure 2.7: Subsonic wind tunnel in Auburn University's Aerodynamics Laboratory.

$$P_1 + \frac{1}{2} \rho V_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho V_2^2 + \rho g h_2 \quad (2.16)$$

shows that a faster moving fluid leads to a lower pressure gradient. On the bottom side of the wing, where the air is slower, a higher pressure gradient forms. This difference in pressure creates a force on the wing that lifts the wing and entire airplane into the air (Figure 2.8) [29]. In equation 2.16, V represents the aircraft forward velocity.

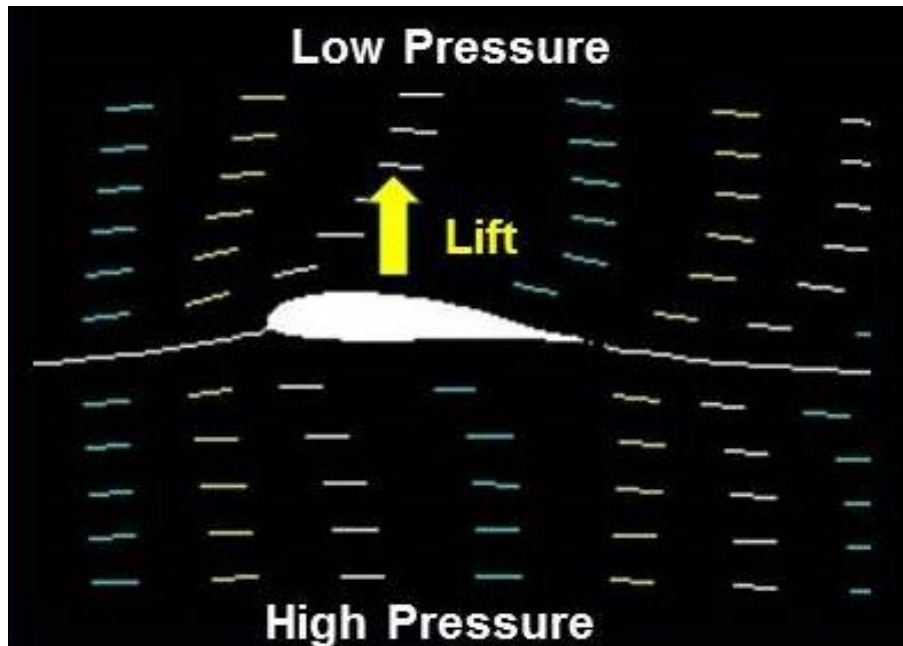


Figure 2.8: Pressure gradient and subsequent lift produced onto an airfoil. The continuous white line in the middle signifies the freestream incident wind [29].

The lift generated by the wing (and subsequently the aircraft) can be calculated using Equation 2.17 [30],

$$L = \frac{C_L \rho V^2 A}{2}, \quad (2.17)$$

where L is the lifting force, C_L is the lift coefficient for the aircraft, ρ is the air density, V is the velocity of the aircraft, and A is the surface area of aircraft. For the modeling of the Diamond DA-40 presented in this thesis, the aerodynamic coefficients are approximated using strip theory [31]. Strip theory is covered in great detail later in this chapter.

Lift is not the only force produced by the wing during flight. When any object moves through air an opposing force known as drag is generated. For the purpose of the modeling presented in this thesis, drag always opposes the motion of the aircraft. Similar to Equation 2.17, drag can be calculated using Equation 2.18.

$$D = \frac{C_D \rho V^2 A}{2} \quad (2.18)$$

Above, C_D is the drag coefficient and A is the surface area of the aircraft. The aerodynamic moments generated during flight of any aircraft are the single reason that control surface deflections cause the aircraft the roll, pitch or yaw in a certain direction. Just like the equations for lift and drag, the total aerodynamic moments are described by equations 2.19, 2.20, and 2.21.

$$M_l = \frac{C_l \rho V^2 A}{2} \quad (2.19)$$

$$M_m = \frac{C_m \rho V^2 A}{2} \quad (2.20)$$

$$M_n = \frac{C_n \rho V^2 A}{2} \quad (2.21)$$

In Equations 2.17, 2.18, 2.19, 2.20, and 2.21, the rolling (C_l), pitching (C_m) and yawing (C_n) coefficients need to be defined in order to calculate the aerodynamic forces and moments correctly. If known before hand, these coefficients can be calculated programmatically. Equation 2.22 describes the equation for finding the lift coefficient.

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_q} \frac{qc}{2V} + C_{L_{\delta_e}} \delta_e \quad (2.22)$$

Where C_{L_0} is the zero angle of attack lift coefficient, C_{L_α} is the lift coefficient due to angle of attack, α is the angle of attack, C_{L_q} is the lift coefficient due to the dynamic pressure, q , and $C_{L_{\delta_e}}$ is the lift coefficient due to the elevator deflection, δ_e . c is the chord length of the wing and V is the velocity of the aircraft. Equation 2.23 describes the equation for finding the drag coefficient.

$$C_D = C_{D_0} + K C_L^2 \quad (2.23)$$

Where C_{D_0} is the zero angle of attack drag coefficient and K is the lift induced drag coefficient of the aircraft.

The aerodynamic moment coefficient is split into three components, each defined by the direction they act about the flight vehicle (described by Section 2.1.1). The rolling, pitching, and yawing moment coefficients are described by their respective Equations 2.24, 2.25, and 2.26.

$$C_l = C_{l_\alpha} \alpha + C_{l_\beta} \beta + \frac{b}{2V} (C_{l_p} p + C_{l_r} r) + C_{l_{\delta_e}} \delta_e + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r, \quad (2.24)$$

$$C_m = C_{m_0} + C_{m_\alpha} \alpha + C_{m_\beta} \beta + \frac{c}{2V} (C_{m_\dot{\alpha}} \dot{\alpha} + C_{m_q} q) + C_{m_{\delta_e}} \delta_e \quad (2.25)$$

$$C_n = C_{n_\alpha} \alpha + C_{n_\beta} \beta + \frac{b}{2V} (C_{n_p} p + C_{n_r} r) + C_{n_{\delta_e}} \delta_e + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \quad (2.26)$$

In the above equations, $C_{(\cdot)(\cdot)}$ are the coefficients relating to their respective aircraft states. p , q , and r are the angular rates about the forward, right, and down direction in the flight vehicle reference frame. δ_e , δ_a , and δ_r are control surface deflections of the elevator, ailerons, and rudders, respectively. β is the sideslip angle of the aircraft and represents the angle between the motion of the aircraft and the direction the aircraft is pointing. Finally, b describes the half-wing span of the aircraft.

Unless these coefficients and their accompanying variables are somehow known *a priori*, determining what they should be can be cumbersome. However, one way to approximate the coefficients is to use a CFD program and collect tables of data for the aircraft under various flight condition and control surface deflections. The remaining part of this section describes the process of using *FlightStream* in combination with strip theory to approximate their true values.

For the aircraft presented in this thesis, strip theory is a means to better approximate the lift, drag, and moments coefficients described earlier. Figure 2.9 draws the horizontal and vertical strips the aircraft is discretized into. Their reference frame is static and is defined by the strips position and orientation relative to the center of mass of the aircraft. To rotate the aerodynamic forces generated from each strip, a simple yaw–pitch–roll transformation is used.

Strip theory is a process in which a 3D model of the aircraft is discretized into independent strips and then processed through a CFD program to approximate the performance of the aircraft under various flight conditions. The discretization of the modeled aircraft allows for strip classification where control surfaces are located on the aircraft. Identifying the control surfaces

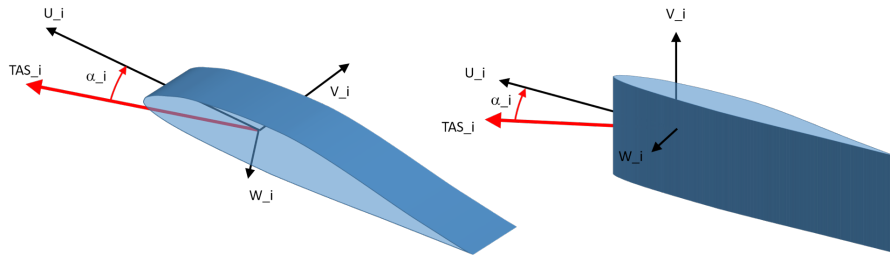


Figure 2.9: Horizontal and vertical strips that are analyzed for aerodynamic coefficients in FlightStream [32].

allows the CFD analysis to test different deflection angles, giving a more realistic data set of aerodynamic coefficients when a control surface is actuated (i.e. the deflection of the ailerons during a banked turn results in a larger aerodynamic moment compared to flying straight.). FlightStream [28] is used to generate aerodynamic coefficients for each strip. In FlightStream, the aircraft was evaluated on a range of angles of attack and sideslips, along with different control surface deflections. The aircraft was evaluated at a single velocity of 50 meters per second as the velocity component for the coefficients is just a scaling value.

Downwash is an aerodynamic property where the finite length of the lifting surfaces results in 3-dimensional flow fields causing reduction in effective angles of attack. To introduce downwash into the model and simulation, a reduced-ordered model utilizing integrated circulation was implemented [33, 34].

2.5 Aero-Propulsive Forces

Calculating lift, drag, and aerodynamic moments are useless unless the aircraft is moving. For any flight vehicle (except a glider), there is some form of engine or propeller that accelerates the incident freestream around the aircraft, allowing the lift, drag, and moment calculations to be of benefit. Similar to a ground vehicle, there is an infinite amount of complexity that can go into the modeling of the propulsive elements within a flight vehicle. The work in this thesis focuses on the engine, shaft, and propeller – followed by the effects these components have on the aircraft. The following subsections describe the aero-propulsive flight mechanics that allow aircraft to accelerate, take off, and maintain altitude during the length of the flight.

2.5.1 Piston Engine Model

Small general aviation aircraft generate propulsive forces through use of an engine that spins a shaft which is connected to a propeller. The engine modeled and simulated is a piston, air-cooled engine. The typical aircraft piston engine is modeled as a four-stroke Otto cycle [30]. The Otto cycle is a well-known thermodynamic theory that relies on a large air mass flow rate to generate power [35]. The components of the engine that produce the power are described below.

Because the engine is air-cooled, one of more important components is the engine manifold. The engine inlet manifold is a set of vents that allow the ambient air to feed directly into the combustion chambers where the oxygen in the air is combusted with the fuel to generate power. Based on the commanded throttle input, these manifold vents can be open or closed to let in more or less air, and in return the engine delivers a proportional amount of power to the shaft. The power production from the engine is heavily reliant on the density of air. The Gagg and Ferrar model,

$$\text{power} = \text{power}_0 \left(\frac{\rho}{\rho_0} - \frac{1 - \rho/\rho_0}{7.55} \right), \quad (2.27)$$

relates power production to the density of air, where ρ is the density of the ambient air as a function of altitude (Section 2.3), ρ_0 is the density of ambient air at sea-level on a standard day, and power_0 is the max engine power provided the density ratio and the nominal power production of the engine. From [36], the engine modeled in this work has a nominal power output on a sea-level standard day of 600 horsepower (447.42 kW). Figure 2.10 shows the power output of the engine modeled in this thesis based on the Gagg and Ferrar model and nominal sea-level power production.

However, not all power generated by the engine is absorbed by the shaft. There are losses that include shaft slippage or uneven distribution of fuel inside the combustion chamber. These *errors* are modeled as a power factor and compares the the ideal power produced by the engine to the power absorbed by the shaft. This proportional amount is queried at every time step in the simulation as seen in Figure 2.11.

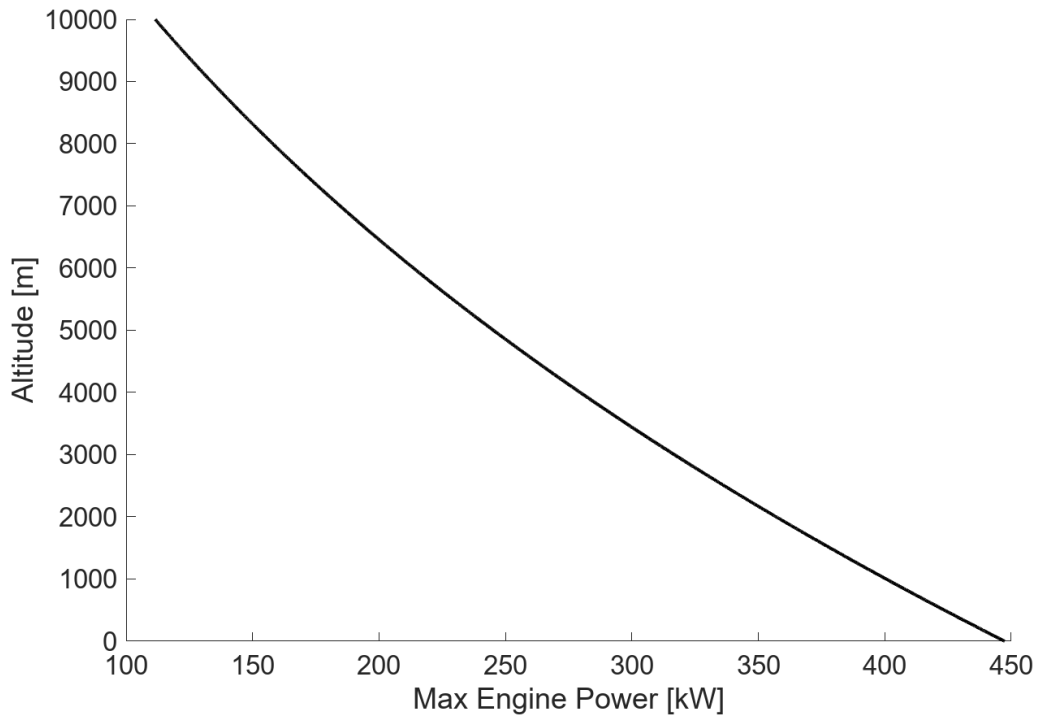


Figure 2.10: Power output for modeled engine as a function of aircraft altitude.

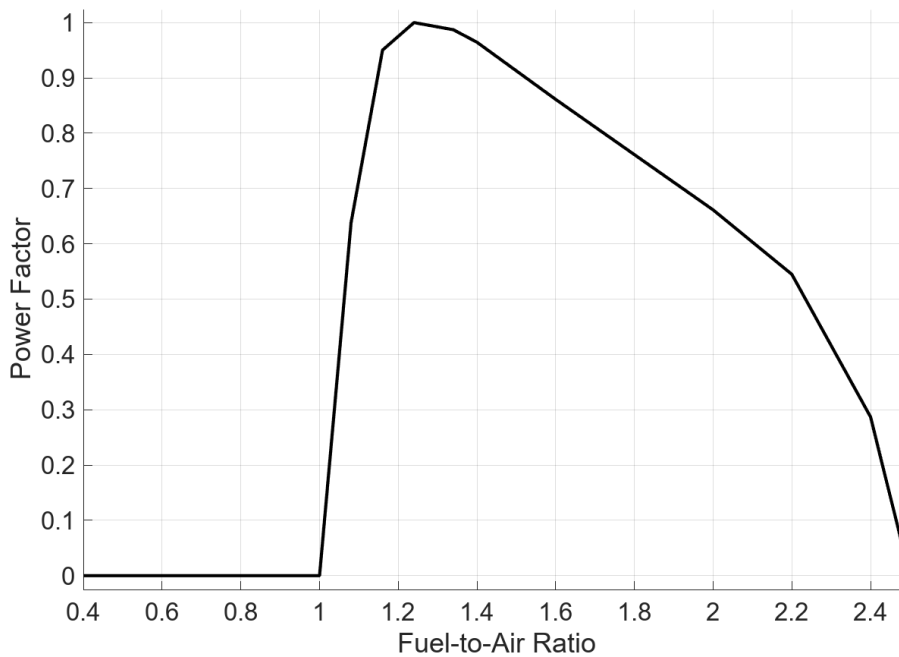


Figure 2.11: Power factor look up table for modeled engine.

The governor that exists in ground and flight vehicles exists such that drastic changes in throttle do not result in extreme ramps of torque that could structurally damage engine components. It limits the rate of commanded throttle to be linear such that rotational acceleration of the shaft and propeller is safely increased or decreased. The governor also controls fuel

consumption of the engine by controlling the engine speed. The fuel consumption is queried at every time step using a look up table that is pre-compiled before the simulation starts (Figure 2.12).

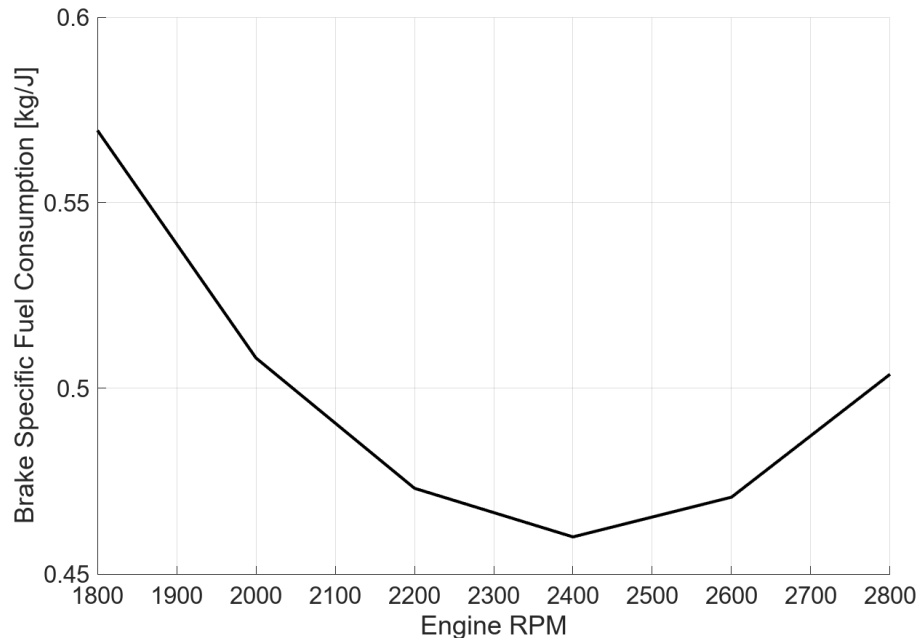


Figure 2.12: Brake specific fuel consumption look up table for modeled engine.

2.5.2 Propeller Modeling

The purpose of an aircraft propeller is to accelerate the volume of ambient air in front of the flight vehicle such that the lifting surfaces on the aircraft can generate lift and keep the aircraft at altitude. There are 3 main components to focus on when designing and manufacturing propellers. These are the materials used in building the propeller, the number of blades on the propeller, and the curvature and shape of each blade. While the focus of this thesis is not on details in propeller design, it is important to show how the history and differences between these three items affect the efficiency and performance a propeller has in generating thrust power for the aircraft.

The first modern propellers were designed in the early 1900's [37]. Originally made of wood, they featured 3 blades and crudely resembled airfoil shapes (Figure 2.13).

These propellers had a fixed pitch, meaning they could not rotate up and down during flight – severely costing the propeller thrust power and lowering overall fuel efficiency. In 1929,



Figure 2.13: Collection of historic wood and steel propellers [37].

Wallace Turnbull invented the variable pitch propeller (Figure 2.14), allowing pilots to control the pitch of the propeller, dramatically increasing propeller efficiency during flight [38].

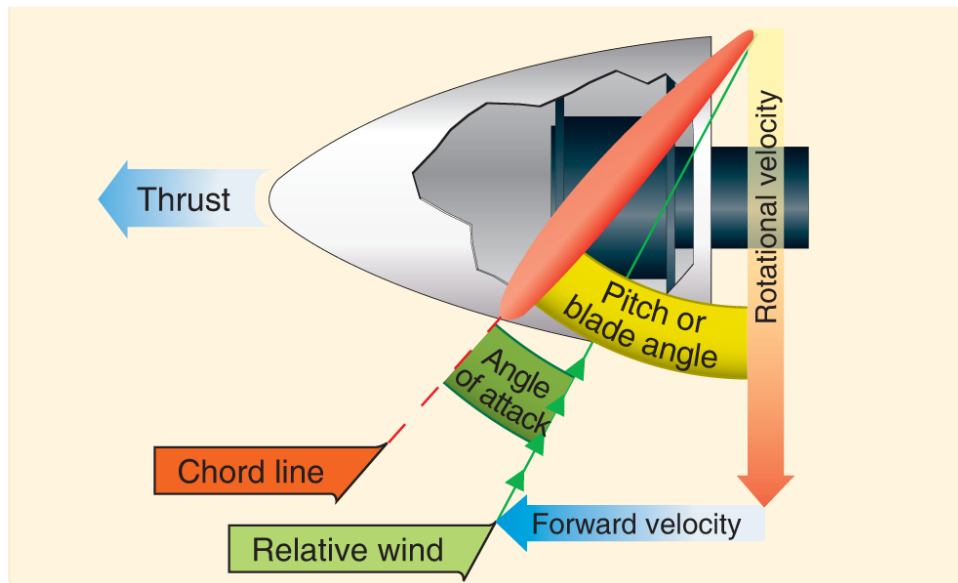


Figure 2.14: Illustration of the variable pitch propeller [39].

Through time and the advent of computers, CFD analyses shaped a handful of equations that define the performance and efficiency of propeller design. In this work, a three-blade Hartzell [40] composite propeller is used as it is the Original Equipment Manufacturer (OEM) propeller on the Diamond DA-40 aircraft. This propeller is a variable-pitch, constant speed propeller, meaning the pitch of the propeller is optimally adjusted for the current flight condition,

while at the same time, maintaining a constant rotational speed to conserve fuel and hold a consistent propeller efficiency.

Before the simulation is started, the propeller is analyzed through *FlightStream* [28], a surface vorticity flow solver, to analyze the propeller for its lift coefficient (C_L) at varying flight conditions.

Once the lift coefficient is determined, numerical look up tables are generated such that the calculation of the forces and moments generated by the propeller can be interpolated, allowing the simulation to be run in real time.

$$AF_{\text{per blade}} = \frac{1 \times 10^5 c_{\text{root}}}{16 D} (0.25 + 0.2 \lambda - 0.2) \quad (2.28)$$

To determine the amount of thrust and torque generated by the propeller, the activity factor (Equation 2.28) of the propeller must first be determined. The activity factor is a measure of the propeller's ability to absorb power and the effectiveness of each blade's width. $AF_{\text{per blade}}$ is the activity factor per blade, c_{root} is the length of a blade's chord at the root, D is the diameter of the propeller, and λ is the taper ratio. Table 2.2 describes the design characteristics of the propeller used in this thesis.

Table 2.2: Characteristics of the propeller modeled for this work.

Characteristics	Value	Units
c_{root}	0.1475	meters
Diameter (D)	1.9	meters
λ	0.8	–
C_L	0.5	–

Because the dimensions of the propeller are known, we can determine the activity factor of the blade to be (Equation 2.29, 2.30)

$$AF_{\text{per blade}} = \frac{1 \times 10^5 (0.1475)}{16 (1.9)} (0.25 + 0.2 (0.8) - 0.2) \quad (2.29)$$

$$AF_{\text{per blade}} = 101.89. \quad (2.30)$$

Now that the *Activity Factor* is known for the modeled propeller, the power (Equation 2.31) and thrust (Equation 2.32) coefficients can be calculated to show how well the propeller generates thrust when the aircraft is static (i.e. starting takeoff). For this analysis, we assume the engine to be producing a nominal 600 horsepower (447.42 [kW]) and applying a torque equivalent to 2200 [rpm] (36.6 [rev/s]).

$$C_P = \frac{P}{\rho n^3 D^5} \quad (2.31)$$

$$C_T = \frac{T}{\rho n^2 D^4} \quad (2.32)$$

In Equations 2.31 and 2.32, P is the engine power, denoted in kiloWatts, n is the rotational speed of the shaft, denoted in revolutions per second, T is the thrust of the propeller, denoted in kiloNewtons, and ρ is density of the ambient air. For these calculations, an assumed sea-level density is used (1.225 [kg/m^3]). Using Figure 2.15, nominal engine power and torque output, we can approximate the thrust coefficient (Equations 2.33, 2.34, 2.35 and 2.36).

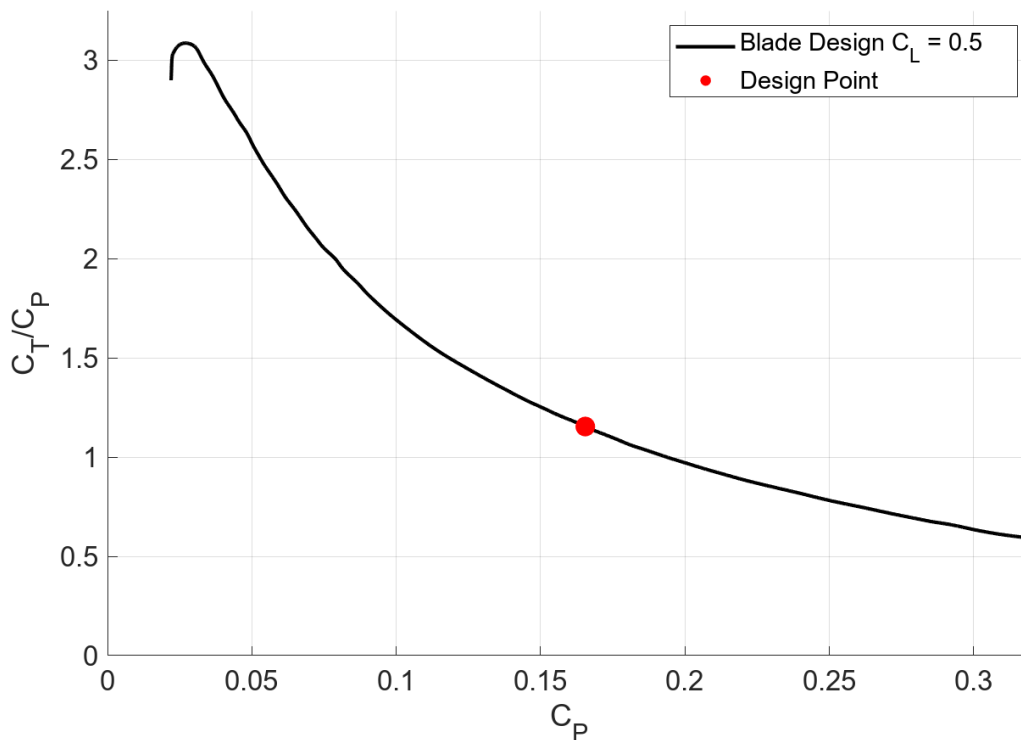


Figure 2.15: Static propeller thrust for the modelled propeller (Adapted from [41]).

$$C_P = \frac{(447.42)(550)}{(1.225)(36.6)^3(1.9)^5} \quad (2.33)$$

$$C_P = 0.16547 \quad (2.34)$$

$$\frac{C_T}{C_P} (@C_P == 0.16547) = 1.1544 \quad (2.35)$$

$$C_T = 0.18997 \quad (2.36)$$

Mapping C_T and C_P allows for computationally inexpensive queries with calculating propeller efficiency during flight (Equation 2.37).

$$\eta_P = \frac{C_T}{C_P} J \quad (2.37)$$

Propeller efficiency, η_P compares the amount of power produced by the engine and shaft to the amount of power applied to the ambient air. An ideal propeller efficiency would be 1, where all of the produced engine power is used to accelerate the ambient air. The *Advance Ratio*, J

$$J = \frac{V}{nD}, \quad (2.38)$$

describes how far the flight vehicle moves at each full revolution of the propeller.

The final step in calculating the forces and moments generated by the propeller is querying the thrust based on the previous equations (Equation 2.39).

$$T = \frac{P \eta_P}{V} \quad (2.39)$$

A handful of steps, provided below, list the necessary calculations needed to simulate the thrust for the Diamond DA-40 modeled in this thesis. Because of some of the quantities are constant (i.e. propeller diameter and lift coefficient), look up tables can be created before hand to lower the computational load.

1. Calculate $AF_{\text{per blade}}$ for given propeller design (Equation 2.28).

2. Query coefficients data table (Figure 2.15) to define C_P and C_T (Equations 2.31 and 2.32).
3. Query J (Equation 2.38) data table for given flight velocity and shaft rotational speed.
4. Calculate η_P (Equation 2.37), for given J , C_P , and C_T .
5. Calculate T using Equation 2.39.
6. The moment generated by the propeller is simply the cross-product between the thrust power and moment arm.

2.6 Landing Gear Model

Although not calculated often, the modeling of the aircraft's landing gear are important and should not be overlooked. However, because of the flight paths investigated in this thesis focus solely on the aircraft during flight, a simplified dynamic model is used to describe the forces and moments acting on the landing gear during landing. It should be noted that the aerodynamic calculations of the landing gear occur in the aerodynamically modeling section, while this section focuses on the moments and forces generated from the runway opposing the weight of the aircraft.

To describe the forces and moments generated during landing, a mass-spring damper system (Figure 2.16) can be used to simulate the struts, levers, and tire depression (Figure 2.17) that absorb much of the forces, moments and vibrations that act onto the aircraft during landing.

Expanding Newton's second law, the forces on each landing gear are solved in the vertical direction (Equation 2.40)

$$\sum F_z^i = m a_z = k_t^i \Delta z_t + k_{ldg}^i \Delta z_{a/c} + c_{ldg}^i \Delta \dot{z}_t, \quad (2.40)$$

where k_{ldg}^i and c_{ldg}^i are the spring and damper coefficients of the struts and levers respectively (see Table 2.3 for the values used in this simulated model). $\Delta z_{a/c}$ and $\Delta \dot{z}_t$, are the deflection and rate of deflection of the aircraft during landing. k_t^i and Δz_t are the tire *spring* coefficient and tire depression respectively. For a general aviation aircraft, the depression of the tire upon landing is relatively small such that this term is thrown out.

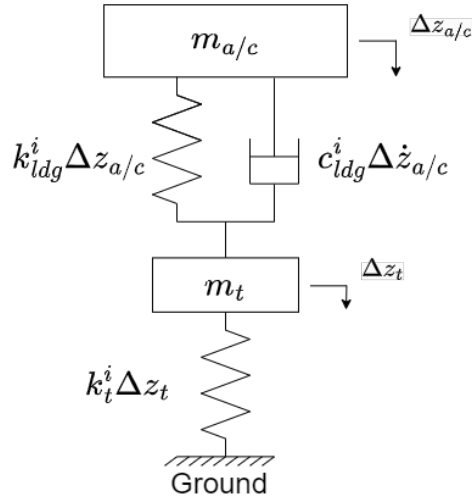


Figure 2.16: Mass-spring damper system, representing the components of landing gear on the aircraft (adapted from [42]).

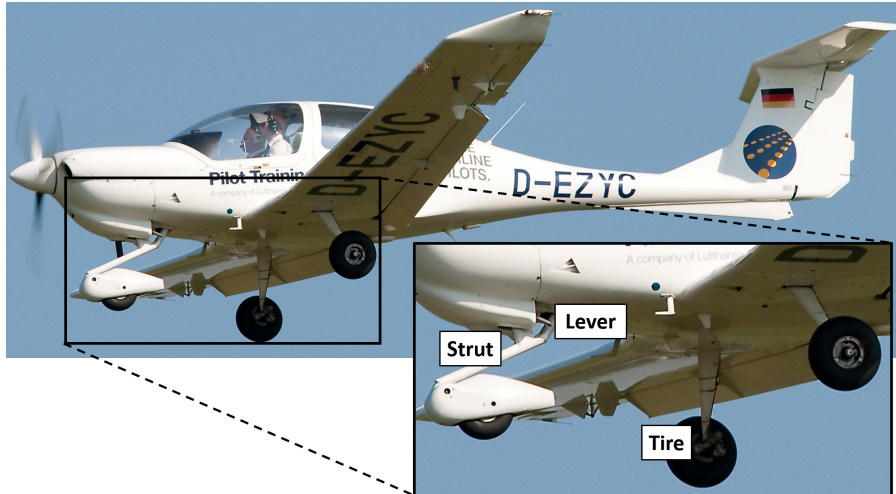


Figure 2.17: Identification of the landing gear components on the Diamond DA40.

Table 2.3: List of spring and damper coefficients for nose and rear landing gear.

i	Location	k $\left[\frac{kN}{m}\right]$	c $\left[\frac{kN s}{m}\right]$
1	Nose	50	11.3
2	Rear Right	80	14.3
3	Rear Left	80	14.3

The observed moments are solved by taking the cross product between the calculated forces of each landing gear and the moment arm (Equation 2.41).

$$\sum M^i = \text{cross}([0; 0; F_z^i], [x^i; y^i; z^i]) \quad (2.41)$$

In Equation 2.41, x^i , y^i , and z^i represent the moment arm that is derived from the center of gravity for the aircraft down to where each tire makes contact with the ground.

As a final note, it should be made clear that the forces and moments generated from the landing gear if and only if a tire has made contact with the ground.

2.7 Forces and Moments Calculations

The final product of the aforementioned systems sums to the total force and total moment acting on the body of the aircraft. This work demonstrates the high-fidelity modelling of engines, propellers, landing gear, and aerodynamic forces and moments the simulated flight vehicles generates while in flight. The final step of these calculations is to add them together in the body-fixed X , Y , and Z directions. This is demonstrated by Equation 2.42 and Equation 2.43.

$$\sum \mathbf{F} = \mathbf{F}_{prop} + \mathbf{F}_{aero} + \mathbf{F}_{LDG} + \mathbf{F}_{grav} \quad (2.42)$$

$$\sum \mathbf{M} = \mathbf{M}_{prop} + \mathbf{M}_{aero} + \mathbf{M}_{LDG} + \mathbf{M}_{grav} \quad (2.43)$$

It should be noted that \mathbf{F}_{LDG} and \mathbf{M}_{LDG} are only calculated when the aircraft is landing.

Once the forces and moments are calculated, they are used within the equation of motions that propagate the position, velocity, angular rates, and Euler attitude during the simulation of the flight.

2.8 Measures of Fidelity

This focus of this work is the development of a high-fidelity FVDM that closely resembles the behavior of a Diamond DA-40. That being said, the fidelity of the model could always be improved. There is a common question about how much fidelity is needed to provide accurate predictions of the behavior for the modeled flight vehicle. The research performed by [14] evaluates this question quite well, where the focus of the work was implementing a 47-state Extended Kalman Filter (EKF) in which 24 of the states were estimating aerodynamic

coefficients. For systems where the navigation filter is not fully cognizant of the flight vehicle it is installed on, this method can provide a more generalized solution when extensive knowledge of the flight vehicle does not exist. The work of [16] showcases the value of estimating the aerodynamic coefficients by presenting decent positioning estimates during periods of low satellite visibility. For the focus of this thesis, the compilation of look up tables and data sheets represents the DA-40 accurately to predict the behavior of the aircraft during flight given a set of controller inputs.

2.9 Conclusions

This section provided an overview of the reference frames used heavily when calculating the forces and moments the aircraft experiences during flight. Following a discussion of reference frames, the ISA model used in this work was defined with accompanying equations. The bulk of this chapter contains information about the use of strip theory for modeling the aerodynamics of the aircraft and a discussion of the engine and propeller modeling that produces the thrust for the Diamond-DA40 during flight. This chapter concludes with a description of the landing gear model and the summation of the forces and moments generated from the aforementioned sections that will be used in the equations of motions. Because the equations of motions are embedded within the proposed navigation filter for this work, they will be discussed in chapter 4.

Chapter 3

GPS Software Defined Receiver Architecture

Since 1993, GPS has provided users with capable hardware to determine their global position within seconds. The focus of this chapter revolves around the inner-workings of a scalar tracking SDR, capable of processing GPS L1 C/A (largely based on [43]). Split into four sections, the chapter begins with an explanation on how the receiver discretizes and digitizes the continuous, low-power signal collected by the antenna (Section 3.1). Following that, details about how the receiver knows what satellite signals are received are described. The third section draws on the different algorithms in scalar tracking that allow the received signal data to be replicated and decoded for satellite orbital parameters that play an important role in the navigation algorithms (Section 3.2, 3.3). Wrapping up this chapter, an overview on the weighted nonlinear least squares algorithm is provided along with detailed descriptions of the three measurements that each satellite is indirectly broadcasting to the receiver (Section 3.5).

3.1 Front End

The signals received by an antenna must be down converted and digitized before the processing of the signal can take place. The *Front End* of the receiver performs this conversion through a series of amplifiers, filters, and a Analog-to-Digital Converter (ADC) [44]. First, a signal is received by a Right-Hand Circularly Polarized (RHCP) antenna. The antenna can be passive, but for challenging scenarios, a powered, active antenna may be necessary. Because of the low received signal power that GPS constellations provide, the signal is amplified through a series of Low Noise Amplifiers (LNA) and Band Pass Filters (BPF). The LNA raise the power

of the received GPS signal and the BPF act as a first-step in removing non-GPS signals from processing. The last stage is passing the continuous received signal through the ADC where the signal is converted to digitized samples at a frequency of the receiver-embedded oscillator. The oscillator is filtered using a phase lock loop (PLL), described later.

The purpose in this mixing process is to transform the signal into a more manageable intermediate frequency while still maintaining the same modulation and Doppler applied to the signal [19]. Figure 3.1 describes the process on converting the analog, continuous signal into a discrete, digitized IF signal in block diagram form.

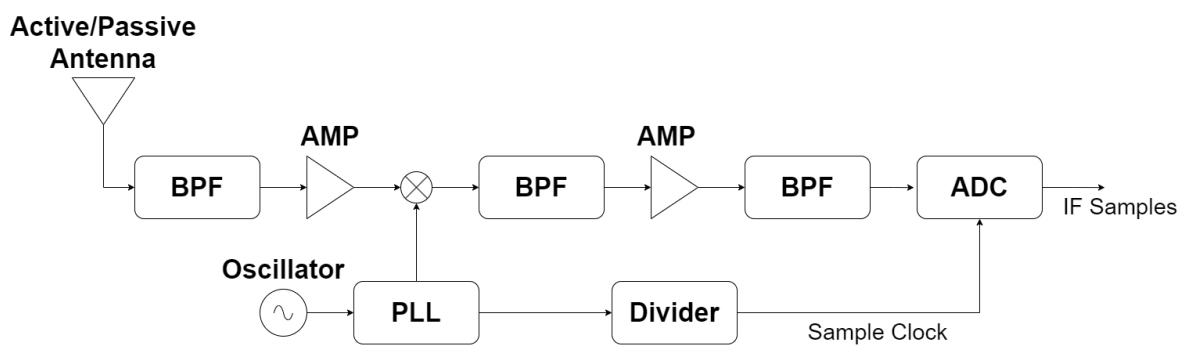


Figure 3.1: Block diagram of a software defined receiver front end.

3.2 Acquisition

Once the received signals are converted to a discrete form, the receiver will determine which satellites are transmitting and in-view. Acquisition correlates local replicas of a signal with the received data [45]. In order for there to be a large correlation magnitude, Pseudo-Random Number (PRN) codes must be within 1-chip and the frequency of the carrier wave must also be within 250 hertz of the true frequency. Correlation with the carrier frequency can be difficult due to the motion of the satellite, and even more difficult if the collection platform is also moving. The motion of the satellites with respect to the receiver bring a change to the carrier frequency known as the Doppler shift. For the algorithms in acquisition to successfully determine which satellites are in-view of the receiver, it is beneficial to correlate with each satellite for the specified constellation, at each code offset, and at a wide range of carrier frequency offsets [44].

A modern, ubiquitous approach to correlating the local replica signals with the received data is using a Parallel Code Phase Search (PCPS) algorithm in which the correlations occur in the frequency domain [46]. PCPS parallelizes the PRN code offset search space by converting the space from the time to frequency domain. This process is shown in Figure 3.2.

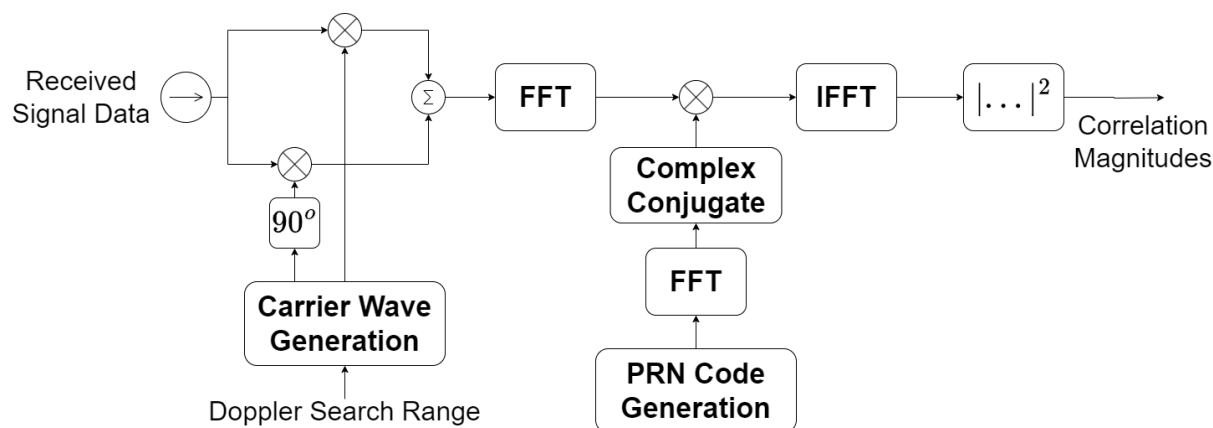


Figure 3.2: Block diagram of the PCPS acquisition algorithm applied to GPS L1 C/A signal modulation.

To start, the local replicas of the carrier wave are multiplied with the received data across both in-phase and quadrature channels in the complex domain. These resulting multiplications are summed together and passed through a Fast-Fourier Transform (FFT) to convert the vectors into the frequency domain. The second step involves converting the upsampled PRN codes of the satellites to the frequency domain and then applying the complex conjugate to frequency-based, PRN vectors. Multiplying the PRN replicas with the transformed carrier replicas provides the user with a cross-correlation in the frequency domain. To appreciate the correlation magnitudes, the output is passed through an inverse FFT, converting the correlation values from a frequency domain, complex matrix to a time domain, matrix. The last step requires determining the magnitude of each row and column in the matrix and then processing the next satellite until all satellites in the constellation are processed. If a correlation exceeds a certain magnitude, its location in the matrix of correlations indicates the estimated PRN code delay and Doppler shift carrier frequency for that particular satellite (Figure 3.3).

The resolution of the PCPS algorithm greatly depends on the grid of Doppler search bins; for a static receiver processing GPS L1 C/A transmissions, a typical resolution is $-15,000$ to $15,000$ Hertz in bins of 500 Hertz [46].

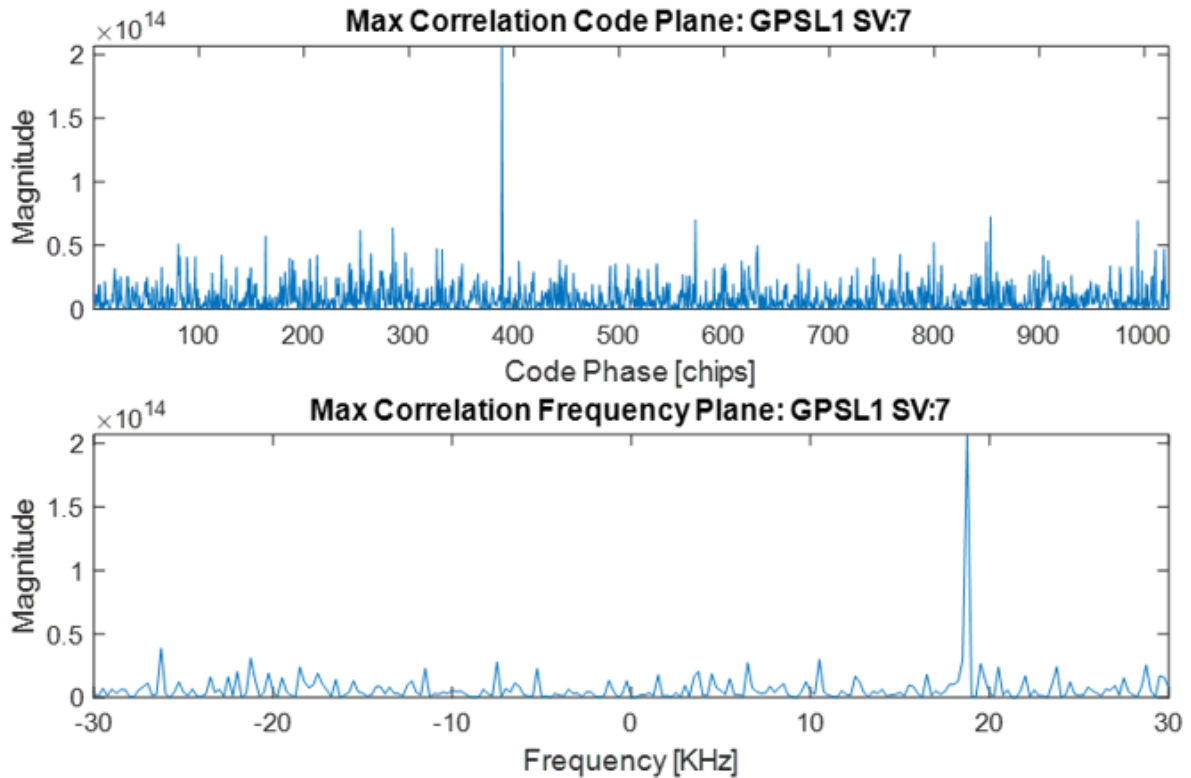


Figure 3.3: Successful acquisition of GPS SV:7 using the PCPS algorithm.

3.3 Tracking

Acquisition provides initial estimates of code delay and carrier frequency for each satellite signal that is in-view of the receiver. However, because of the motion of the satellites and the receiver, the delays of the code and changes in the carrier frequencies must continue to be estimated. Scalar tracking performs this process by opening a channel for each in-view satellite. For the duration of the recording, a number of algorithms within tracking continue to correlate the local receiver replica with the received signal data. For scalar tracking, Figure 3.4 describes the flow of how these algorithms produce accurate code replicas for the receiver to calculate satellite Position, Velocity, and Time (PVT) solutions.

3.3.1 Correlators

For a receiver to know it is replicating the signal correctly, it generates several correlators that are functions of the error in frequency, phase, and code phase between the replicated signal and the received signal data (Equation 3.1).

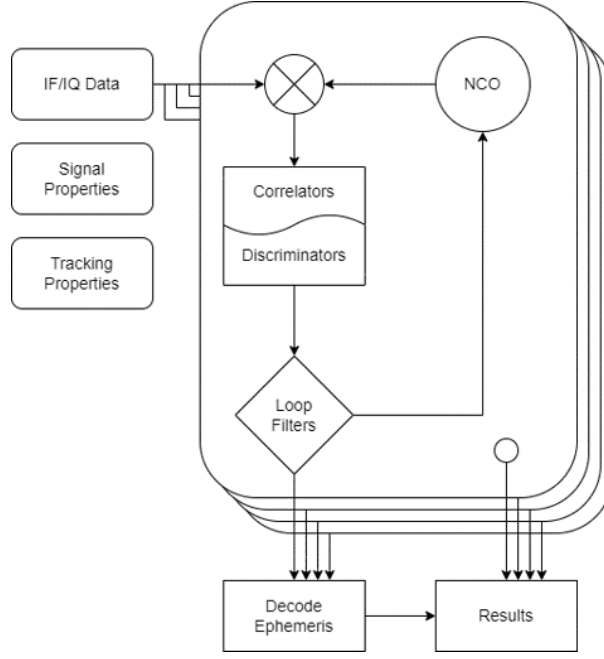


Figure 3.4: Flow diagram of scalar tracking algorithms.

$$\begin{aligned}
 IP(k) &= A R(\epsilon) D(k) \cos(\pi f_{err} T + \theta_{err}) + \eta_{IP} \\
 QP(k) &= A R(\epsilon) D(k) \sin(\pi f_{err} T + \theta_{err}) + \eta_{QP} \\
 IE(k) &= A R(\epsilon + d) D(k) \cos(\pi f_{err} T + \theta_{err}) + \eta_{IE} \\
 QE(k) &= A R(\epsilon + d) D(k) \sin(\pi f_{err} T + \theta_{err}) + \eta_{QE} \\
 IL(k) &= A R(\epsilon - d) D(k) \cos(\pi f_{err} T + \theta_{err}) + \eta_{IL} \\
 QL(k) &= A R(\epsilon - d) D(k) \sin(\pi f_{err} T + \theta_{err}) + \eta_{QL}
 \end{aligned} \tag{3.1}$$

Above, d is the offset specified in chips. This correlation process is also referred to as *integration and dump* in the literature as receivers will calculate the different correlators and dump the received signal data for that integration period, T . The integration period is usually set as the number of seconds for a full code cycle. In GPS L1 C/A, the standard coherent integration period, T , is 0.001 seconds [47]. The amplitude (Equation 3.2) is the measure of received signal power as a function of the integration period, frequency error (f_{err}), and carrier-to-noise ratio $\left(\frac{C}{N_0}\right)$.

$$A = \sqrt{2T \frac{C}{N_0} \frac{\sin(\pi f_{err} T)}{\pi f_{err} T}} \quad (3.2)$$

The rest of the terms in Equation 3.1 are as follows: θ_{err} is the phase error, described as the difference between phase of the replicated signal and the received signal data, $R(\epsilon)$ is the auto-correlation function defined by Equation 3.3, where ϵ describes the error in code phase. Finally, η describes additive white Gaussian process noise on each of the correlators.

$$R(\epsilon) = \begin{cases} 1 - |\epsilon| & \epsilon \leq 1 \\ 0 & \epsilon > 1 \end{cases} \quad (3.3)$$

3.3.2 Discriminators and Loop Filters

Once the correlators are calculated for a single integration period, discriminators use them to calculate errors in carrier and code phase. While a number of discriminators exist for each error term, the discriminators presented in this work are the most optimal for low and high carrier-to-noise ratios at the cost of being computationally expensive. Hardware receivers that must work in real-time may use simpler, faster discriminators.

The estimated error in carrier phase is generated using a Costas loop discriminator. It applies an arc-tangent function to the in-phase and quadrature prompt correlators to calculate the error in carrier phase from the most recent replicated signal (Equation 3.4)

$$\phi_{PLL} = \arctan\left(\frac{QP}{IP}\right) \frac{1}{2\pi} \approx \theta_{err} + \eta_{PLL} \quad (3.4)$$

The discriminator is divided by 2π to convert the error from radians to cycles. The Costas discriminator differs from other PLL discriminators as it is immune to inversion of the data bit integers. However, in more dynamic scenarios, the Frequency Lock Loop (FLL) discriminator may be more heavily relied upon than the Costas discriminator due to its high sensitivity [44].

The FLL discriminator derives its error by analyzing the change in carrier phase error across a single integration period

$$\phi_{FLL} = \arctan 2(\text{cross}, \text{dot}) \frac{1}{2\pi T} \approx f_{err} + \eta_{FLL}, \quad (3.5)$$

where

$$\begin{aligned} \text{cross} &= IP_1 QP_2 - IP_2 QP_1 \\ \text{dot} &= IP_1 IP_2 + QP_1 QP_2. \end{aligned} \quad (3.6)$$

Conventionally, $(IP, QP)_1$ are correlators in the middle of the integration period, while $(IP, QP)_2$ are correlators at the end of the integration period. The FLL discriminator is not as sensitive to dynamic stress and can handle changes in frequencies up to 500 Hertz.

The code phase error is determined using a DLL discriminator. It takes the early and late correlators that are shifted by a constant chip spacing, d , to determine if the current replicated signal is advanced or delayed relative to the received signal data. Equation 3.7 describes the DLL discriminator utilized in this work.

$$\phi_{DLL} = \frac{\sqrt{IE^2 + QE^2} - \sqrt{IL^2 + QL^2}}{2\sqrt{IE^2 + QE^2} + \sqrt{IL^2 + QL^2}} \approx \epsilon + \eta_{DLL} \quad (3.7)$$

The DLL discriminator produces code phase errors within 0.5 chips, but becomes unstable if the correlator errors are greater than 1.5 chips from the true code phase. This is allowable as errors of this magnitude are beyond the linear pull-in range of the DLL loop filter [44].

For scalar tracking, loop filters apply the aforementioned errors to more accurately replicate the signal for the next integration period. For this work, a second-order PLL with a first-order FLL assist is used to track changes in carrier frequency and phase (Figure 3.5).

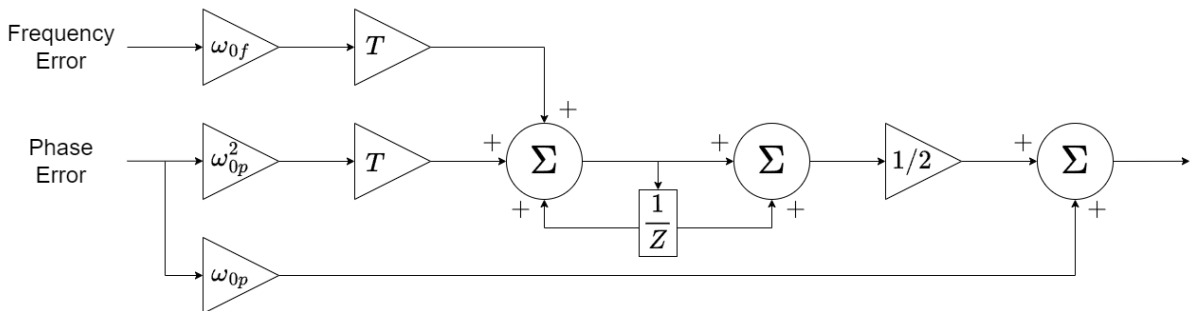


Figure 3.5: Discrete block diagram of the PLL-FLL loop filters used in this work [44].

The natural-radian frequency, ω_0 , is set by the user, and common values for processing GPS L1 C/A exists in [44]. The subscripts f and p represent ω_0 for the FLL and PLL, respectively.

The DLL loop filter works in a similar fashion, but without the FLL loop filter assist. The output of the PLL-FLL loop filter (Figure 3.5) is used to aid in replicating the true code phase (Figure 3.6).

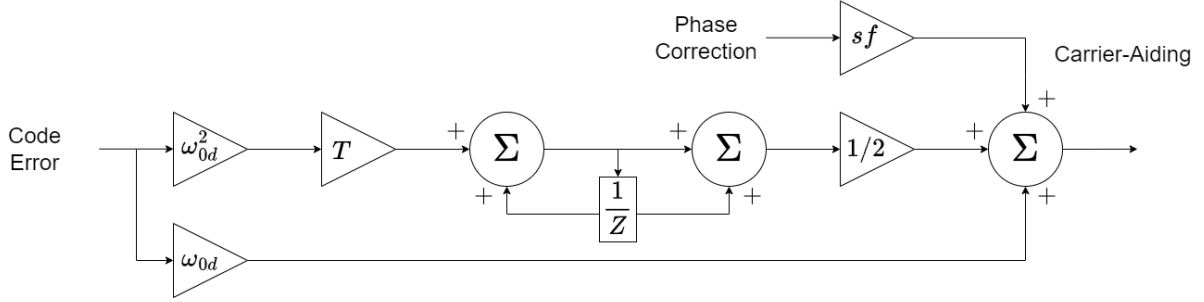


Figure 3.6: Discrete block diagram of the DLL loop filter used in this work [44].

In Figure 3.6, ω_{0d} is the natural-radian frequency for the DLL and sf is known as the scale factor,

$$sf = \frac{R_c}{f_L}, \quad (3.8)$$

where R_c is the spreading (i.e. PRN) code rate, and f_L is the signal carrier frequency. For GPS L1 C/A, these are 1.023×10^6 Hertz and 1575.420×10^6 Hertz, respectively.

If the correlators, discriminators, and loop filters are correctly working in tandem, the replicated signal should accurately represent the received signal data and the receiver can decode the bits that translate to the broadcast navigation message from each satellite (Figure 3.7). Five subframes in the navigation message provide satellite ephemeris, or orbital parameters, that the receiver can use to propagate satellite PVT and determine its position through time [48].

3.4 Navigation Algorithms

Once the signal is accurately replicated and correctly decoded, the receiver begins to estimate a PVT solution. A position solution is found by multi-late ration, meaning if there exists at least three unique ranges from known locations, then a position solution exists [49]. Parallel to finding the position solution, receiver velocity can be found in the same way, using

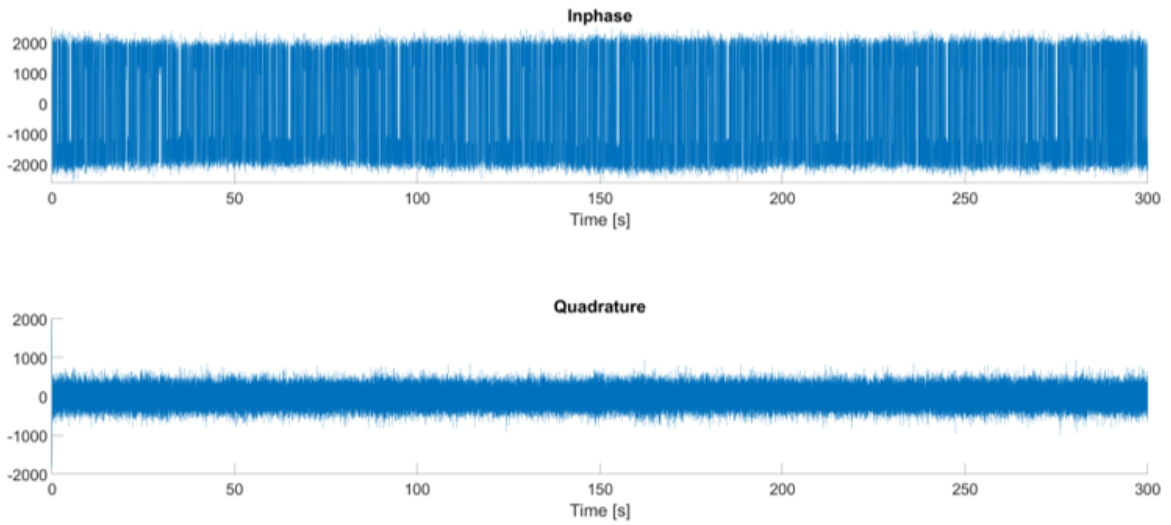


Figure 3.7: Tracked navigation message for GPS SV:7 using the ascribed methods.

range rates in place of range. This sections describes calculating the ranges and range rates, and how they are used in the navigation processor for an estimated PVT solution.

3.4.1 GPS Measurements

There exists three main measurements from GPS satellites. The first of the three is pseudorange. By definition, it is the total time of transit from transmission to reception of the signal by the receiver. This value is then multiplied by the speed of light, providing this transit time in units of meters (Equation 3.9).

$$\tilde{\rho}^j = (t_r - t_t^j) c \quad (3.9)$$

In order to calculate a pseudorange, ρ , from the j^{th} satellite the transmitted time of the signal, t_t , is subtracted from the received time, t_r , and then multiplied by the speed of light in vacuum, c , as described before. For the rest of the work presented in this thesis, variables with (\sim) represent a measurement. The pseudorange from Equation 3.9 is a measurement because both the receiver clock and satellite clock have inherit biases and drifts that perturb their values. In addition to the errors in the clocks, the signal in space travels through the atmosphere,

delaying the signal until it reaches the receiver. We can model Equation 3.9 as a function of errors and delays. Describing the pseudorange in meters gives us,

$$\tilde{\rho}^j = r_u^j + ct^j + ct_u + I_u^i + T_u^i + M_u^i + \eta_{\rho}^i, \quad (3.10)$$

where,

$$r_u^j = \sqrt{(x^j - x_u)^2 + (y^j - y_u)^2 + (z^j - z_u)^2}, \quad (3.11)$$

t_u is the clock bias of the user, t^j is the clock bias of the j^{th} satellite, I is the ionospheric delay, T is the tropospheric delay, M is the delay of multipath, and η is additive Gaussian white noise all time i . In Equation 3.11, variables noted $(\)_u$ are the respective user positions and $(\)^j$ are the respective satellite positions.

Positioning with GPS is possible because of the atomic clocks on board each satellite. These clocks are highly stable and allow the Time Of Week (TOW) to be transmitted in the data message. Although the on board clocks are incredibly stable, the PRN sequence cannot be precisely transmitted at every millisecond. Fortunately, these clock errors are modeled in the data message, so t^j is known in the pseudorange equation. Clocks on board receivers are not quite as stable, and the received time of the signal is not known upon a cold start. A solution to this problem is initialize the received time to the first satellite transmit time and add a nominal offset of 66.7 milliseconds. This offset stems from a back of the envelope calculation based on the distance between MEO satellite orbits and the center of the Earth (Equation 3.12).

$$(t_u - t^j)^{i=0} = \frac{d_{MEO}}{c} \approx \frac{20000000}{299792458} \approx 0.0667 \quad (3.12)$$

The addition of the unknown receiver clock bias adds a fourth dimension to the trilateration position solution. This effectively requires that the receiver needs the ranges from four satellites to estimate x_u, y_u, z_u , and t_u .

The second measurement from GPS satellites is pseudorange-rate. By definition, pseudorange-rate is the measurement of the line of sight velocity that can be directly derived from changes in carrier frequency, also known as the Doppler frequency (Equation 3.13).

$$\tilde{\rho} = -\frac{(f_c - f_{IF}) c}{f_t} \quad (3.13)$$

Above, f_c is the estimated frequency of the carrier wave, f_{IF} is the receiver intermediate frequency, and f_t is the nominal transmit frequency. Another way to represent pseudorange-rate measurement is by deriving Equation 3.10 with respect to time. This geometric representation is seen in Equation 3.14.

$$\tilde{\rho} = \dot{r}_u^j + c \dot{t}_u + \dot{I}_u^i + \dot{T}_u^i + \eta_{\dot{\rho}}^i \quad (3.14)$$

From Equation 3.14, it is assumed that the atomic clocks on board the satellites are stable enough that the satellite clock drift term is 0. The same can be said for \dot{M}_u^i where the error-rate due to multipath is miniscule.

Similar to the pseudorange measurement, if the receiver clock was perfect, only three satellites would be needed to measure pseudorange-rate. However, because of the instability in the receiver clock, the drift adds a bias to the frequencies. Therefore, four unique pseudorange-rates are still required in order to calculate receiver velocity.

The last measurement from GPS satellites is the estimate of noise on a signal. The receiver utilizes a Carrier-to-Noise density ratio (C/N_0) to determine the quality of pseudorange and pseudorange-rate measurements from each satellite. Equation 3.15 demonstrates how C/N_0 is calculated in this work.

$$\frac{C}{N_0} = 10 \log_{10} \left(\frac{\tilde{A} - 4\hat{\sigma}_\eta^2}{2T\hat{\sigma}_\eta^2} \right) \quad (3.15)$$

From Equation 3.15,

$$\tilde{A} = (IE + IL)^2 + (QE + QL)^2 \quad (3.16)$$

is the measured power of the accurately tracked signal using correlators defined in a previous section. σ_η^2 is the variance in correlators that are the result of a shifting the replicated signal far outside of any correlation with the data of the received signal data. This work shifts

these correlators by 100, 200, and 300 chips each [8]. Furthermore, this variance is filtered with a moving average using a ratio of 0.99 : 0.01 between the current calculated variance and the previous filtered variance. Shifting the replicated signal by a large number of chips and then correlating the shifted signal with the received signal data is computationally expensive, but is necessary if using a Bayesian estimator or weighted least squares approach in both open-loop and vector tracking navigation algorithms.

3.4.2 Open-Loop Architectures

Open-loop architectures are navigation algorithms that provide no feedback to the tracking scheme described in the previous section. In benign, static and constant velocity scenarios, these algorithms still provide accurate PVT solutions and are critical for a closed-loop architecture like vector tracking to work properly. The following section covers Weighted Non-Linear Least Squares (WNLS), the method used to initialize the vector tracking algorithms in this work.

The state vector for receiver initialization comprises the position and velocities of the receiver in the ECEF frame, along with receiver clock bias and clock drift terms (Equation 3.17)

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_u & \hat{\dot{x}}_u & \hat{y}_u & \hat{\dot{y}}_u & \hat{z}_u & \hat{\dot{z}}_u & c\hat{t}_u & c\hat{\dot{t}}_u \end{bmatrix}^T \quad (3.17)$$

For the rest of the work presented in this thesis, variables with a (^) represent an estimate of that variable. In Equation 3.17, T simply means the transpose of the array.

However, WNLS tries to estimate the errors between the true states and the estimated states (Equation 3.18).

$$\delta\hat{\mathbf{x}} = \begin{bmatrix} \delta\hat{x}_u & \delta\hat{\dot{x}}_u & \delta\hat{y}_u & \delta\hat{\dot{y}}_u & \delta\hat{z}_u & \delta\hat{\dot{z}}_u & c\delta\hat{t}_u & c\delta\hat{\dot{t}}_u \end{bmatrix}^T \quad (3.18)$$

These error states are then mapped to measurement residuals using the observation matrix, \mathbf{H} (Equation 3.22). \mathbf{H} is defined by the Jacobian of \mathbf{Y} with respect to $\delta\hat{\mathbf{x}}$, or in a mathematical form,

$$\mathbf{H} = \frac{\partial \mathbf{Y}}{\partial \delta\hat{\mathbf{x}}}. \quad (3.19)$$

If we define \mathbf{Y} as

$$\mathbf{Y} = \begin{bmatrix} \tilde{\rho}^1 - \hat{\rho}^1, & \tilde{\rho}^1 - \hat{\rho}^1, & \dots & \tilde{\rho}^1 - \hat{\rho}^1, & \tilde{\rho}^1 - \hat{\rho}^1 \end{bmatrix} \quad (3.20)$$

and then find the Jacobian of \mathbf{H} , a relationship forms between the pseudoranges and pseudorange-rates and the ECEF position and velocity estimates of the receiver in the form of a unit vector (Equation 3.21).

$$\mathbf{u}_{x,y,z}^j = \frac{\begin{bmatrix} x^j - \hat{x}_u, & y^j - \hat{y}_u, & z^j - \hat{z}_u \end{bmatrix}}{\hat{r}_u^j} \quad (3.21)$$

Because the bias and drift of the clocks directly map to their estimated states, they are equal to 1.

$$\mathbf{H} = \begin{bmatrix} -u_x^1 & 0 & -u_y^1 & 0 & -u_z^1 & 0 & 1 & 0 \\ 0 & -u_x^1 & 0 & -u_y^1 & 0 & -u_z^1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -u_x^j & 0 & -u_y^j & 0 & -u_z^j & 0 & 1 & 0 \\ 0 & -u_x^j & 0 & -u_y^j & 0 & -u_z^j & 0 & 1 \end{bmatrix} \quad (3.22)$$

The last piece in WNLS algorithm is defining the weights. \mathbf{W} is utilized by the algorithm to place more confidence in signals with a high C/N_0 compared to satellite signals with lower C/N_0 . The weighting matrix is shown in Equation 3.23.

$$\mathbf{W} = \begin{bmatrix} \sigma_{\rho^1}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\rho^1}^2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\rho^j}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\rho^j}^2 \end{bmatrix}^{-1} \quad (3.23)$$

Where

$$\sigma_{\rho^j}^2 = \frac{\lambda_{code}^2}{2T^2 \left(\frac{C}{N_0} \right)^2} + \frac{\lambda_{code}^2}{4T \frac{C}{N_0}} \quad (3.24)$$

is the variance of the pseudorange measurement from the j^{th} satellite and

$$\sigma_{\rho^j}^2 = \left(\frac{\lambda_{carrier}}{\pi T} \right)^2 \left(\frac{2}{T^2 \left(\frac{C}{N_0} \right)^2} + \frac{2}{T \frac{C}{N_0}} \right) \quad (3.25)$$

is the variance of the pseudorange-rate measurement from the j^{th} satellite. In Equations 3.24 and 3.25, λ_{code} is the wavelength of data signal and $\lambda_{carrier}$ is the wavelength of the carrier signal. In this case, C/N_0 is specified in dB-Hz.

Once the above matrices are created, the WNLS algorithm (Equation 3.26) can be performed iteratively until the magnitude of the error state vector is small, meaning the algorithm has converged.

$$\delta \hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{W} \mathbf{Y} \quad (3.26)$$

Once an initial receiver position has been found, vector tracking channels can open and a closed form PVT solution can start to be processed (Figure 3.8). Vector tracking loops and algorithms are discussed in the next chapter.

3.5 Conclusions

The architecture of a scalar-tracking software defined receiver was explained. For more information on any of the subsections discussed in this chapter, detailed descriptions of each can be found in the literature. Because the focus of this thesis is vector tracking, the meaning of this chapter was to provide the reader with a fundamental understanding of tracking loops that will be expanded on further in the next chapter.

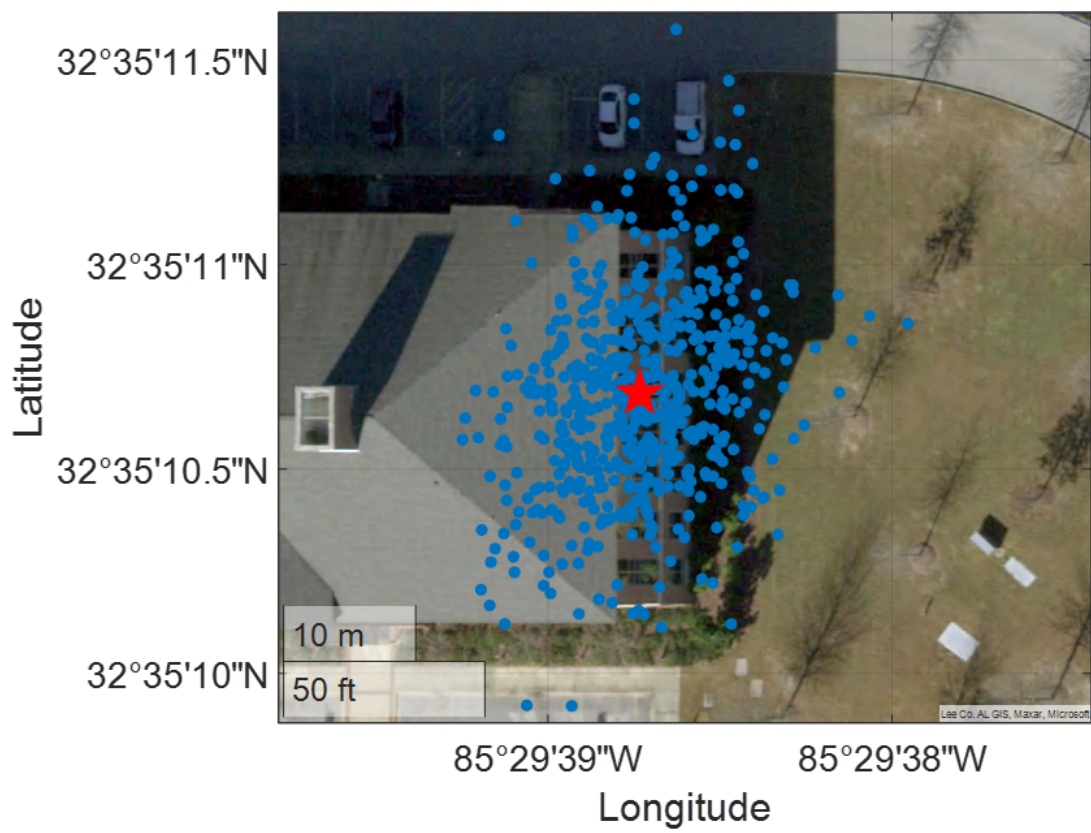


Figure 3.8: Static position of receiver using a WNLS navigation algorithm.

Chapter 4

Proposed Navigation Filter Architecture

Tracking loops discussed in Chapter 3 are critically important to receivers as they adapt the replica signal to match the received signal data for proper decoding. However, traditional scalar loop filters assume a static noise bandwidth, regardless of receiver or satellite dynamics. If either platforms have unmodeled dynamics, these static bandwidths can permit too much noise into the navigation solution, or neglect some of the dynamics by filtering too much of the signal. One solution to this problem is implementing an adaptive Kalman filter to optimally select bandwidths [50]. In the adaptive Kalman filter, the Kalman filter estimates the proper bandwidth based on discriminator residuals and modeled variances, but is agnostic to the dynamics of the receiver or the satellite dynamics. The addition of an adaptive Kalman filter is an improvement, but leaves a lot to be desired as each channel is still being tracked individually, resulting in low-powered channels having a high likelihood of being lost.

Another, more optimal, solution is to estimate the local replica signal from receiver and satellite dynamics at every integration period. This requires an updated estimate of the navigation solution at every integration period. This approach combines the adaptive bandwidth from [50] along with knowledge of the receiver and satellite dynamics stemming from the navigation solution. This closed-loop approach is known as vector tracking and will be discussed in greater detail later on in this chapter. Specifically, the Vector Delay and Frequency Lock Loop (VDFLL) is the vector tracking implementation used in this thesis.

To build on vector tracking, this work proposes an addendum to the existing navigation filter architecture by adding a FVDM to predict the trajectory of flight vehicle. When coupling external propagation schemes to GPS measurements, there are three common regimes.

The simplest form is loosely-coupled, where receiver position and velocity measurements are coupled with the predicted states from the external sensor to provide an improved receiver state estimate (Figure 4.1). The advantages of loose coupling is that the GPS receiver does not need to be modeled extensively; however, the noise provided by the receiver is not white, making the Kalman filter sub-optimal [21]. Furthermore, a loosely-coupled architecture only works with at least 4 satellites are transmitting to the receiver – which is not always the case for GPS-challenged environments [51].

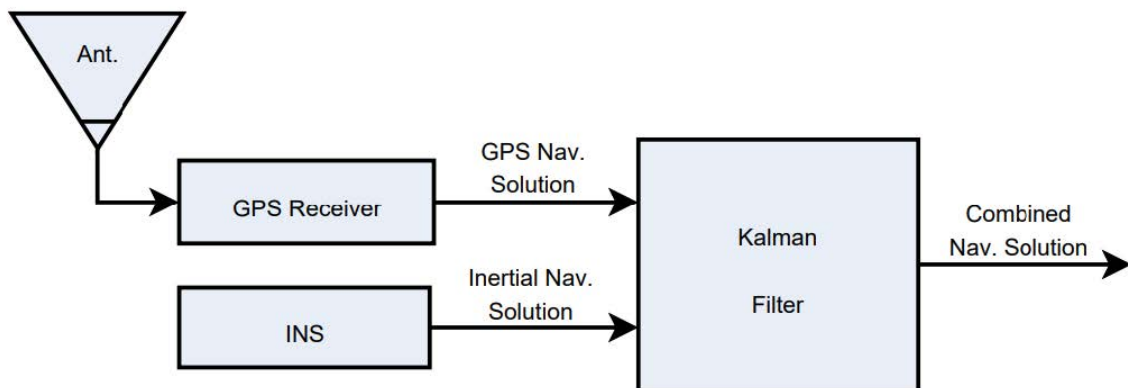


Figure 4.1: Loosely-coupled architecture between GPS and INS systems [52].

A more complicated coupling is tightly-coupled, where the navigation filter receives raw pseudorange and pseudorange rate measurements from the GPS receiver [44] (Figure 4.2). Because of this, tightly-coupled systems are still able to perform with less than 4 satellites, for a limited time. The draw back to tightly-coupled navigation architectures is the complexity of implementation. However, a working tightly-coupled system provides generous improvements over loosely-coupled frameworks, especially in GPS-challenged environments [9].

Deeply-coupled, or ultra-tight coupling, is the last of the common methods to couple external state predictions to GPS measurements. The downsides to deeply integrating an external sensor or vehicle dynamic model to GPS measurements is increased complexity and the reliance on the receiver to know its initial position – as is common with vector tracking. Generally, a deeply integrated system revolves around adapting the EKF that exists for a vector tracking architecture and appending the external dynamic model. This way, the GPS receiver and the added dynamic model are tied together at the most basic level. The benefits of a complete

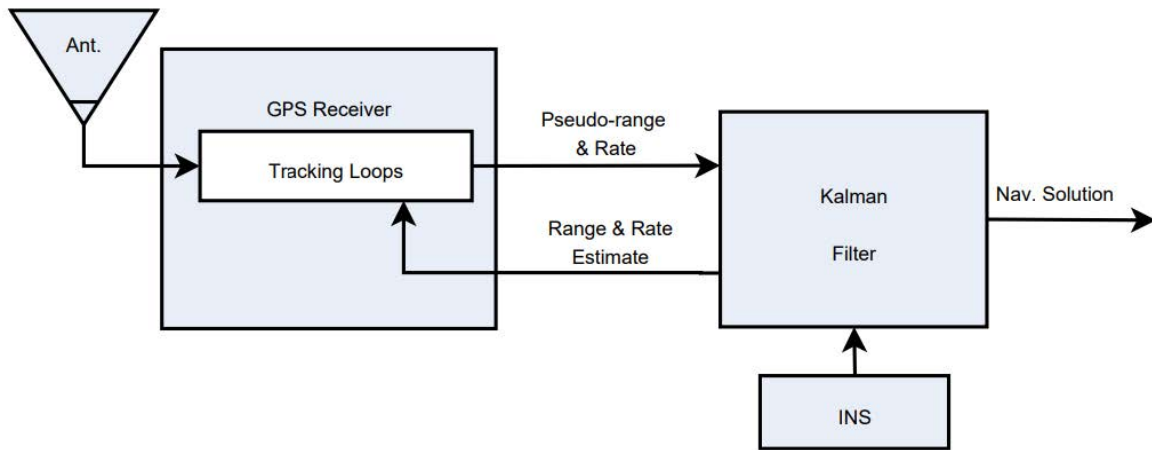


Figure 4.2: Tightly-coupled architecture between GPS and INS systems [52].

deeply-coupled system allow continued tracking of a degraded signal 2 – 6 dB higher than their scalar loop counter-parts [8]. Furthermore, the addition of external dynamics allows better predictions of receiver pose due to the navigation filter acknowledging the dynamics of the collection platform. This is especially beneficial for high-dynamic vehicles such as hyper-velocity aircraft [53] or vehicles in GPS-challenged environments [54]. For these benefits, a deep integration of the FVDM with GPS correlator-level measurements is the focus of this work.

4.1 Vector Delay and Frequency Lock Loop

Vector tracking first utilized a Vector Delay Lock Loop (VDLL) and was proposed by [55]. In a VDLL, the EKF provides continuous estimates of the code frequency, updating the DLL, improving overall tracking performance. Later on, [19], explores tracking both code frequency and carrier frequency in an EKF, coined the VDFLL. This method showed great improvements over scalar tracking algorithms and moderate improvements over the VDLL. The VDFLL proves best when tracking weaker GNSS signals under high dynamic stress [21]. Furthermore, recent analyses from [56] prove the VDFLL has improved resilience to multipath delay as well. A block diagram of the VDFLL is shown in Figure 4.3.

From Figure 4.3, the *RF Front End* block refers to the discussion in Section 3.1. The signal correlation blocks represent Equation 3.1 and also the the FLL and DLL discriminators specified in Equations 3.5 and 3.7, respectively. The basic flow of the VDFLL requires the receiver

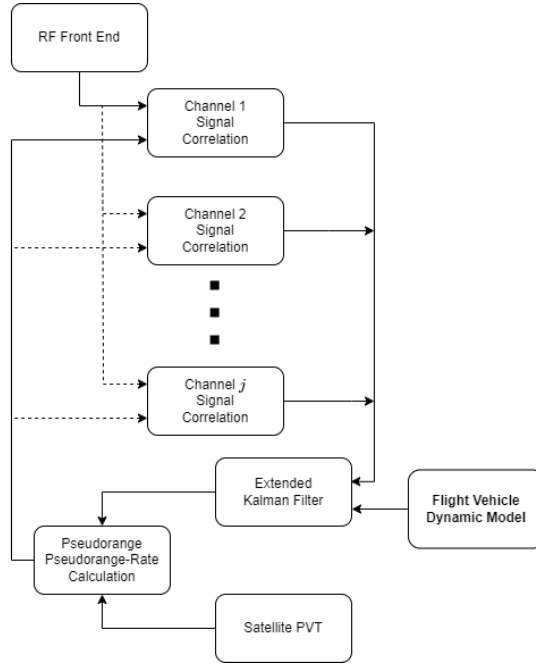


Figure 4.3: Block diagram of the VDFLL used in this work (Adapted from [9]).

to know its position and the positions of the satellites *a priori*. Preferably, initial positions and satellite positions that are fed into the VDFLL are from processing the received signal for a length of time required to decode the navigation message using an open-loop architecture like the scalar tracking loops and WNLS discussed in Chapter 3. The measurement inputs of the EKF in Figure 4.3 are residual pseudorange and pseudorange-rate errors in the form of discriminator outputs. The EKF uses the measurements from the current signal correlations to directly estimate the pose of the receiver. Using the ephemeris of the satellites and the corrected position estimates, new code phase and carrier frequency estimates are generated for the next integration period. To improve the estimated position from the EKF, the FVDM is used as the process model to propagate the non-linear motion of the aircraft in time. The next section covers the time update and measurement update within the EKF.

4.2 Deeply Coupled GPS and FVDM Navigation Filter

As stated previously, the VDFLL replaces the scalar DLLs and FLLs with a single EKF. This section describes the design of the EKF for the proposed navigation filter. The EKF for this work represents a position-state filter where the state vector is defined by Equation 4.1.

$$\mathbf{X} = \left[\mathbf{X}_V \quad \mathbf{X}_\omega \quad \mathbf{X}_P \quad \mathbf{X}_\psi \quad \mathbf{X}_t \right]^T \quad (4.1)$$

The essential elements for the state vector are sectioned into five terms. \mathbf{X}_V (Equation 4.2) describes the velocity states of the aircraft from Earth to body with respect to the Local Navigation frame.

$$\mathbf{X}_V = \begin{bmatrix} V_N & V_E & V_D \end{bmatrix} \quad (4.2)$$

V_N , V_E , and V_D are the velocity components of the aircraft in the North, East, and Down directions, respectively. The angular rates (\mathbf{X}_ω) are represented from inertial to body with respect to the body frame (Equation 4.3).

$$\mathbf{X}_\omega = \begin{bmatrix} p_{ib}^b & q_{ib}^b & r_{ib}^b \end{bmatrix} \quad (4.3)$$

p_{ib}^b , q_{ib}^b , and r_{ib}^b are the angular rates components of the aircraft about the body-fixed x, y , and z axes. The position estimates of the aircraft are from Earth to body with respect to the Local Navigation frame (Equation 4.4), similar to the velocity states.

$$\mathbf{X}_P = \begin{bmatrix} \lambda_{eb}^n & L_{eb}^n & h_{eb}^n \end{bmatrix} \quad (4.4)$$

λ_{eb}^n , L_{eb}^n , and h_{eb}^n are the latitude, longitude, and altitude of the aircraft with respect to the geodetic reference frame described in Chapter 2. Equation 4.5 represents the Euler angles of the aircraft, represented from body to the local navigation frame.

$$\mathbf{X}_\psi = \begin{bmatrix} \phi_{bn} & \theta_{bn} & \psi_{bn} \end{bmatrix} \quad (4.5)$$

ϕ_{bn} , θ_{bn} , and ψ_{bn} are the roll, pitch, and yaw of the aircraft from the body to the navigation frame. Completing the state vector are the clock terms that represent estimates of the clock bias and clock drift of the receiver during flight (Equation 4.6). The clock bias and clock drift are scaled by the speed of light (c) to give them units of meters and meters/second, respectively.

$$\mathbf{X}_t = \begin{bmatrix} c\delta t & c\delta t \end{bmatrix} \quad (4.6)$$

The dynamics of the aircraft are defined by

$$\dot{\mathbf{X}} = F(\mathbf{X}, f_{ib}^b, M_{ib}^b) + \mathbf{B}_{dyn}w_{dyn} + \mathbf{B}_{clk}w_{clk}, \quad (4.7)$$

where \mathbf{B}_{dyn} is the noise distribution matrix related to the dynamics (Equation 4.8).

$$\mathbf{B}_{dyn} = \begin{bmatrix} \frac{\Delta t^2}{2} \mathbf{I}_{6,6} \\ \Delta t \mathbf{I}_{6,6} \\ \mathbf{0}_{2,2} \end{bmatrix}_{14,8} \quad (4.8)$$

w_{dyn} is the disturbance vector for the aircraft dynamics, as shown in Equation 4.9. Equation 4.8 shows that only the linear and angular accelerations are affected by the disturbances, but these errors trickle down into the kinematic equations for the position and Euler derivatives.

$$w_{dyn} = \left[\sigma_{V_N}^2 \quad \sigma_{V_E}^2 \quad \sigma_{V_D}^2 \quad \sigma_p^2 \quad \sigma_q^2 \quad \sigma_r^2 \right]^T \quad (4.9)$$

σ_{\cdot} are additive white Gaussian noise that characterizes the stochastic motion of the aircraft for their respective, sub-scripted variables. The discretized process noise covariance matrix is a function of the time step and phase and frequency clock variances for the specified oscillator (Equation 4.10).

$$\mathbf{Q}_{clk} = \begin{bmatrix} \sigma_b^2 \Delta t + \sigma_{br}^2 \Delta t^2 + \sigma_r^2 \frac{\Delta t^3}{3} & \sigma_{br}^2 \Delta t + \sigma_r^2 \frac{\Delta t^2}{2} \\ \sigma_{br}^2 \Delta t + \sigma_r^2 \frac{\Delta t^2}{2} & \frac{\sigma_r^2}{\Delta t} + \sigma_{br}^2 + \frac{4}{3} \sigma_r^2 \Delta t \end{bmatrix} \quad (4.10)$$

σ_b^2 is the variance of the noise process that drives the clock phase error in m^2/s , σ_r^2 is the noise variance that drives the clock frequency error in m^2/s^3 and σ_{br}^2 is the variance on the noise process that drives both the phase and frequency error in units of m^2/s^2 . For the receiver simulated in this work, an Oven Controlled Crystal Oscillator (OCXO) is used. More

information on calculating the clock variance based on oscillator type can be found in [57]. The time step (Δt) is 200 Hz.

F is a set of non-linear equations that define the motion of the aircraft as a function of the current state in time (\mathbf{X}), the forces in the body frame (f_{ib}^b) and the moments about the body frame (M_{ib}^b). Calculation of the forces and moments were discussed in Chapter 2.

Calculations of the forces and moments and moments must be done in the body frame. Rotating all of the equations that build the total forces and moments acting onto the airframe into a global or local navigation reference frame would be cumbersome and introduce more complexity than necessary. However, because of the measurements generated from the correlators and discriminators are composed in the ECEF reference frame, the FVDM must be propagated with respect to the curvature of the Earth in a global frame. For this work, the equations of motion are rotated into the local navigation frame.

The state derivatives of the velocity components are defined in Equation 4.11.

$$\begin{bmatrix} \dot{V}_N \\ \dot{V}_E \\ \dot{V}_D \end{bmatrix} = \mathbf{C}_b^n \frac{\mathbf{f}_{ib}^b}{m} - (2\boldsymbol{\Omega}_{ie}^n - \boldsymbol{\Omega}_{en}^n) \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} \quad (4.11)$$

The first term above represents the forces acting onto the airframe with respect to the body frame divided by the mass of aircraft, m . For the purposes of this work, the mass of the aircraft is assumed constant. This specific force vector is rotated into the local navigation frame using \mathbf{C}_b^n , defined in Equation 2.9. The latter term in Equation 4.11 represents the rotation rate (Equation 4.12) of the Earth in skew-symmetric form and the transport rate (Equation 4.13) in skew-symmetric form, both rotated into the local navigation frame and multiplied by the current velocity of the aircraft.

$$\boldsymbol{\Omega}_{ie}^n = \omega_{ie} \begin{bmatrix} 0 & \sin(L_{eb}^n) & 0 \\ -\sin(L_{eb}^n) & 0 & -\cos(L_{eb}^n) \\ 0 & \cos(L_{eb}^n) & 0 \end{bmatrix} \quad (4.12)$$

Where ω_{ie} is 7.27×10^{-5} radians/second.

$$\mathbf{\Omega}_{en}^n = \begin{bmatrix} 0 & -\omega_{en,z}^n & -\omega_{en,y}^n \\ \omega_{en,z}^n & 0 & -\omega_{en,x}^n \\ -\omega_{en,y}^n & \omega_{en,x}^n & 0 \end{bmatrix} \quad (4.13)$$

Where

$$\omega_{en}^n = \begin{bmatrix} V_E/(R_E + h_{eb}^n) \\ -V_N/(R_N + h_{eb}^n) \\ V_E \tan(L_{eb}^n)/(R_E + h_{eb}^n) \end{bmatrix} \quad (4.14)$$

Above, R_E and R_N refer to the meridian and transverse radii of curvature as described in Equations 2.1 and 2.2, respectively.

The derivatives of the angular rates are defined as

$$\begin{bmatrix} \dot{p}_{ib}^b \\ \dot{q}_{ib}^b \\ \dot{r}_{ib}^b \end{bmatrix} = \mathbf{I}_{cg}^{b-1} \left[\mathbf{M}_{ib}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\mathbf{I}_{cg}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \right], \quad (4.15)$$

where \mathbf{I}_{cg}^b are the mass moments of inertia for the aircraft. For the purpose of this work, the aircraft is modelled symmetrically about each of the axes such that \mathbf{I}_{cg}^b only has terms along the diagonal.

The local navigation frame position derivatives are described by Equation 4.16.

$$\begin{bmatrix} \dot{\lambda}_{eb}^n \\ \dot{L}_{eb}^n \\ \dot{h}_{eb}^n \end{bmatrix} = \begin{bmatrix} \frac{V_N}{R_N + h_{eb}^n} \\ \frac{V_E}{(R_E + h_{eb}^n) \cos(L_{eb}^n)} \\ -V_D \end{bmatrix} \quad (4.16)$$

The derivatives of the Euler angles are seen in Equation 4.17

$$\begin{bmatrix} \dot{\phi}_{bn} \\ \dot{\theta}_{bn} \\ \dot{\psi}_{bn} \end{bmatrix} = \mathbf{C}_\omega \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} - \mathbf{C}_n^b (\omega_{ie}^n + \omega_{en}^n) \right) \quad (4.17)$$

Calculation of the Euler rates is the difference between the current angular rates of the aircraft and the rotation of the Earth along with the transport rate, similar to the calculation of linear acceleration in Equation 4.11. This difference is rotated by \mathbf{C}_ω defined by Equation 4.18.

$$\mathbf{C}_\omega = \begin{bmatrix} 1 & \tan(\theta) \sin(\phi) & \tan(\theta) \cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \quad (4.18)$$

The last of the state derivatives are the clock terms. Both the clock drift and clock drift rate are scaled by the speed of light to give them units of $m s^{-1}$ and $m s^{-2}$.

$$\begin{bmatrix} c\delta\dot{t} \\ c\delta\ddot{t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c\delta t \\ c\delta\dot{t} \end{bmatrix} \quad (4.19)$$

Once the state derivatives are calculated using the aforementioned equations of motions, they are integrated using Euler integrations to propagate the states forward in time. This provides the EKF with the predicted states for the current time step.

The other part of the prediction step in the EKF is the formation of the predicted covariance matrix \mathbf{P}_k^- . This is defined by Equation 4.20.

$$\mathbf{P}_k^- = \Phi \mathbf{P}_{k-1}^- \Phi^T + \mathbf{Q}_d \quad (4.20)$$

Φ is defined as the state transition matrix. The state transition matrix is composed of \mathbf{X} and the relationship with each state derivative $\dot{\mathbf{X}}$. This done by taking the Jacobian and is represented by Equation 4.21.

$$\mathbf{J} = \frac{\partial \dot{\mathbf{X}}_{14,1}}{\partial \mathbf{X}_{14,1}} \quad (4.21)$$

The evaluated Jacobian, \mathbf{J} , is a square 14 row, 14 column matrix that varies as function of the forces and moments in time. These forces and moments can vary based on the disturbances and the controls inputs, so \mathbf{J} must be evaluated at every time step. Because of the complexity of the Jacobian, it is solved using the symbolic toolbox in MATLAB.

The state propagation is continuous, but Φ is discrete. This means that the Jacobian must be discretized. The discretization of the Jacobian from continuous to discrete is introduced by Equation 4.22

$$\Phi = \text{expm}(\mathbf{J}\Delta t) \quad (4.22)$$

The discrete process noise covariance (\mathbf{Q}_d) encapsulates the disturbances onto the aircraft dynamics such that the EKF can better correct the states during the measurement update. To transform the process noise from continuous to discrete, Equation 4.23 is used.

$$\mathbf{Q}_d = \text{blkdiag}([\Phi \mathbf{B}_{dyn} (w_{dyn} w_{dyn}^T) \mathbf{B}_{dyn}^T \Phi^T \Delta t], \mathbf{Q}_{clk}) \quad (4.23)$$

Once the state and covariance prediction are calculated, the *a priori* part of the EKF is complete. The EKF will continue to predict the state and covariance until measurements from the vector tracking receiver are available. The next subsection covers the measurement update of the EKF, correcting the predicted covariance and predicted states.

4.2.1 Update *a posteriori*

The measurement update in the EKF utilizes the current predicted covariance and measurement covariance to optimally correct the predicted states and predicted covariance. Equation 4.24 shows the calculation of the Kalman gain.

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (4.24)$$

The Kalman gain, \mathbf{K} , is a function of the observation matrix, \mathbf{H} , measurement covariance matrix, \mathbf{R} , and the predicted covariance, described previously. The observation matrix maps the residual pseudorange and pseudorange rates to the local navigation states. This is demonstrated by Equation 4.25.

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{u}_{1,3}^{n,j} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & -1 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{h}_{\rho p 1,3}^j & \mathbf{0}_{1,3} & -1 & 0 \end{bmatrix}_{2j,14} \quad (4.25)$$

Using the predicted estimate of the receiver position Euler attitude, \mathbf{H} rotates the ECEF residuals to the local navigation frame. There are arrays of zeros where the measurements do not correlate to the corresponding states. The rotation of the unit vectors from ECEF to the local navigation frame is shown by Equation 4.26 and the rotation for the position measurements is given by Equation 4.27.

$$\mathbf{u}^{n,j} = \mathbf{C}_e^n \mathbf{u}^{e,j} \quad (4.26)$$

$$\mathbf{h}_{\rho p}^j = \begin{bmatrix} (R_N + h_{eb}^n) \mathbf{u}_N^{n,j} \\ (R_E + h_{eb}^n) \cos(L_{eb}^n) \mathbf{u}_E^{n,j} \\ -\mathbf{u}_D^{n,j} \end{bmatrix} \quad (4.27)$$

The measurement covariance matrix, \mathbf{R} , was described as the weighting matrix, \mathbf{W} , in Chapter 3. The variances of the measurements are still calculated as a function of the carrier-to-noise ratio. Once the Kalman gain is calculated for the current time step, it can be used with the measurement vector, Δz , to update the predicted state estimate (Equation 4.28 and 4.29).

$$\Delta z = \begin{bmatrix} \Delta \dot{\rho}^1 & \Delta \rho^1 & \dots & \Delta \dot{\rho}^j & \Delta \rho^j \end{bmatrix}^T \quad (4.28)$$

$$\mathbf{X}_k^+ = \mathbf{X}_k^- + \mathbf{K}_k \Delta Z \quad (4.29)$$

The last step in the measurement update is correcting the predicted covariance (Equation 4.30). The predicted covariance will drift based on the process noise covariance matrix and correcting the covariance by using the newly calculated Kalman gain will inform the filter of the confidence in the process model.

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (4.30)$$

After the measurement update of the EKF is complete, the estimated states and covariance matrices redefine the predicted state and covariances, closing the loop in the proposed navigation filter.

4.3 Conclusions

Vector tracking is a closed loop solution to estimating navigation states in a GPS receiver. The proposed navigation filter augments the existing VDFLL architecture by instantiating the FVDM as the process model in the EKF. This chapter covered the differences between scalar and vector tracking loops and expanded on the utilization of the FVDM to propagate the states of aircraft while also using correlator-level GPS measurements to correct the states and maintain channel lock for the duration of the simulations. This chapter provided a general overview of the VDFLL and covered the additions to the VDFLL architecture that did not exist before. For a more nuanced implementation of the VDFLL, the reader is asked to read the sources cited throughout the chapter for more information.

Chapter 5

Simulation Environment

Up until this point, this thesis has covered the navigation architecture in the overarching Guidance, Navigation, and Control (GNC) flight vehicle model. This chapter will expand on both the guidance and control architecture utilized for this work. Although these are not the primary focus of this thesis, they are important to discuss if others wish to replicate the study.

5.1 Guidance System

The guidance system for a flight vehicle holds responsibility for providing reference points to the controllers that provide actuation of the control surfaces and engine controls. These reference points are a function of various flight conditions including the atmospheric conditions, estimated pose of the aircraft, and engine efficiency. For the purposes of this work, the primary goal of the guidance system is to accept a user-defined trajectory consisting of waypoints defined in the local navigation frame and convert these into aircraft heading, pitch, throttle, and propeller pitch reference commands for the controllers to interpret. The following subsection provides an overview of creating trajectories for the guidance system.

5.1.1 Waypoint Generation

In this work, the generation of waypoints is a simple process where the user manually picks a trajectory using Google Earth [58]. Once a trajectory is created (Figure 5.1) and saved as a *kml* file, it can be brought into MATLAB and converted into a matrix of values in the local navigation frame. Note, at this point, it may be beneficial to alter the altitude values such that

the aircraft climbs, descends, or maintains a certain altitude during the length of the simulation. The conversion of the *kml* file can be done using the *KML Toolbox* from [59].

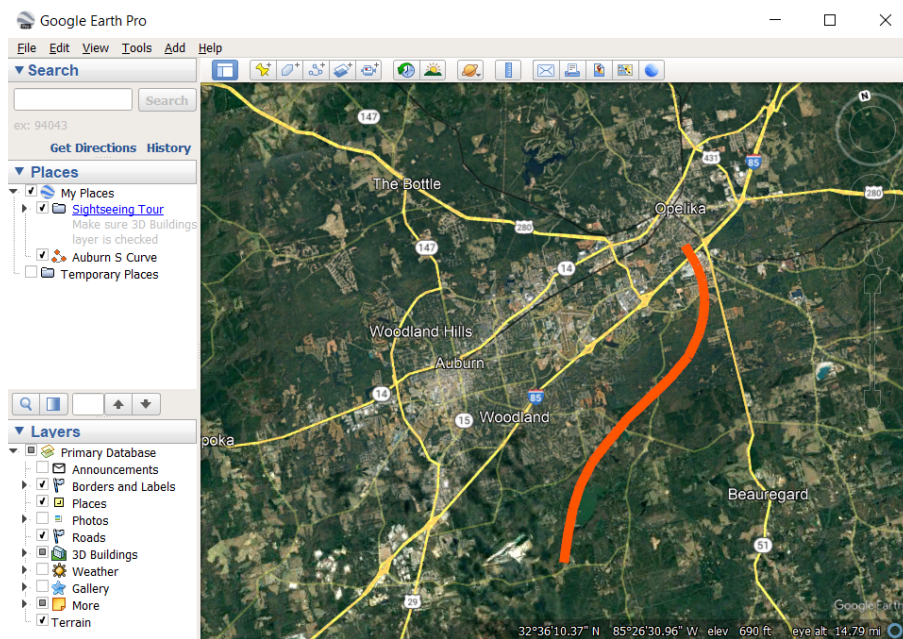


Figure 5.1: A custom trajectory created in Google Earth for the guidance system discussed in this work.

Once the trajectory is imported into the model, it is used within the guidance system to produce reference commands of desired heading, pitch, throttle, and propeller pitch. This work uses the *uavWaypointFollower* class provided by the UAV Toolbox in MATLAB. Other methods of waypoint following can be used, as long as they output the aforementioned reference commands.

The waypoint follower calculates the reference heading and pitch by using trigonometric relationships between the flight vehicles current pose and the selected waypoints position. The throttle is chosen such that time between waypoints is consistent. The propeller pitch is controlled such that propeller efficiency is maximized (Equation 2.37). These reference commands are passed to the control law, mapping commanded values to normalized aircraft stick movements, throttle lever position, and propeller lever position. This processed in discussed in the next section.

5.2 Control Scheme

As stated previously, the guidance system and control law are not the primary focus of this work. Because of this, classical Proportional-Integral-Derivative (PID) controllers are used to actuate the controls surfaces to the desired values from the guidance system. During simulations presented for this thesis, only four of the available eleven inputs are being controlled to maintain the desired trajectory of the aircraft. Table 5.1 lays out the available inputs available for the Diamond-DA40 modeled in this work.

Table 5.1: List of available, controllable inputs to the Diamond DA40 modeled in this thesis.

Input	Definition	Controlled
Lateral Stick	L/R movement of pilot stick, maps to δ_a	X
Longitudinal Stick	Fwd/Aft movement of pilot stick, maps to δ_e	X
Rudder Pedals	In/Out position of left/right rudder pedals, maps to δ_r	
Throttle Lever	Lever % w.r.t. throttle, maps to engine RPM	X
Propeller Lever	Lever % w.r.t. propeller pitch normal to freestream	X
Mixture Lever	Lever % w.r.t. fuel-to-air ratio within engine	
Left Brake	Position of L rudder pedal when a/c is on ground	
Right Brake	Position of R rudder pedal when a/c is on ground	
Aileron Trim	δ_a needed to maintain a/c stability	
Rudder Trim	δ_r needed to maintain a/c stability	
Elevator Trim	δ_e needed to maintain a/c stability	

Control of the lateral stick is a function of two closed-loop controllers based on the commanded heading from the guidance system and the current heading and roll angle of the aircraft. It is assumed that the nose of the aircraft always points in the direction of the aircraft's velocity. Figure 5.2 shows the two closed-loop controllers working in series with their listed gains for more detail.

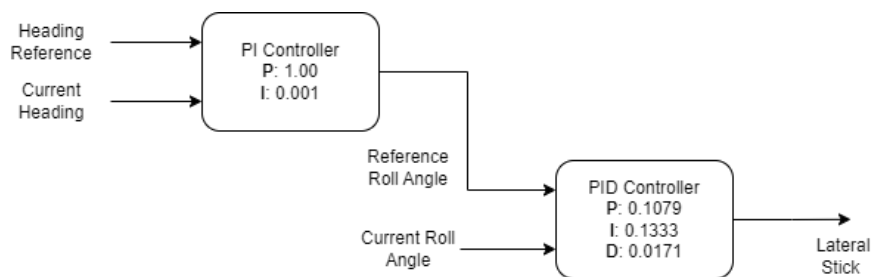


Figure 5.2: Block diagram of lateral stick control through two closed-loop, PID controllers.

The movement of the pilot stick in the longitudinal direction is a function of the commanded altitude from the guidance systems and the current altitude and pitch of the aircraft. Two closed-loop PID controllers are used in series to control the pilot stick that actuates the elevator to the correct deflection angle. Figure 5.3 shows a block diagram of the two controllers with their proportional, integral, and derivate gains.

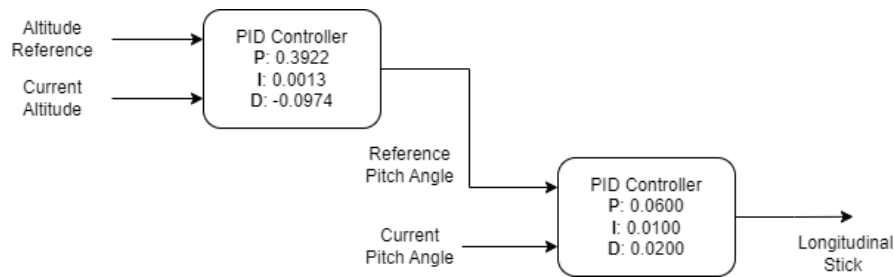


Figure 5.3: Block diagram of longitudinal stick control through two closed-loop, PID controllers.

The throttle and propeller lever controllers use a PID and PI controller to correct their inputs (Table 5.2). The throttle control is based on the commanded and current airspeed of the aircraft where the propeller lever is controlled based on maintaining at least 93% propeller efficiency. The airspeed for the throttle control is defined as the magnitude of the aircraft's velocity.

Table 5.2: Gains for the throttle and propeller lever closed-loop PID controllers.

Controller	Proportional	Integral	Derivative
Throttle Lever	0.498	0.098	-0.184
Propeller Lever	0	1	0

The controllers provide inputs such that actuation of the control surfaces are deflected and can guide the aircraft in the desired direction at the desired speed while maintaining lower fuel consumption. With the implementation of the guidance system and control law, the full GNC loop is now complete. The following section describes the disturbances modeled on the aircraft so that controller inputs and outputs are different when running Monte-Carlo simulations for further analysis of the proposed navigation filter architecture.

5.3 Disturbance Modeling

As specified in Equation 4.7, the dynamics of the aircraft are disturbed by external forces that changes the behavior of the aircraft between simulations of the same trajectory. Ultimately, these changes lead to different control inputs and allow the Monte-Carlo analyses to emphasize the improvements of the proposed FVDM navigation filter.

For the study presented in this thesis, two disturbances are modeled – a wind model and uncertainties in the current altitude. The wind model is represented as a zero-mean white noise process from [14]. Changes in the wind are important to simulation as wind greatly affect aircraft trajectories in reality. The modeling of wind as a white noise process provides a stochastic process that more closely resembles wind data collected from actual data collections [16]. In Chapter 2, it is clear that the pressure, temperature and density calculation stem from the ISA model; however, aircraft do not use this model on board. To simulate changes in these atmospheric parameters, measurements from a Pitot tube and temperature sensor are modeled as a zero-mean white noise process from [60]. Similar to the white-noise wind model, the measurements of pressure, temperature, and density must be variant enough to have different control inputs over the course of many Monte-Carlo simulations. The variance in these white noise models can be substituted for their respective variables in \mathbf{Q}_d (Equation 4.23).

5.4 Conclusions

The guidance systems and control laws provide the aircraft with meaningful control surface deflection that guide the aircraft to the desired destination. This chapter provided an overview about how the guidance system generates commanded control inputs from a user-defined trajectory created using Google Earth. Following the guidance system, the control law used in this thesis was covered with discussions about the PID control gains used for each of the four piloted control inputs during simulation. Finally, a discussion of the disturbances on the aircraft dynamics that influence different control inputs between simulations was described. The focus of this work is not on the guidance or controls of the flight vehicle presented in this thesis, for

better detail of each the sections discussed during this chapter, the reader is asked to follow-up with the references cited above for more information.

Chapter 6

Scenario Implementation and Results

This chapter presents results of two trajectories using the proposed navigation filter and its underlying components discussed in previous chapters. Each trajectory is subject to varying degrees of signal interference across multiple Monte-Carlo simulations. First, an overview of the Monte-Carlo analyses that will be presented for each trajectory is provided. Next, a discussion of the first trajectory and configuration file used for the simulations is presented. Following a description of trajectory one, Monte-Carlo analyses highlight the strong and weak points of the proposed navigation filter. Once the analyses of the first trajectory conclude, a description of trajectory two is presented, followed by a similar Monte-Carlo statistical analysis. Each Monte-Carlo case is composed of 100-run simulations for each trajectory subject to seven cases of signal degradation. To show the improved performance of the proposed navigation filter, a constant-velocity, kinematic model VDFLL [9] is shown as a comparison. This standard VDFLL processes the same trajectories at the same cases of signal degradation.

6.1 Monte-Carlo Analyses

Originally named after a famous casino in Monaco, the Monte-Carlo analysis is a model to predict the probability of an outcome given the presence of random variables [61]. For the deeply-coupled FVDM, a number of random variables exists. For one, the true states of the aircraft are disturbed by wind, causing the aircraft to veer from the desired target line, that is, between simulations, the simulated aircraft will not fly the same flight path every time, regardless of having the same set of waypoints. For the purpose of this work, these wind disturbances

are modeled with the same magnitude – interference and trajectory notwithstanding. Within the VT algorithms, the correlator simulator features noise that occurs at every measurement step, meaning no correlators (Equation 3.1) will be the same between simulations for the same time step. For the correlator simulation tool presented in this thesis, the random noise placed onto the correlators is a function of the interference level. As demonstrated by Equation 3.15, higher variance noise when compared to the amplitude of the cross-correlated local and received signal leads to an overall lower C/N_0 ratio, which is demonstrated by the results later on in this chapter.

For each of the trajectories presented in this chapter, the 100-run Monte-Carlo simulations will evaluate the two different navigation filters for their probability to track the GPS signal given C/N_0 degradation. Furthermore, Monte-Carlo analysis on code phase and carrier frequency error over a range of C/N_0 values is presented. For the pose and attitude estimation results, the Monte-Carlo simulation will provide an average Latitude, Longitude and Altitude estimate compared to the true state of the receiver, along with velocity magnitudes (speed) for the same comparison. As will be discussed later, one of the benefits in using the proposed navigation solution is the ability to estimate angular rates and Euler attitude of the aircraft during simulation. The Monte-Carlo analysis will present the average error in comparison with the true angular rates and Euler attitude as well, but only for the deeply-coupled FVDM, as the standard VDFLL implementation does not have this capability. Lastly, the Monte-Carlo analysis of the 100-run simulations will compare the clock bias and clock drift estimates from both filters compared to the true clock bias and clock drift of the embedded OCXO on board.

6.2 First Trajectory

For the first trajectory, the aircraft is simulated for a straight flight path while maintaining a constant altitude of 500 meters above sea level (Figure 6.1).

It is assumed that the initial position of the receiver is known when the simulation begins, avoiding the need to perform scalar tracking loops on the received signal data as discussed previously. For the first trajectory, the receiver aboard the aircraft is subject to seven different

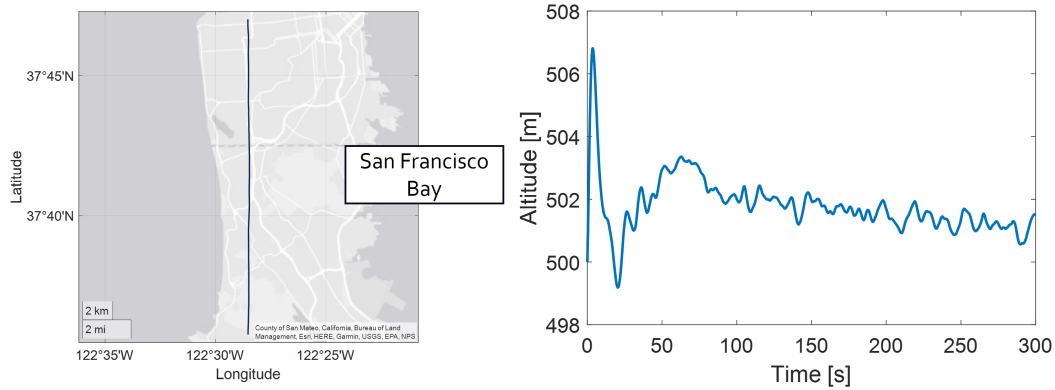


Figure 6.1: (Left) Top-view of simulated flight path for trajectory one. (Right) Altitude of flight path for trajectory one.

cases of interference that degrade the signals from the nine tracked channels (Table 6.1). The receiver is subject to these degraded power levels for the entirety of the 60 second simulations.

Table 6.1: Signal power for each case applied to each trajectory.

Case	C/N_0 [dB-Hz]
1	45
2	35
3	25
4	22
5	20
6	18
7	16
8	15
9	10
10	5
11	2

From Table 6.2, several parameters are configurable to meet the desired simulation of the user.

Date is used to pull the specified Rinex file from [62]. This Rinex file is then parsed for its ephemeris and used to propagate the GPS satellites during the simulation. Figure 6.2 shows the available satellites at the first time step for June 15, 2022 used in this work.

Duration and Monte-Carlo Runs specify the length of each simulation in seconds and the number of simulations for each scenario and/or case. For the work presented in thesis, 100 simulation Monte-Carlo runs are used for analyses. Based on [15, 14, 16] and time to

Table 6.2: Initial conditions for simulated trajectory from Figure 6.1.

Condition	Value	Units
Date	June 15, 2022	<i>DateTime</i> Object
Duration	300	<i>s</i>
Monte-Carlo Runs	100	–
Frequency	200	Hz
Trajectory	<i>StraightFlightPath.mat</i>	See Chapter 5
Velocity Disturbance	[300, 300, 0]	<i>m/s</i>
Angular Rate Disturbance	[10^{-12} , 10^{-12} , 10^{-12}]	<i>rad/s</i>
Clock Type	OCXO	–
Initial Velocity	[75, 0, 0]	<i>m/s</i>
Initial Angular Rate	[0, 0, 0]	<i>rad/s</i>
Initial Position	[0.65617, -2.1376, 500]	[<i>rad, rad, m</i>]
Initial Attitude	[0, 0, 0]	<i>rad</i>
Initial Clock Terms	[0, 0]	[<i>m, m/s</i>]
Channel C/N_0	45, 35, 25, 22, 20, 18, 16, 15, 10, 5, 2	dB-Hz

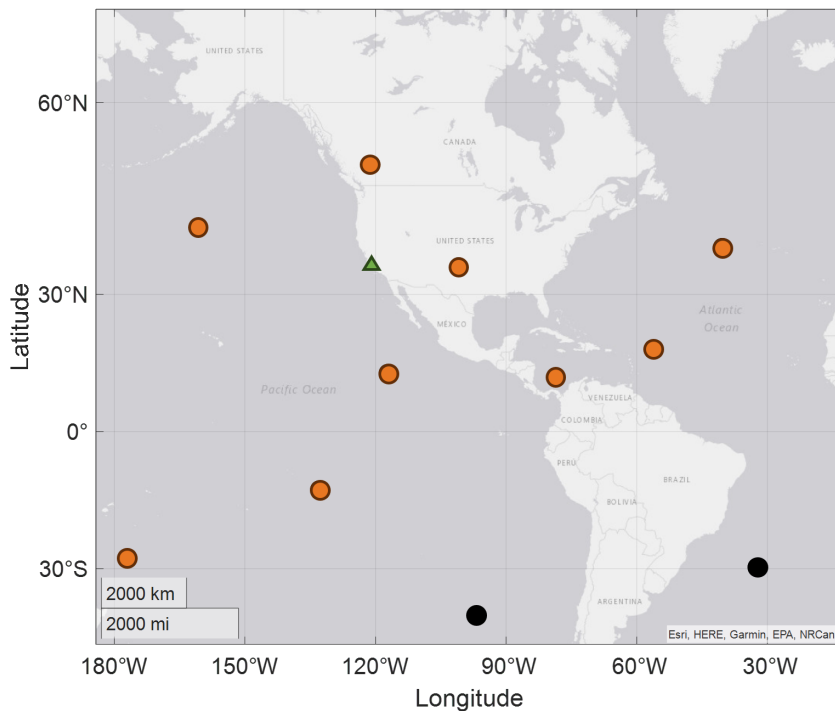


Figure 6.2: Orange dots signify satellite locations at the start of the simulations given the date of broadcast ephemeris. Black dots signify satellites that are in-view but are discarded due to the 10 degree mask angle used. The green triangle represents the initial receiver position.

completion for each simulation, 100 is enough to show the general trend for statistical purposes.

The trajectory is specified as a -mat file. The creation of the trajectory is detailed in Chapter 5.

Trajectory one is a baseline case where it is expected that both the deeply-coupled FVDM and constant-velocity kinematic model both perform well. As explained in Chapter 5, disturbances are modeled onto the trajectory of the aircraft as the FVDM is not perfect. External conditions such as wind and various atmospheric effects alter the behavior of the Diamond DA-40 slightly. These disturbances are defined as Velocity and Angular Rate in Table 6.2. Clock Type is the embedded receiver clock modeled during the simulation. More information on the different types of clocks can be found in Chapter 4. For the purposes of this thesis, both trajectories and all cases of signal interference will use the OCXO as the embedded receiver oscillator. The initial states of the aircraft are defined by the given values. Since our trajectory specifies a mostly-north flight path, the north velocity component is specified to be 75 meters per second, while the other components are zero. Also, the pitch of the aircraft is specified as 4 degrees up, this is so the aircraft generates lift at the beginning of the simulations and does not enter a stall upon initialization. Lastly, the initial position of the aircraft is specified as the first waypoint location, for simplicity. The last configurable parameter is the initial channel C/N_0 of the available satellites.

6.2.1 Monte-Carlo Analyses

From the Monte-Carlo results, several parameters can be analyzed for the performance improvements of the proposed navigation filter over the standard VDPLL kinematic model. This section begins with a comparison of the tracking-level results of the two filters subject to different levels of signal degradation.

For the range of signal interference presented in Table 6.1, the Root Mean Square Errors (RMSE) of both the code phase and carrier frequency shows the improved performance by using the deeply-coupled FVDM in GPS-challenged environments (Figure 6.3). During simulations where the signal was slightly degraded or benign (i.e. channel power greater than 35 dB-Hz) the standard velocity implementation of the VDPLL actually performs better on average than the proposed navigation filter. This is most likely due to the FVDM becoming over confident during simulation. This is more evident from Tables 6.3 and 6.4 where the position RMSE,

Standard Deviation (STD), and maximum error from the constant velocity kinematic model are marginally better.

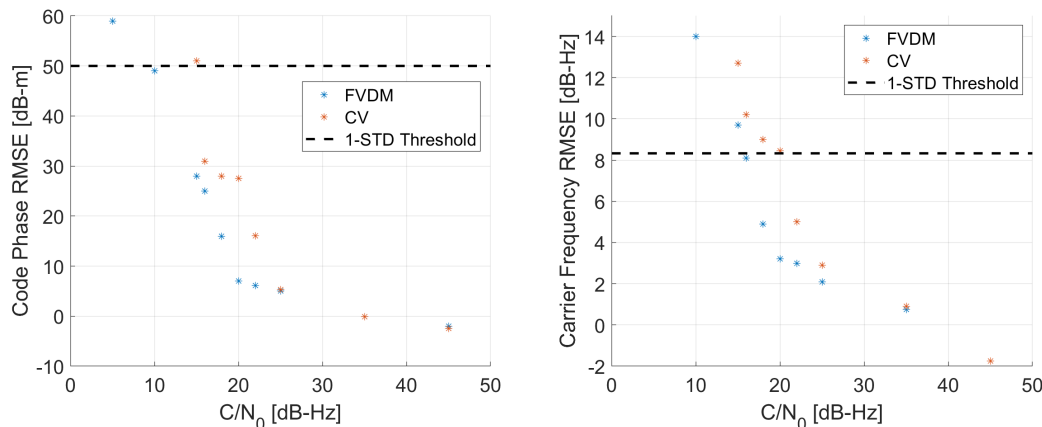


Figure 6.3: Code phase and carrier frequency RMSE as a function of signal power, specified in dB-Hz.

Table 6.3: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the deeply-coupled FVDM is subject to a degraded signal power level of 35 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	0.23443	0.066059	0.013062	0.002834
STD	0.11305	0.030674	0.17509	0.0032795
Max Error	0.72197	0.22588	0.05835	0.009

Table 6.4: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the standard VT receiver is subject to a degraded signal power level of 35 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	0.093008	0.087402	0.013897	0.0032735
STD	0.048771	0.038728	0.015186	0.0040279
Max Error	0.25751	0.26045	0.033083	0.009487

However, when the signal power is less than 35 dB-Hz, the deeply-coupled FVDM presents steady performance improvements as the interference grows stronger. For carrier frequency error, the deeply-coupled filter breaks down at roughly 16 dB-Hz, where the theoretical 8.33 dB-Hz STD from [21] is met. For the standard VDFLL implementation, this criteria is met at roughly 20 dB-Hz. The probability that the vector tracking loops are able to maintain channel lock for an entire simulation is shown in Figure 6.4.

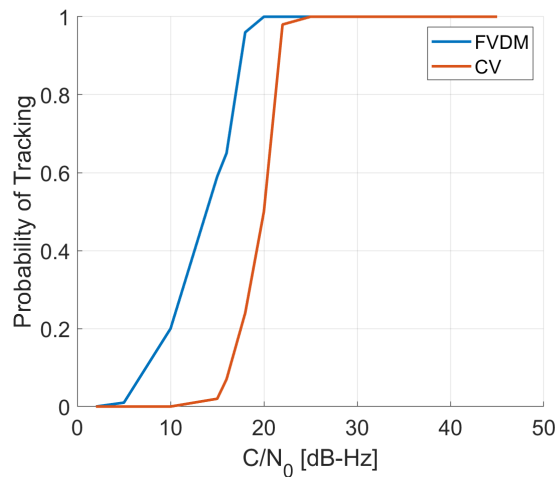


Figure 6.4: The probability that each navigation filter is able to maintain channel lock throughout the simulation across different levels of signal interference.

Based on Figure 6.4, the proposed navigation filter shows approximately 100% tracking probability 5 dB-Hz greater than the standard constant-velocity kinematic model. As stated previously, one of the benefits in utilizing the FVDM within a sensor fusion framework is the acknowledgment of aircraft behavior given a set of control inputs. This allows the presented filter to rely less on the degraded correlator measurements from GPS. The pose estimates from the navigation filter reflect the performance of the vector tracking loops to maintain lock under different levels of signal degradation (Figure 6.5).

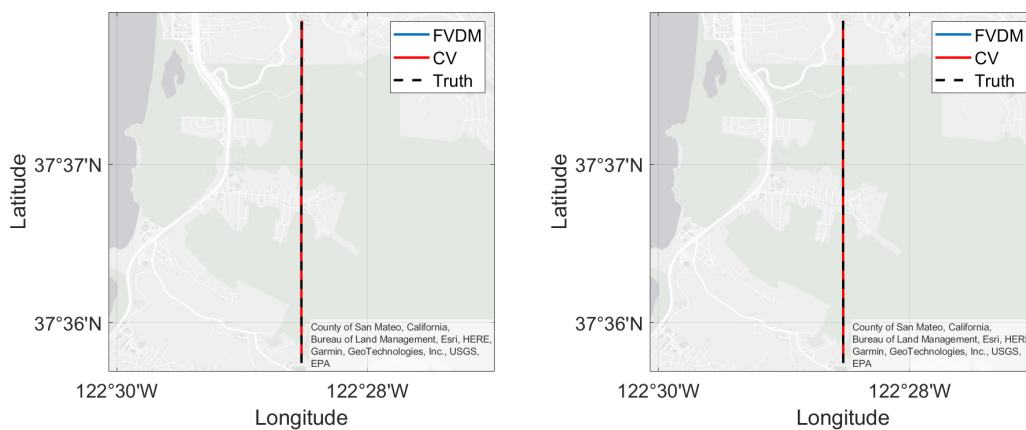


Figure 6.5: Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

Even with the straight, baseline trajectory, the standard constant velocity filter begins to drift slightly when the GPS measurements are unreliable. This is partially due to the non-linear velocity at the beginning of the simulation. However, even though this is the case, the proposed navigation filter has no problem maintaining accurate tracking estimates. Based on Figure 6.3, the deeply integrated FVDM maintains great position estimates with each channel's signal power down to 16 dB-Hz (Figure 6.6). For the position estimates from the standard VDFLL implementation, the greatest error is shown in the downward direction (Figure 6.7).

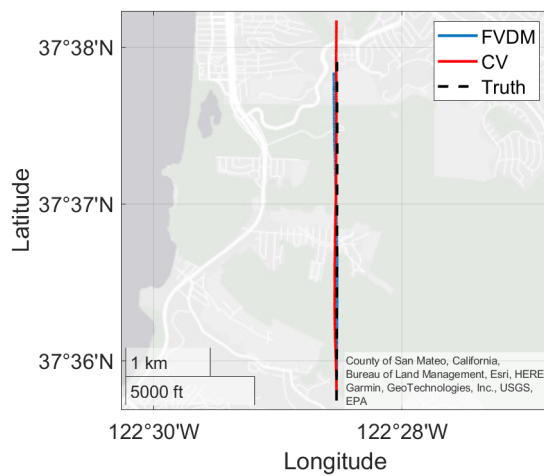


Figure 6.6: Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory when subject to a degraded signal power of 16 dB-Hz.

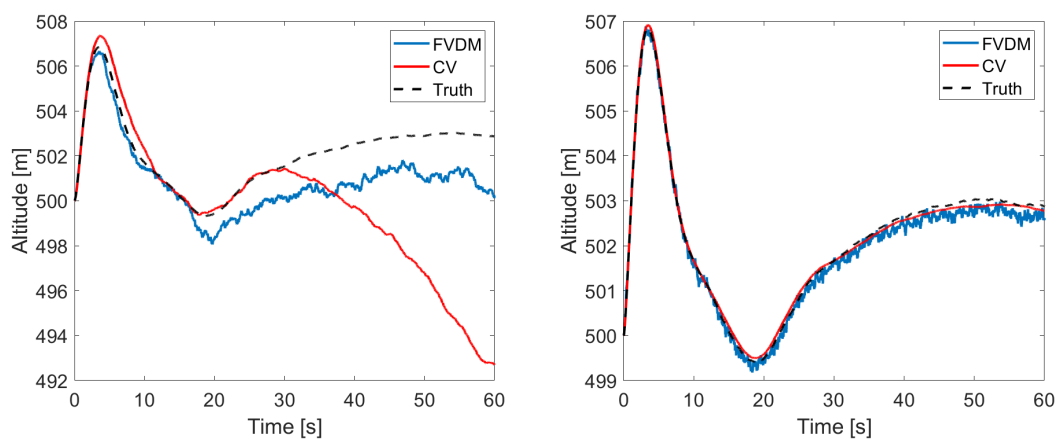


Figure 6.7: Average altitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

This is simply due to the lack of geometric diversity between the satellites sending the measurements. One way to solve this problem would be to include LEO satellites or signals of opportunity for improved altitude estimates from more diverse measurements. The poor assumption that the acceleration of the aircraft is zero-mean can be seen by the speed estimates in the constant-velocity kinematic model (Figure 6.8)

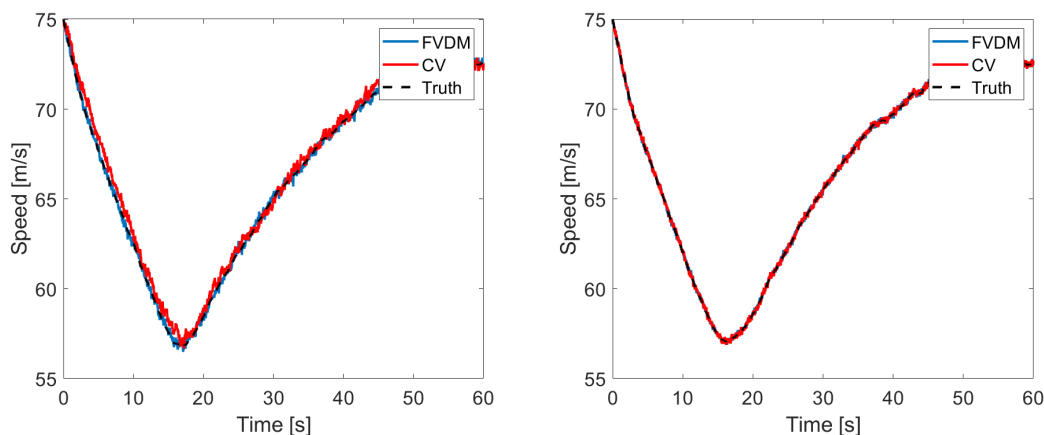


Figure 6.8: Average speed estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

When the correlator measurements from the receiver become unreliable, the constant velocity filter has no choice but to rely on its own prediction of aircraft velocity. On the fundamental basis that the velocity is the integral of a zero-mean acceleration, it leads the standard implementation to a heavily biased velocity prediction, thus deviating from the true states. Apart from the position and velocities, both filters also estimate the bias and drift of the embedded clock. As stated before, the clock used during the simulations is an OCXO. Figure 6.9 presents the average clock bias estimates for each filter at both 20 and 25 dB-Hz signal power, while Figure 6.10 presents the average clock drift estimates from each filter for the same interference cases.

The clock model utilized from [63] is the same used on both filters, so similar performance should be expected. However, the deeply-coupled navigation filter still out performs the constant-velocity EKF in both cases. The errors in clock bias and clock drift are directly linked to position and velocity estimates seen before (Figures 6.5 and 6.8). That is, the less error on the positional

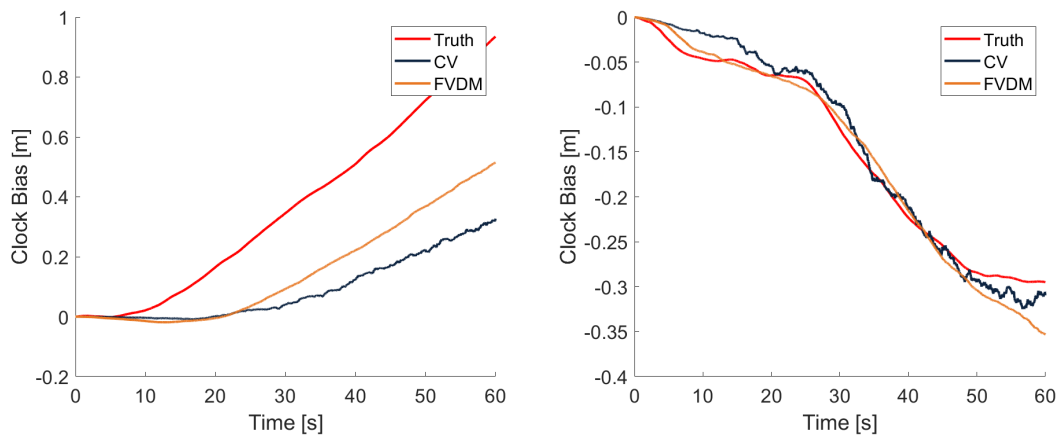


Figure 6.9: Average clock bias estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

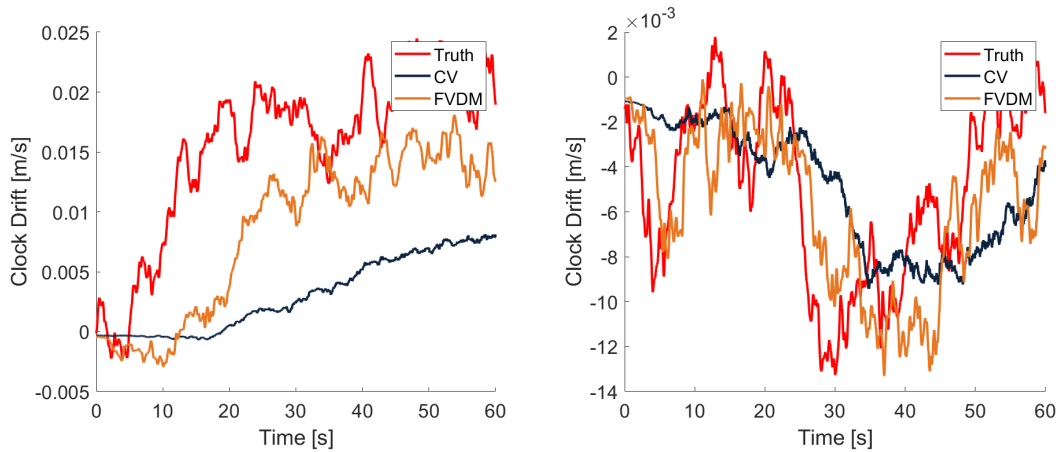


Figure 6.10: Average clock drift estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

estimates means less likelihood of more error on the clock bias estimates. The same can be said for the velocity and clock drift estimates between the two filters.

As mentioned previously, the first trajectory was a baseline, litmus test where it is expected that both the proposed navigation filter and the constant velocity kinematic model would perform similarly. Regardless of the objective for the trajectory, in cases where the signal interference left the signal power to be less than 25 dB-Hz, the deeply-coupled FVDM begins to show improved performance. This is further illustrated by Tables 6.5 and 6.6 where the RMSE, STD, and maximum error found from the 100-run Monte Carlo analysis are shown for a subjected signal power of 20 dB-Hz.

Table 6.5: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	0.83733	0.27649	0.14072	0.0087105
STD	0.32434	0.13394	0.13147	0.0072989
Max Error	1.8783	0.90375	0.48541	0.024451

Table 6.6: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	24.567	0.88208	0.17748	0.0088036
STD	10.287	0.34973	0.14068	0.005286
Max Error	36.193	2.0585	0.50134	0.022316

6.3 Second Trajectory

For the second trajectory, the aircraft is simulated for a more dynamic flight pattern while commanded to climb to an altitude of 1150 meters above sea level (Figure 6.11). The dynamics induced by trajectory two show the effectiveness of the deeply-coupled FVDM to predict the behavior of the aircraft due to the additional angular rates and Euler attitude being estimated. It is assumed that the receiver knows its position beforehand when the simulation begins. Similarly to the first trajectory, the receiver aboard the aircraft is subject to seven different cases of interference that degrade the signals from the nine tracked channels (Table 6.1). The receiver is subject to these degraded power levels for the entirety of the 60 second simulation.

It should be noted that the same cases of interference from the first trajectory still apply to trajectory two (Table 6.1). Furthermore, the only change in configuration between the first and second trajectory is the specified flight path. For this dynamic trajectory, *SCurveFlightPath.mat* will be used in place of the *StraightFlightPath.mat* used previously. The satellites found in-view utilizing the broadcast ephemeris file for the specified date are the same satellites used during the following simulations (Figure 6.2). One of the benefits (on top of improved estimate performance) of utilizing the deeply-coupled FVDM is the capability to estimate the attitude and attitude rate of the aircraft during flight. This can be imperative for acrobatic or urban air mobility aircraft that roll, pitch, and yaw as their nominal flight motion. Similar to IMU and

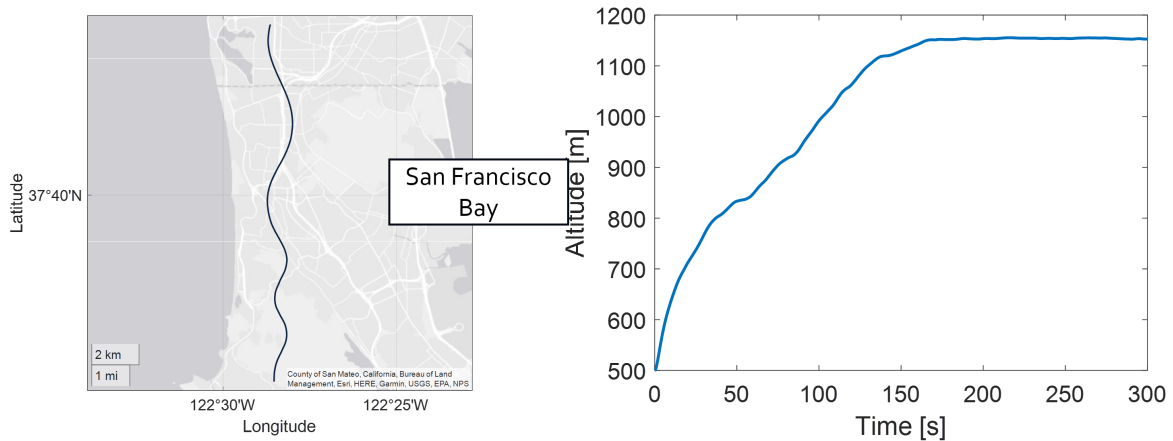


Figure 6.11: (Left) Top-view of simulated flight path for the second trajectory. (Right) Altitude of second flight path where aircraft is commanded to climb to 1150 meters and then maintain the altitude for the remainder of the simulation.

other hardware sensors, the excitation caused by the rolling and pitching motion of aircraft for trajectory two is expected to increase the performance of the position and velocity estimates along with the capability to maintain channel lock at heavier signal degradation.

6.3.1 Monte-Carlo Analyses

From Monte-Carlo results, several parameters can be analyzed for the performance improvements of the proposed navigation filter over the standard VDFLL kinematic model. This section begins with an analysis of the signal-level results between the two filters and follows with an analysis of the state estimate performance for a range of C/N_0 values. For the range of signal interference presented in table 6.1, the RMSE of both the code phase and carrier frequency shows improved performance by using the deeply-coupled FVDM in GPS-challenged environments (Figure 6.12).

As previously mentioned, the excitation in the angular rates provides better observability for those estimated states along with their integrated Euler attitude counter parts. Compared to trajectory one, Figure 6.12 shows that the proposed navigation filter can track beyond 16 dB-Hz. Unlike the previous trajectory, even in cases of little to no interference, the deeply-coupled FVDM outperforms the standard VDFLL implementation right up to benign signal power (45 dB-Hz). This effective increase in performance is further shown in Figure 6.13 where the probability to maintain lock across all channels is shown again for trajectory two.

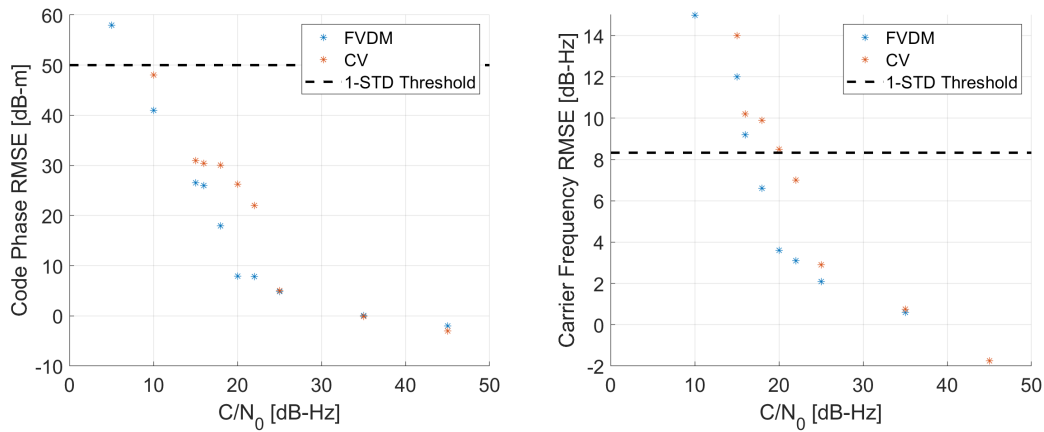


Figure 6.12: Code phase and carrier frequency RMSE as a function of signal power, specified in dB-Hz.

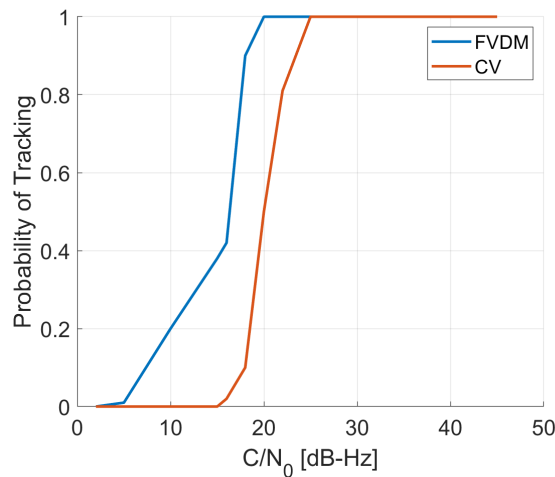


Figure 6.13: The probability that each navigation filter is able to maintain channel lock throughout the simulation across different levels of signal interference.

While the deeply integrated FVDM shows approximately 100% tracking probability at the same 20 dB-Hz compared to Figure 6.4, the constant velocity kinematic model effectively loses lock faster after the the signal power drops below 25 dB-Hz. Similar to trajectory one, the same 5 dB-Hz improvement in tracking probability can be seen for trajectory two. The position estimates reflect the improved ability of the deeply-coupled FVDM to maintain lock in GPS-challenged environments. Figure 6.14 presents the Latitude and Longitude estimates for both filters simulated for trajectory two.

When subject to a signal power of 20 dB-Hz, the average Latitude and Longitude estimates for the constant velocity EKF shows a slight drift from the true location of the aircraft. Like

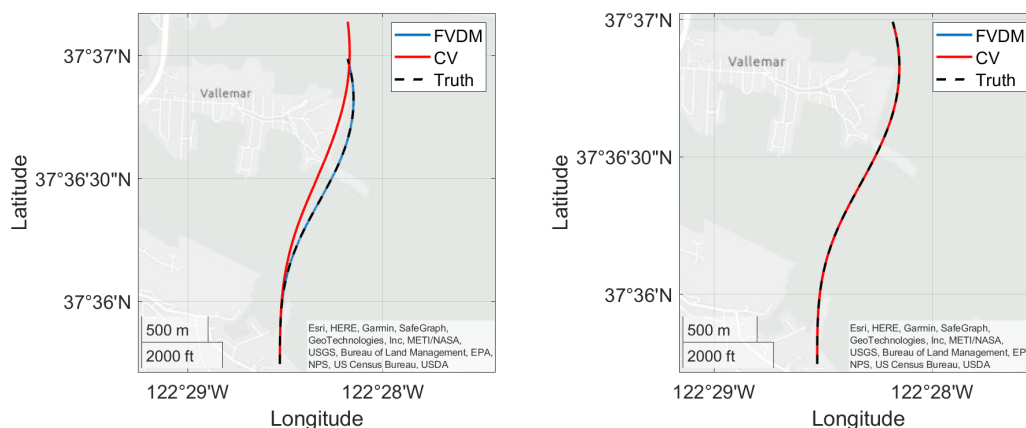


Figure 6.14: Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

before, this is likely due to the zero-mean acceleration assumption of the kinematic model. Furthermore, the standard VDFLL implementation is unable to predict the attitude of the aircraft, making it unable to correlate velocity and Euler attitude errors. For consistency, Figure 6.15 shows the Latitude and Longitude estimates when the receiver is subject to a signal power of 18 dB-Hz.

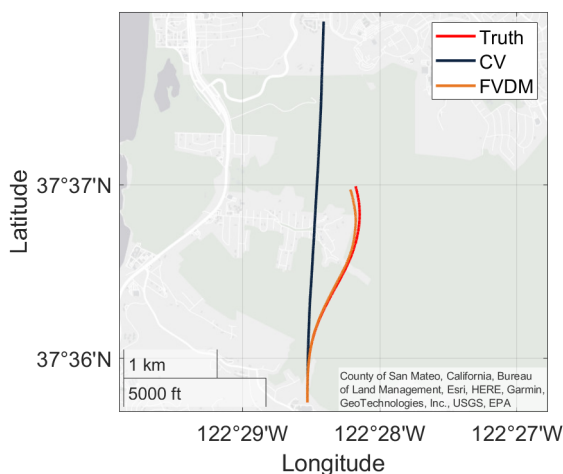


Figure 6.15: Average Latitude and Longitude of the FVDM and standard VDFLL implementation compared to the truth trajectory when subject to a degraded signal power of 16 dB-Hz.

At this level of interference, it is clear the constant velocity EKF faults just after initialization. With the lack of GPS correlator measurements to make corrections, the predicted location

of aircraft quickly falls off target. The excited pitching motion of the aircraft brings about an improvement on the altitude estimates of the aircraft during the simulation (Figure 6.16).

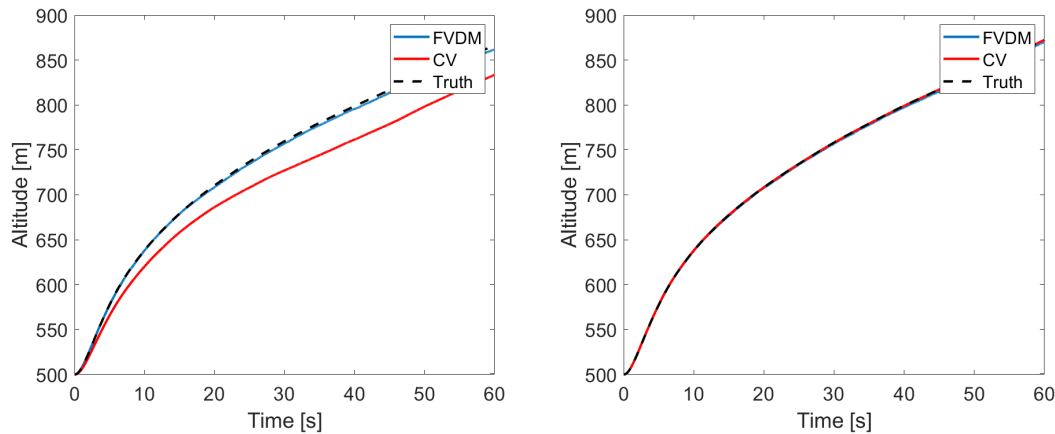


Figure 6.16: Average altitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

Again, the standard VDFLL implementation shows altitude errors of roughly 65 meters when provided with unreliable GPS correlator measurements. The lack of diversity is to blame and adding a ground station or fusing the solution with a external sensor or FVDM would mitigate this problem. For a more detailed reference on the difference in pose estimates between the two filters, Tables 6.8 and 6.7 are included at the end of this section for further inspection. The dynamics of trajectory two prove ever more that the zero-acceleration assumption from the constant velocity kinematic model is poor. Figure 6.17 presents the speed estimates of both the proposed navigation filter and the standard VDFLL implementation subject to 20 and 25 dB-Hz of signal power.

At roughly 10 m/s of error in the speed estimate (according to Table 6.8), The average drift of the estimated position presented by Figure 6.15 is understandable. It should be noted here that the throttle reference command (Chapter 5) is commanding the aircraft to keep a consistent velocity of 70 m/s , limitations on engine power and the advance ratio of the propeller prevent it from reaching such speeds during the climb segment. One of the primary benefits of the proposed navigation filter is the ability to estimate the angular rates and Euler attitude of the aircraft during flight. Although these states are not directly measurable by the correlator

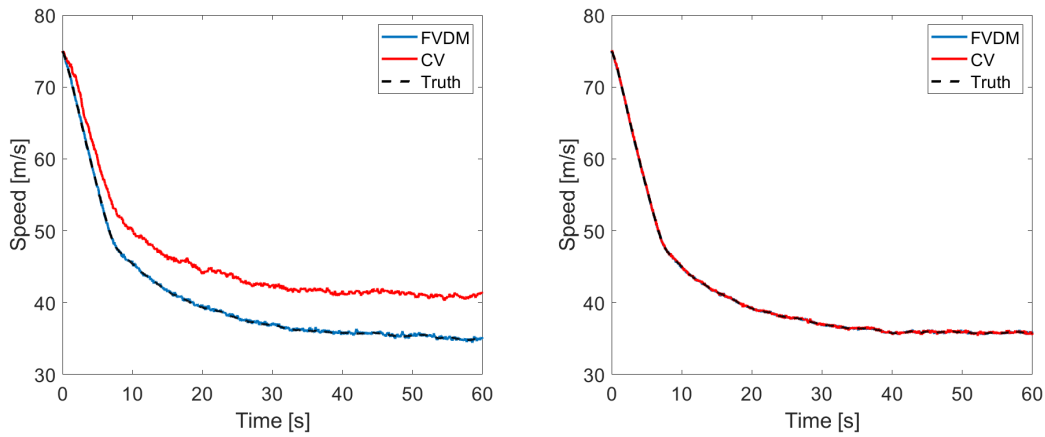


Figure 6.17: Average speed estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

measurements, their correlation within the position and velocity equations (Equations 4.16 and 4.11) is substantial enough to hold accurate estimates (Figure 6.18).

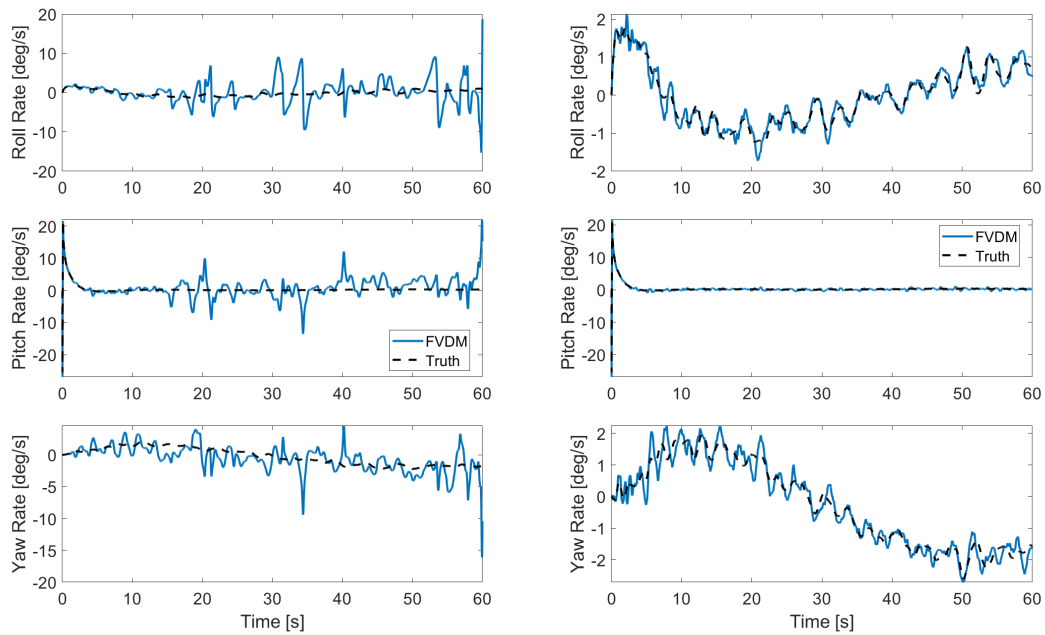


Figure 6.18: Average angular rate estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 15 dB-Hz. The right figure is with a signal power of 20 dB-Hz.

Although the aircraft is commanded to reach an altitude of 1150 meters above sea level, this does not equate to radical change in pitch rate. As seen from Figure 6.19, the aircraft reaches a steady state pitch and maintains that pitch until the altitude command is met. Since the pitch

command is near-constant, unlike the roll and yaw angles, leaves the pitch rate to be near-zero for the duration of the simulation.

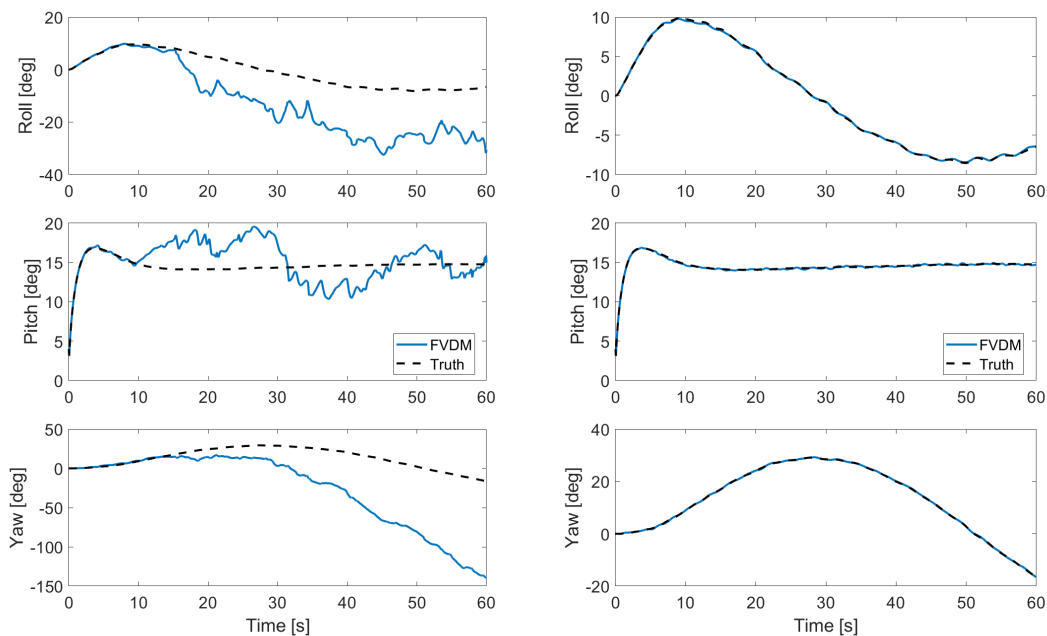


Figure 6.19: Average Euler attitude estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 15 dB-Hz. The right figure is with a signal power of 20 dB-Hz.

Because the Euler attitude estimates are the integral of the angular rate estimate, barring the rotation of C_ω (Equation 4.17), it is clear that worse estimates of angular rates lead to drifting error in the Euler attitude estimates of the aircraft. If not corrected, this can lead to drifting position and velocity estimation at lower C/N_0 levels. To maintain better estimates of the angular rates and Euler attitude, an IMU or additional hardware sensor could be tied-in with the FVDM to provide direct measurements of those states. Another option, although sub-optimal, would be the addition of a second GPS antenna, in which GPS course measurements could be calculated and provided to the filter. It should further be noted that GPS course and the yaw of the aircraft are not the same as discussed in Chapter 2.

The clock bias and clock drift estimates from both filters are analyzed for trajectory two. Similar to trajectory one, the receiver in trajectory two is embedded with an OCXO. And as mentioned previously, the performance of the clock bias and drifts estimates correspond with the accuracy of the position and velocity estimates, respectively. Figure 6.20 presents the clock bias

estimates for both filters subject to the dynamics of trajectory two and signal power degradation. Figure 6.21 dictates the clock drifts estimates under the same circumstances.

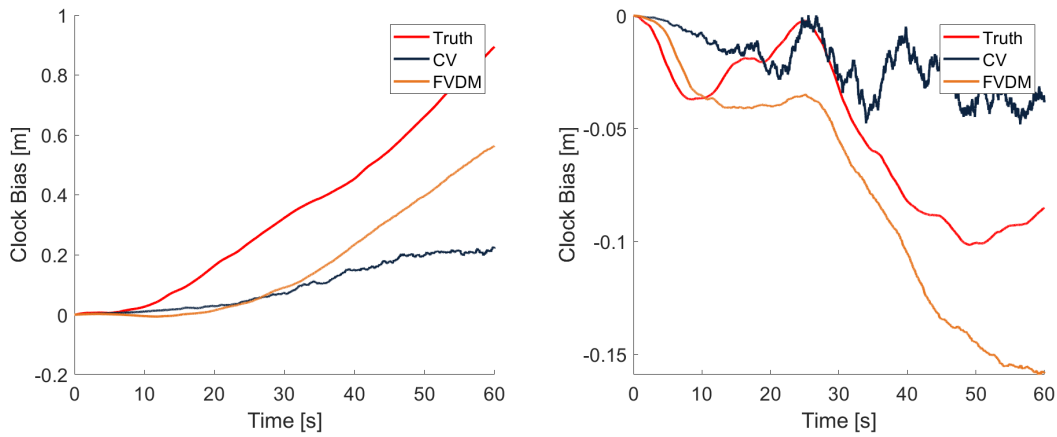


Figure 6.20: Average clock bias estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

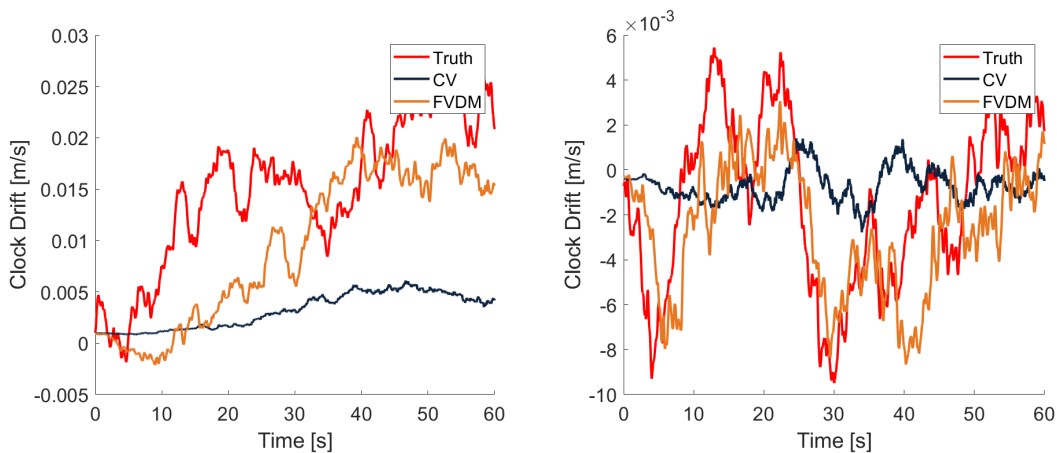


Figure 6.21: Average clock drift estimates of the FVDM and standard VDFLL implementation compared to the truth trajectory. The left figure is when both simulations had a signal power of 20 dB-Hz. The right figure is with a signal power of 25 dB-Hz.

Lastly, Tables 6.7 and 6.8 provide the RMSE, STD, and maximum error calculated across the 100-run Monte Carlo simulation when the receiver is subject to interference that leaves the signal power at 20 dB-Hz.

Table 6.7: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	0.64521	0.2673	0.1911	0.010044
STD	0.34121	0.12531	0.18278	0.0073514
Max Error	1.5828	0.76501	0.5855	0.23049

Table 6.8: RMSE, STD, and maximum error from 100-run Monte Carlo simulation when the receiver is subject to a degraded signal power level of 20 dB-Hz.

	Position [m]	Speed [m/s]	Clock Bias [m]	Clock Drift [m/s]
RMSE	196.98	6.0576	0.20098	0.010995
STD	126.68	0.92248	0.19358	0.0077631
Max Error	431.72	8.468	0.62891	0.024964

6.4 Conclusions

This chapter presented a detailed explanation of the two trajectories used for the performance analysis of the proposed navigation filter compared to a standard, constant-velocity kinematic model VDFLL. Trajectory one is the baseline case where the aircraft is commanded to fly straight and maintain a constant altitude for the entirety of the simulation. Trajectory two is a more dynamic trajectory with alternating, banking turns throughout the simulation on top of the aircraft being commanded to climb and maintain an altitude of 1150 meters. Each trajectory was subject to a range of interferences starting from benign signal power to low signal power. Results from trajectory one show that when flying a standard, unexcited trajectory, the standard VDFLL and the proposed navigation filter are comparable. Results from trajectory two showcase the improved performance from the deeply-coupled FVDM because of the inability in the standard EKF to predict angular rates and Euler attitude. However, the results from trajectory two also show that the FVDM is not the sole-solution due to the unobservable angular states when excitement is lost in the simulated trajectory. External sensors such as a IMU or multi-antenna GPS system could improve the system observability for better estimates. Improvements to the proposed navigation filter are discussed in greater details in the next chapter.

Chapter 7

Conclusions and Future Work

This work described the development of a deeply-integrated flight vehicle dynamic model within a GPS L1 C/A vector tracking software defined receiver. A performance analysis of the proposed navigation filter in GPS-challenged environments compared to the standard VDFLL was presented. The work presented in this thesis extends the state of the art by modeling a flight vehicle dynamic model deeply-coupled with GPS correlator measurements. The fidelity of the model stems from multiple flight mechanic modules presented in Chapter 2. The aerodynamic module models aerodynamic forces and moments on the basis of strip theory, using pre-processed CFD tables to propagate the aerodynamic coefficients based on the current flight condition of the aircraft. The engine and propeller module comprise of pre-processed numerical tables based on historical engine and propeller data sheets and performance charts. The FVDM improves the extended Kalman filter time update by fully acknowledging the behavior of the presented flight vehicle – this can be cumbersome with hardware sensors such as an IMU or barometer. The presented FVDM can be run at any update frequency, typically a limiting factor for hardware sensors. An increased update frequency for any vehicle dynamic model increases the likelihood that changes in the states are linear between time steps, this is especially important for high-dynamic systems such as hyper-velocity vehicles. Unlike hardware sensors, the FVDM is not subject to vibration, which is a common amongst rotor craft and fixed-wing flight vehicles. The downside to the FVDM is the requirement to have extensive knowledge of the flight vehicle in order to model it with high fidelity. This can be difficult if modeling government or proprietary vehicles.

For the straight, constant altitude trajectory, the proposed navigation filter was shown to have slight improvements over the standard VDFLL implementation in scenarios where the signal power dropped below 25 dB-Hz. With signal power above 25 dB-Hz, the standard EKF proved marginally better, likely due to an over confidence in the FVDM. When given a more dynamic trajectory, the deeply-integrated FVDM shows improvement by at least double in all scenarios of signal degradation. This is partly due to the FVDM being able to estimate the angular rate and Euler attitude of aircraft, where as the standard VDFLL implementation does not have this capability. For both trajectories, the proposed navigation filter improved tracking performance 4-6 dB-Hz over the standard VDFLL implementation. Overall, the deeply integrated FVDM is able to better predict the behavior of the aircraft given a set of control inputs, especially in scenario where GPS measurements are less reliable.

7.1 Concerns of Observability

From the presented navigation filter, measurements of position and velocity appear in the form of pseudorange and pseudorange-rates. These measurements are created based on DLL and FLL discriminator outputs that are converted to meters and meters per second, respectively. When a measurement update occurs within the EKF, these measurements indirectly correct the current state estimate through the observation matrix, \mathbf{H} (Equation 4.25). As stated previously, one reason for faulty estimates of the aircraft in heavily GPS-degraded environments is the lack of observability in the angular states. Once the estimated angular rates of the aircraft drift, and through integration, their respective Euler angles, the FVDM will continue to propagate the aircraft with these drifting estimates. For example, if the pitch angle estimate of the FVDM starts to drift from 10 degrees to 20 degrees to 30 degrees, the aircraft will naturally begin to pitch up and gain altitude. If the VT algorithm loses lock with signal channels, or the channel measurements are poor, there is no other measurements to correct the FVDM and it will continue to propagate worse estimates of the state. This can be further examined by exploring the observability matrix, \mathbf{O} through time.

$$\mathbf{O} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_k \Phi_k \\ \mathbf{H}_k \Phi_k^2 \\ \mathbf{H}_k \Phi_k^3 \\ \vdots \\ \mathbf{H}_k \Phi_k^{n-1} \end{bmatrix} \quad (7.1)$$

A system is fully-observable if the all of the states of the system can be known by the outputs of the systems, that is, if the rank of Equation 7.1 is equal to the number of states in \mathbf{X} (Equation 4.1), then the observability matrix is full rank and the system is fully-observable. A quick calculation of \mathbf{O} shows that the deeply-coupled FVDM is not fully-observable. Improvement upon the observability of the proposed navigation filter is expanded upon in the future work section of this chapter.

7.2 Future Work

For future work, the author recommends several items, both relating to the proposed navigation filter and to components of the FVDM presented in this work. With the complexity of the model, a common question is asked: *What level of fidelity is needed to acquire accurate estimates?* To answer this question, a new architecture of the FVDM is theorized (Figure 7.1), with its benefits listed herein. One of the downsides to the presented FVDM is the extremely catered modeling of one aircraft, the Diamond DA-40. The cascaded architecture would improve this such that aerodynamic coefficients are estimated inside of the flight mechanics module instead of being compiled beforehand using a CFD program. This would make the FVDM more generalizable, theoretically allowing it to be applicable to any aircraft, barring any configuration differences. The second improvement would be the fusion of other available sensors with the FVDM. For the presented work, it was assumed that the sensor suite became faulty or the size of the flight vehicle limited the capacity in which it could carry sensors. In reality, the form factor of modern low-cost sensors allow them to fit within the casing of a smart phone, and as such, should be fused with the FVDM for better estimates of forces and moments acting onto

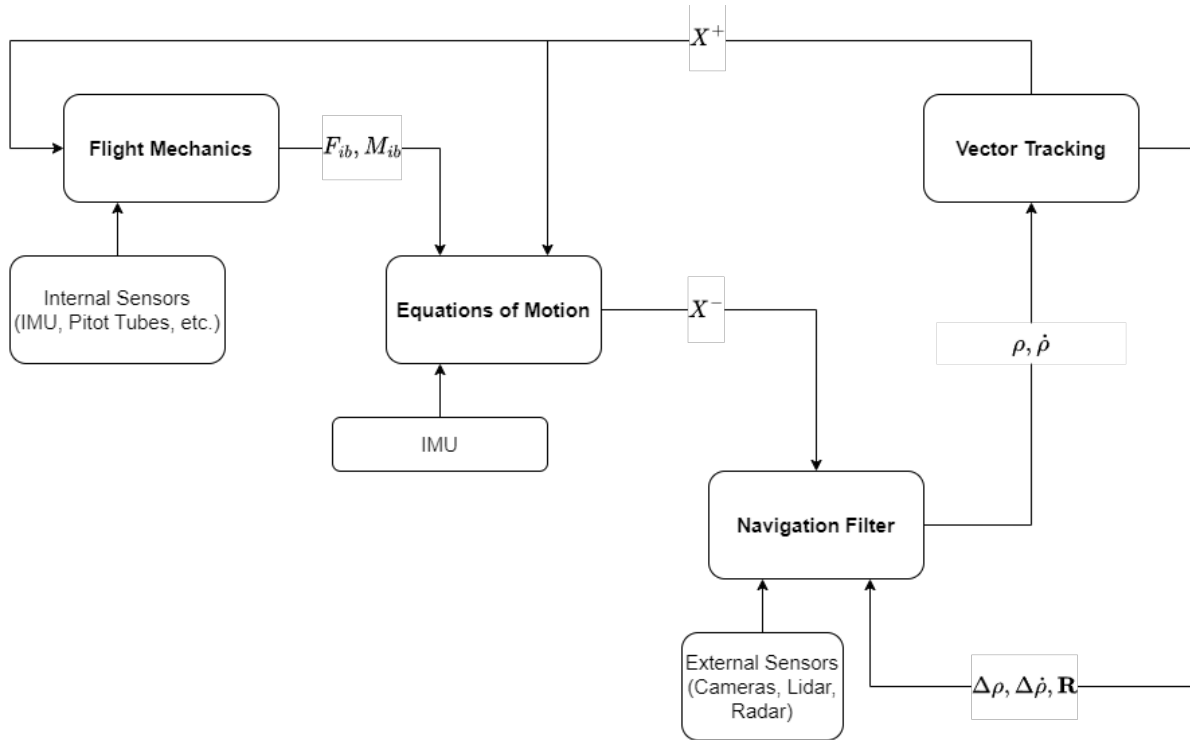


Figure 7.1: Cascaded architecture for an improved FVDM coupling.

the airframe. Typically, low-cost sensors are unable to update at high frequencies, discouraging their use onto high-dynamic systems. However, the FVDM is able to run at higher frequencies, so a fusion between the FVDM and low-cost sensors is intriguing and possible with the cascaded architecture. If fusing multiple sensors with the FVDM using the cascaded architecture, it would be possible to measure the effectiveness of each sensor. Through decades of prior research, multi-sensor fusion algorithms have shown to provide robust pose estimates of the collection platform for a variety of dynamics and signal interference. The cascaded architecture proposed in this section would allow further fusion of sensors not typically used in navigation algorithms (i.e. pitot tubes, engine temperature sensor) via the modeling equations from the FVDM.

Bibliography

- [1] Xiaojie Zhao, Gang Wang, and Shuzhi Sam Ge. “Cooperative Localization Based on Robust GPS and Radar Fusion for Multiple Aerial Vehicles”. In: *International Journal of Control, Automation and Systems* 15.2 (Apr. 2017), pp. 732–742. ISSN: 1598-6446, 2005-4092. DOI: 10.1007/s12555-014-0246-6. URL: <http://link.springer.com/10.1007/s12555-014-0246-6> (visited on 05/16/2023).
- [2] Justin R. Rufa and Ella M. Atkins. “Sensor Fusion for Unmanned Aircraft System Navigation in an Urban Environment”. In: *2014 American Control Conference*. 2014 American Control Conference - ACC 2014. Portland, OR, USA: IEEE, June 2014, pp. 5313–5318. ISBN: 978-1-4799-3274-0 978-1-4799-3272-6 978-1-4799-3271-9. DOI: 10.1109/ACC.2014.6858975. URL: <http://ieeexplore.ieee.org/document/6858975/> (visited on 04/23/2023).
- [3] Robert Tenny and Todd E. Humphreys. “Robust Navigation for Urban Air Mobility via Tight Coupling of GNSS with Terrestrial Radionavigation and Inertial Sensing”. In: *35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*. Denver, Colorado, Oct. 20, 2022, pp. 1599–1609. DOI: 10.33012/2022.18469. URL: <https://www.ion.org/publications/abstract.cfm?articleID=18469> (visited on 04/16/2023).
- [4] Gorkem Kandemir and Haluk Erdem Bingol. “A Probabilistic Measurement Method for Terrain Aided Navigation for Airborne Platforms”. In: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). Monterey, CA: IEEE, Apr. 2018, pp. 1245–1249.

- ISBN: 978-1-5386-1647-5. DOI: 10.1109/PLANS.2018.8373512. URL: <https://ieeexplore.ieee.org/document/8373512/> (visited on 04/17/2023).
- [5] *Aircraft Characteristics Database* | *Federal Aviation Administration*. URL: www.faa.gov/airports/engineering/aircraft_char_database (visited on 07/20/2023).
- [6] *Diamond Aircraft DA40 Series – Airborne Innovation*. Dec. 31, 1969. URL: <https://www.diamondaircraft.com/en/private-owners/aircraft/da40/overview/> (visited on 07/20/2023).
- [7] Customer Services. *A380 Aircraft Characteristics Airport and Maintenance Planning*. Airbus S.A.S. Customer Service, Nov. 2020.
- [8] Tanner Watts. “A GPS and GLONASS L1 Vector Tracking Software-Defined Receiver”. Auburn University, Feb. 19, 2019. 276 pp. URL: <https://www.ion.org/publications/abstract.cfm?articleID=16686> (visited on 06/19/2023).
- [9] David Alan Grier et al. *Position, Navigation, and Timing Technologies in the 21st Century*. Vol. 1. 2 vols. Piscataway, New Jersey, USA: IEEE Press.
- [10] Airbus Customer Services. *Airbus A380: Aircraft Characteristics Airport and Maintenance Planning*. Airbus, Nov. 1, 2020.
- [11] Daniel Salmon. “An Experimental Exploration of Low-Cost Sensor and Vehicle Model Solutions for Precision Ground Vehicle Navigation”. Auburn University, Dec. 12, 2015. 129 pp.
- [12] Matthew Rhudy. “A Dynamic Model-Aided Sensor Fusion Approach to Aircraft Attitude Estimation”. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). Boston, MA, USA: IEEE, Aug. 2017, pp. 1406–1409. ISBN: 978-1-5090-6389-5. DOI: 10.1109/MWSCAS.2017.8053195. URL: <http://ieeexplore.ieee.org/document/8053195/> (visited on 08/12/2023).

- [13] Robert C. Nelson. *Flight Stability and Automatic Control*. 2nd ed. Boston, Mass: WCB/McGraw Hill, 1998. 441 pp. ISBN: 978-0-07-046273-1.
- [14] Mehran Khaghani and Jan Skaloud. “Autonomous Vehicle Dynamic Model-Based Navigation for Small UAVs: Autonomous VDM Based Navigation for Small UAVs”. In: *Navigation* 63.3 (Sept. 2016), pp. 345–358. ISSN: 00281522. DOI: 10.1002/navi.140. URL: <https://onlinelibrary.wiley.com/doi/10.1002/navi.140> (visited on 08/13/2023).
- [15] Mehran Khaghani and Jan Skaloud. “Assessment of VDM-based Autonomous Navigation of a UAV under Operational Conditions”. In: *Robotics and Autonomous Systems* 106 (Aug. 2018), pp. 152–164. ISSN: 09218890. DOI: 10.1016/j.robot.2018.05.007. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0921889017303792> (visited on 08/12/2023).
- [16] Hery A. Mwenegoha et al. “A Model-based Tightly Coupled Architecture for Low-Cost Unmanned Aerial Vehicles for Real-Time Applications”. In: *IEEE Access* (2020), pp. 1–1. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3038530. URL: <https://ieeexplore.ieee.org/document/9261363/> (visited on 09/27/2023).
- [17] Xin Dong et al. “An Integrated UWB-IMU-Vision Framework for Autonomous Approaching and Landing of UAVs”. In: *Aerospace* 9.12 (Dec. 5, 2022), p. 797. ISSN: 2226-4310. DOI: 10.3390/aerospace9120797. URL: <https://www.mdpi.com/2226-4310/9/12/797> (visited on 04/23/2023).
- [18] Tamás Gróf, Péter Bauer, and Yoko Watanabe. “Positioning of Aircraft Relative to Unknown Runway with Delayed Image Data, Airdata and Inertial Measurement Fusion”. In: *Control Engineering Practice* 125 (Aug. 2022), p. 105211. ISSN: 09670661. DOI: 10.1016/j.conengprac.2022.105211. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0967066122000922> (visited on 04/23/2023).
- [19] Bradford Parkinson and James Spilker Jr. *Global Positioning System: Theory and Application*. Ed. by Paul Zarchan. Vol. 1. California: American Institute of Aeronautics and Astronautics, 1996. 1056 pp.

- [20] Donald Benson. “Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver”. In: ().
- [21] Matthew Lashley, David M. Bevly, and John Y. Hung. “Performance Analysis of Vector Tracking Algorithms for Weak GPS Signals in High Dynamics”. In: *IEEE Journal of Selected Topics in Signal Processing* 3.4 (Aug. 2009), pp. 661–673. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2009.2023341. URL: <http://ieeexplore.ieee.org/document/5166626/> (visited on 09/07/2023).
- [22] W. Luke Edwards, Benjamin J. Clark, and David M. Bevly. “Implementation Details of a Deeply Integrated GPS/INS Software Receiver”. In: *IEEE/ION Position, Location and Navigation Symposium. 2010 IEEE/ION Position, Location and Navigation Symposium - PLANS 2010*. Indian Wells, CA, USA: IEEE, May 2010, pp. 1137–1146. ISBN: 978-1-4244-5036-7. DOI: 10.1109/PLANS.2010.5507232. URL: <http://ieeexplore.ieee.org/document/5507232/> (visited on 08/20/2023).
- [23] Matthew Lashely. “Modeling and Performance Analysis of GPS Vector Tracking Algorithms”. PhD thesis. Auburn, Alabama: Auburn University, Dec. 18, 2009. 417 pp.
- [24] Matthew M Peet. “Spacecraft and Aircraft Dynamics: 6 Degrees of Freedom Equations of Motion”. Lecture Presentation (Illinois Institute of Technology). Aut. 2021.
- [25] Nando van Arnhem et al. “Engineering Method to Estimate the Blade Loading of Propellers in Nonuniform Flow”. In: *AIAA Journal* 58.12 (2020), pp. 5332–5346. ISSN: 0001-1452. DOI: 10.2514/1.J059485. URL: <https://doi.org/10.2514/1.J059485> (visited on 07/31/2023).
- [26] *Force Balance Coordinates*. Dec. 31, 1969. URL: <https://www.grc.nasa.gov/www/k-12/airplane/tunbalaxes.html> (visited on 07/31/2023).
- [27] *US Standard Atmosphere, 1976*. National Oceanic and Atmospheric Administration, Oct. 1976.
- [28] *FlightStream*. Nov. 25, 2022. URL: <https://researchinflight.com/> (visited on 07/25/2023).

- [29] *What Is Lift?* Glenn Research Center | NASA. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/what-is-lift/> (visited on 11/02/2023).
- [30] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series. Reston, VA: American Institute of Aeronautics and Astronautics, Inc, 2018. ISBN: 978-1-62410-490-9.
- [31] Jacob Cook. “A Strip Theory Approach to Dynamic Modeling of eVTOL Aircraft”. In: *AIAA Scitech 2021 Forum*. AIAA Scitech 2021 Forum. VIRTUAL EVENT: American Institute of Aeronautics and Astronautics, Jan. 11, 2021. ISBN: 978-1-62410-609-5. DOI: 10.2514/6.2021-1720. URL: <https://arc.aiaa.org/doi/10.2514/6.2021-1720> (visited on 08/08/2023).
- [32] *Aerodynamic Strip Theory - J2 Aircraft Dynamics*. Oct. 11, 2021. URL: <https://j2aircraft.com/software/universal-toolkit/aircraft-modeling-plugins/j2-elements/aerodynamic-strip-theory/> (visited on 08/10/2023).
- [33] Vivek Ahuja et al. “Integrated Computational Aeroacoustics for UAM Design. Part 1. Farassat F1A Formulation”. In: *AIAA AVIATION 2022 Forum*. AIAA AVIATION 2022 Forum. Chicago, IL & Virtual: American Institute of Aeronautics and Astronautics, June 27, 2022. ISBN: 978-1-62410-635-4. DOI: 10.2514/6.2022-3954. URL: <https://arc.aiaa.org/doi/10.2514/6.2022-3954> (visited on 08/10/2023).
- [34] Rajan Bhandari, Aashutosh A. Mishra, and Imon Chakraborty. “Genetic Algorithm Optimization of Lift-Plus-Cruise VTOL Aircraft with Electrified Propulsion”. In: *AIAA SCITECH 2023 Forum*. AIAA SCITECH 2023 Forum. National Harbor, MD & Online: American Institute of Aeronautics and Astronautics, Jan. 23, 2023. ISBN: 978-1-62410-699-6. DOI: 10.2514/6.2023-0398. URL: <https://arc.aiaa.org/doi/10.2514/6.2023-0398> (visited on 08/10/2023).

- [35] Snorri Gudmundsson. *General Aviation Aircraft Design: Applied Methods and Procedures*. First edition. Oxford ; Waltham, MA: Butterworth-Heinemann, 2014. 1034 pp. ISBN: 978-0-12-397308-5.
- [36] Hartzell Propeller. *Hartzell Propeller Manual No. 159 Rev. 84*. Hartzell Propeller Incorporated, May 2023.
- [37] Ian. *How to Identify Historic Aircraft Propellers*. Hartzell Propeller. Dec. 13, 2016. URL: <https://hartzellprop.com/identify-historic-aircraft-propellers/> (visited on 07/25/2023).
- [38] Ian. *A Short History of the Aircraft Propeller*. Hartzell Propeller. Dec. 19, 2018. URL: <https://hartzellprop.com/a-short-history-of-the-aircraft-propeller/> (visited on 07/25/2023).
- [39] *Aerodynamics: Propeller Basics - Learn To Fly*. July 5, 2016. URL: <https://learntoflyblog.com/aerodynamics-propeller-basics/> (visited on 01/03/2024).
- [40] *Hartzell Propeller Inc. | Aircraft and Airplane Propeller Systems*. Hartzell Propeller. Dec. 31, 1969. URL: <https://hartzellprop.com/> (visited on 07/25/2023).
- [41] *Generalized Method of Propeller Performance Estimation*. Hamilton Standard, June 28, 1963.
- [42] Ying Xing. "The Strength Analysis of the Diagonal Stay of Aircraft Landing Gear Based on ANSYS". In: *Proceedings of the 2nd International Conference on Electronic and Mechanical Engineering and Information Technology (2012)*. 2nd International Conference on Electronic and Mechanical Engineering and Information Technology. China: Atlantis Press, 2012. ISBN: 978-90-78677-60-4. DOI: 10.2991/emeit.2012.176. URL: <http://www.atlantis-press.com/php/paper-details.php?id=3394> (visited on 06/03/2023).
- [43] Noah S Miller et al. "SNAP: A Xona Space Systems and GPS Software-Defined Receiver". In: *Institute of Navigation* (Apr. 1, 2023), p. 8.

- [44] Elliott D. Kaplan and C. Hegarty, eds. *Understanding GPS: Principles and Applications*. 2nd ed. Artech House Mobile Communications Series. Boston: Artech House, 2006. 703 pp. ISBN: 978-1-58053-894-7.
- [45] Dennis M. Akos and Marco Pini. “Effect of Sampling Frequency on GNSS Receiver Performance”. In: *Navigation* 53.2 (June 2006), pp. 85–95. ISSN: 00281522. DOI: 10.1002/j.2161-4296.2006.tb00375.x. URL: <https://onlinelibrary.wiley.com/doi/10.1002/j.2161-4296.2006.tb00375.x> (visited on 10/17/2022).
- [46] Logan Scott, Aleksandar Jovancevic, and Suman Ganguly. “Rapid Signal Acquisition Techniques for Civilian & Military User Equipments Using DSP Based FFT Processing”. In: ION GPS. Salt Lake City, Utah, USA: Institute of Navigation, Sept. 2001, p. 10.
- [47] Dennis Akos et al. “GNSS Software Defined Radio: History, Current Developments, and Standardization Efforts”. In: 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022). Denver, Colorado, Oct. 20, 2022, pp. 3180–3209. DOI: 10.33012/2022.18434. URL: <https://www.ion.org/publications/abstract.cfm?articleID=18434> (visited on 04/16/2023).
- [48] David Vallado and Paul Crawford. “SGP4 Orbit Determination”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. AIAA/AAS Astrodynamics Specialist Conference and Exhibit. Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Aug. 18, 2008. ISBN: 978-1-62410-001-7. DOI: 10.2514/6.2008-6770. URL: <https://arc.aiaa.org/doi/10.2514/6.2008-6770> (visited on 11/02/2023).
- [49] J Sanz Subirana, J M Juan Zornoza, and M Hernández-Pajares. → *GNSS DATA PROCESSING*.
- [50] Tiantian Huang et al. “An Integrated Adaptive Kalman Filter for High-Speed UAVs”. In: *Applied Sciences* 9.9 (May 9, 2019), p. 1916. ISSN: 2076-3417. DOI: 10.3390/app9091916. URL: <https://www.mdpi.com/2076-3417/9/9/1916> (visited on 10/14/2023).

- [51] Paul Groves. *Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems*. 2nd ed. Vol. 39. Artech House, Apr. 27, 2012. 776 pp.
- [52] Christopher Robert Hamm. “Analysis of Simulated Performance of Integrated Vector Tracking and Navigation Loops for GPS”. Thesis. Auburn, Alabama: Auburn University, Dec. 16, 2005. 97 pp.
- [53] Oscar Pozzobon et al. “Supersonic GNSS Authentication Codes”. In: *ION International Technical Meeting*. ION GNSS+. Vol. 27. Satellites. Tampa, Florida, USA: Institute of Navigation, Sept. 2014, p. 8.
- [54] Scott Martin. “GPS Carrier Phase Tracking in Difficult Environments Using Vector Tracking For Precise Positioning and Vehicle Attitude Estimation”. Auburn University, May 6, 2017. 259 pp.
- [55] E. M. Copps et al. “Optimal Processing of GPS Signals”. In: *ION*. Institute of Navigation 27.3 (Aut. 1980), pp. 171–182. DOI: 10.1002/j.2161-4296.1980.tb01417.x.
- [56] Nesreen I Ziedan. “Multipath Channel Estimation and Pattern Recognition for Environment-Based Adaptive Tracking”. In: (2012).
- [57] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering With MATLAB Exercises*. 4th ed. John Wiley & Sons, Ltd, Oct. 2013. 397 pp. ISBN: 978-0-470-60969-9.
- [58] *Google Earth*. Dec. 31, 1969. URL: <https://earth.google.com/web/@0,-0.227,0a,22251752.77375655d,35y,0h,0t,0r> (visited on 10/16/2023).
- [59] *KML Toolbox - File Exchange - MATLAB Central*. Dec. 31, 1969. URL: <https://www.mathworks.com/matlabcentral/fileexchange/34694-kml-toolbox> (visited on 10/16/2023).
- [60] Jacek Pieniżek et al. “Dynamic Response of the Pitot Tube with Pressure Sensor”. In: *Sensors* 23.5 (Mar. 6, 2023), p. 2843. ISSN: 1424-8220. DOI: 10.3390/s23052843. URL: <https://www.mdpi.com/1424-8220/23/5/2843> (visited on 05/25/2023).

- [61] Daniel A Marietta, Kenneth A Fisher, and Clark N Taylor. “Monte Carlo Error Characterization of EKF-Based Image Aided Navigation”. In: *San Diego* (2013).
- [62] C Noll. “The Crustal Dynamics Data Information System: A Resource to Support Scientific Analysis Using Space Geodesy”. In: *Advances in Space Research* 45.12 (June 15, 2010), pp. 1421–1440. ISSN: 0273-1177. DOI: 10.1016/j.asr.2010.01.018. URL: <https://cddis.nasa.gov>.
- [63] Cheng-Wei Wang and Shau-Shiun Jan. “Kalman Filter-Based Integrity Monitoring for MADOCA-PPP in Terrestrial Applications”. In: (). DOI: 10.1109/PLANS53410.2023.10140022.

Appendix A Extra Results from Trajectory Two

This appendix shows comparison of the proposed navigation filter to the standard kinematic model VDFLL when neither are receiving any measurements from GPS for the dynamic flight path of trajectory two. Without any GPS measurements to assist the kinematic model, it fails to predict the motion of the aircraft. However, because of the capability for the FVDM to predict the behavior of the aircraft given a set of control inputs, the state estimates of the aircraft are much improved.

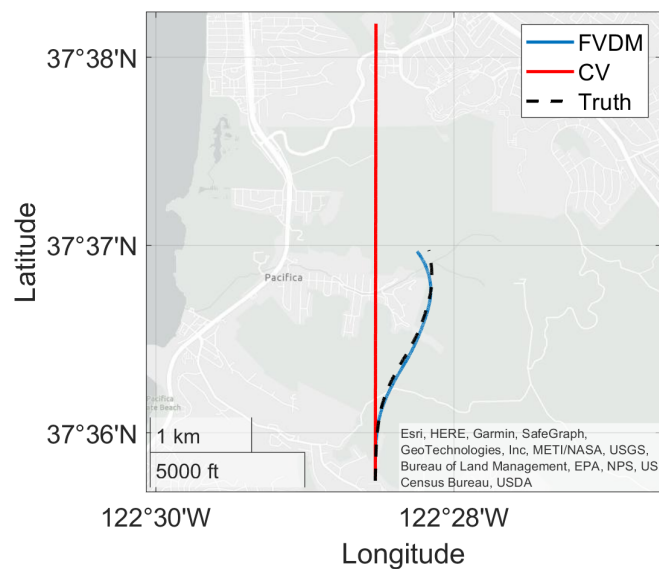


Figure 2: Latitude and longitude of the proposed navigation filter compared to the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.

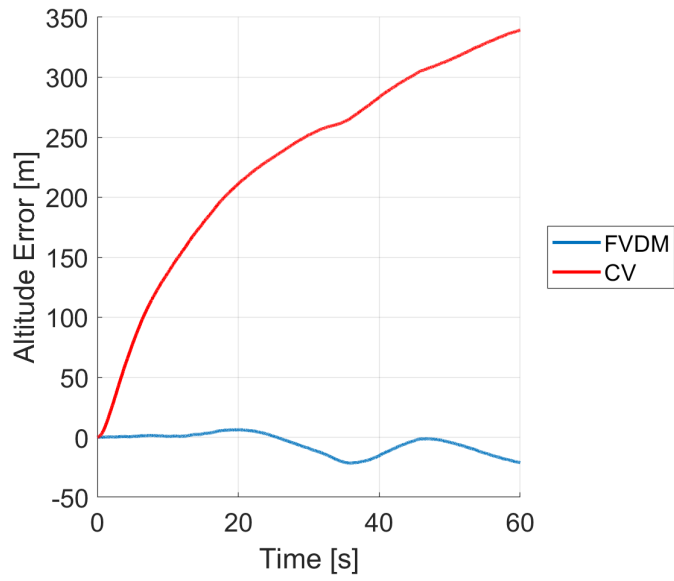


Figure 3: Altitude errors of the proposed navigation filter and the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.

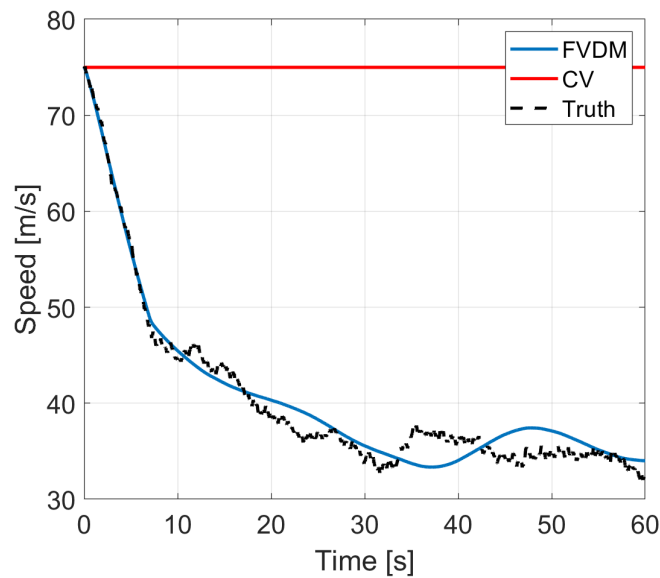


Figure 4: Aircraft speed estimate from the proposed navigation filter compared to the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.

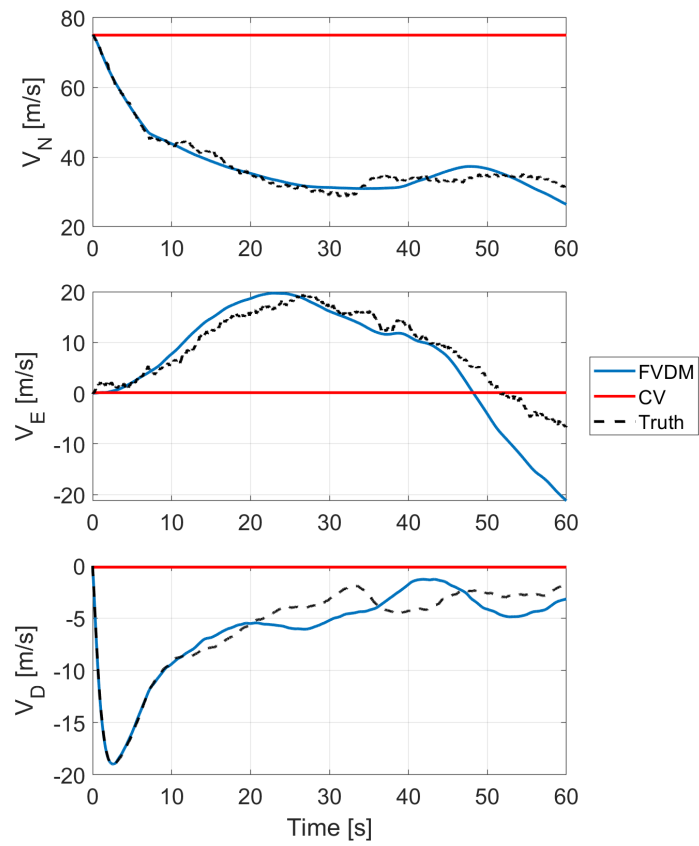


Figure 5: Velocity components of the proposed navigation filter compared with the standard constant velocity EKF VDFLL when receiving no GPS measurements for trajectory two.

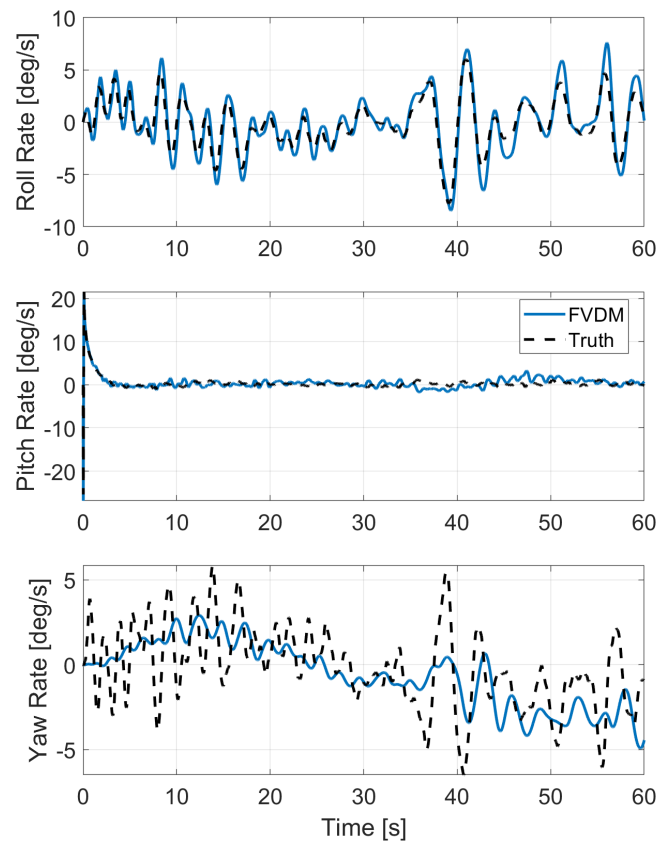


Figure 6: Angular rate estimates of the proposed navigation filter when receiving no GPS measurements for trajectory two.

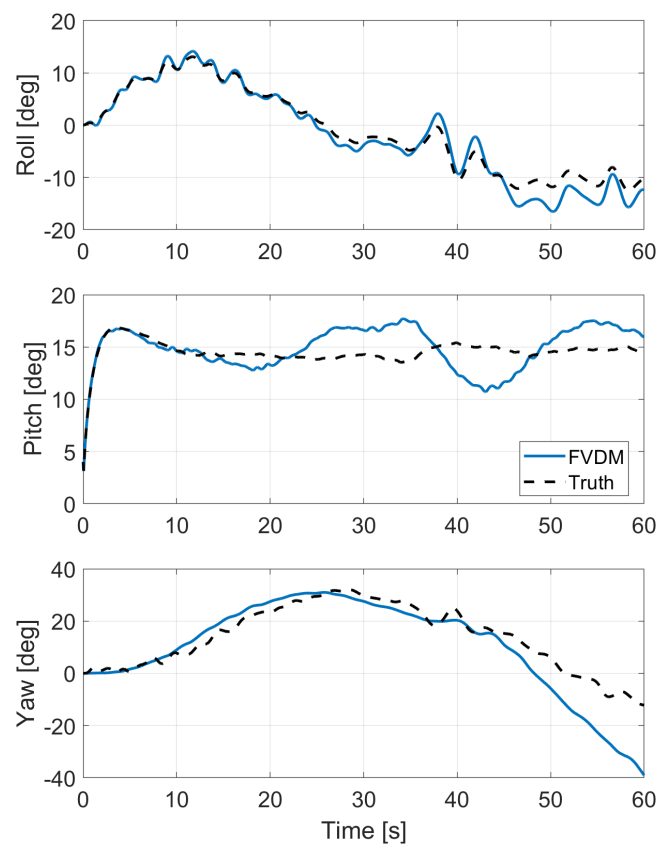


Figure 7: Euler attitude estimates of the proposed navigation filter when receiving no GPS measurements for trajectory two.