

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

**ADVANCED AIDING METHODS
IN INERTIAL NAVIGATION SYSTEMS
FOR GNSS DENIED ENVIRONMENTS**

AYBARS TOKTA

**A THESIS OF DOCTORATE
DEPARTMENT OF ELECTRONICS ENGINEERING**

ADVISOR: ASSIST. PROF. DR. ALİ KÖKSAL HOCAOĞLU

FEBRUARY 2024

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL

ADVANCED AIDING METHODS
IN INERTIAL NAVIGATION SYSTEMS
FOR GNSS DENIED ENVIRONMENTS

AYBARS TOKTA

A THESIS OF DOCTORATE
DEPARTMENT OF ELECTRONICS ENGINEERING

ADVISOR: ASSIST. PROF. DR. ALİ KÖKSAL HOCAOĞLU

FEBRUARY 2024

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KKS ENGELLENEN ORTAMLAR İÇİN
ATALETSEL NAVİGASYON SİSTEMLERİNDE
İLERİ DESTEK METOTLARI

AYBARS TOKTA

DOKTORA TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: DR. ÖĞR. ÜYESİ ALİ KÖKSAL HOCAOĞLU

ŞUBAT 2024



DOCTORATE JURY APPROVAL FORM

A thesis submitted by Aybars TOKTA, defended on 21/02/2024 before the jury formed with the 05/02/2024 date and 2024/10 numbered decision of the GTU Graduate Administration Board, has been accepted as a DOCTOR of PHILOSOPHY thesis in the Department of Electronics Engineering.

JURY

MEMBER

(THESIS ADVISOR) : Dr. Öğr. Üyesi Ali Köksal HOCAOĞLU

MEMBER

: Prof. Dr. Oğuz KUCUR

MEMBER

: Prof. Dr. Tansu FİLİK

MEMBER

: Prof. Dr. Ayşın Baytan ERTÜZÜN

MEMBER

: Dr. Öğr. Üyesi Saliha BÜYÜKÇORAK EDİBALI

APPROVAL

Gebze Technical University Graduate Administration Board
...../...../..... date and/..... numbered decision.

SIGNATURE/SEAL



*For Fahri Tokta,
my beloved grandfather.
You are missed every day.*

ABSTRACT

Inertial navigation systems have a long history of providing kinematic information for various applications, ranging from aviation and maritime navigation to guidance systems for space exploration and military operations. An inertial navigation system includes two essential sensor triads: accelerometers and gyroscopes. Their outputs are integrated to deliver high-frequency, real-time data on attitude, velocity, and position. However, inherent errors in these sensors, such as bias, drift, and noise, lead to degradation in the navigation solution over time. This problem is effectively mitigated by the utilization of global navigation satellite systems, which provide global absolute positioning data with a comparably lower error range that remains bounded over time. Consequently, combining the solutions of these systems has become standard practice on nearly all platforms. However, signals transmitted from satellites are vulnerable to interference such as jamming and spoofing, and can also be obstructed or degraded. This susceptibility has spurred research into leveraging alternative systems to reduce the errors in inertial navigation systems. This thesis primarily focuses on terrain-aided and vision-aided navigation techniques, as well as synthetic inertial measurement unit data generation for testing and validating integrated navigation solutions in a simulation environment. A novel multi-hypothesis filter, designed specifically for terrain-aided navigation, is introduced, offering robustness against diverse terrain shapes and significant initialization errors. In addition, a method for generating synthetic inertial measurement unit data is also proposed that incorporates various sensor errors and vibrational effects. Finally, the direct linear transform-based absolute visual positioning method is tested with various visual feature extraction methods to evaluate their positioning accuracy using a real orthophoto dataset.

Keywords: Inertial Navigation, Terrain and Vision Aided Navigation, Trajectory Shaping, Synthetic IMU Data Generation.

ÖZET

Ataletsel navigasyon sistemleri, havacılık ve denizcilik navigasyonundan uzay keşfi ve askeri operasyonlara kadar çeşitli uygulamalar için kinematik bilgi sağlama konusunda uzun bir geçmişe sahiptir. Bir ataletsel navigasyon sistemi, iki temel sensör üçlüsünü içerir: ivmeölçerler ve dönüölçerler. Bunların çıktıların integralleri alınarak, yüksek frekansta, gerçek zamanlı yönelim, hız ve pozisyon bilgisi elde edilebilir. Ancak, bu sensörlerin içsel hataları, örneğin yanlışlık, sürüklenme ve gürültü, zamanla navigasyon çözümünün bozulmasına neden olur. Bu sorun, küresel mutlak konumlandırma verisi sağlayan ve hata aralığı zamanla sınırlı kalan, daha düşük bir hata aralığına sahip global navigasyon uydu sistemlerinin kullanımı ile etkili bir şekilde giderilebilir. Sonuç olarak, bu sistemlerin çözümlerinin birleştirilmesi neredeyse tüm platformlarda standart uygulama haline gelmiştir. Ancak, uydu sinyalleri, karıştırma ve aldatma gibi müdahalelere, ayrıca engellenmeye karşı savunmasızdır. Bu savunmasızlık, ataletsel navigasyon sistemlerindeki hataları azaltmak için alternatif sistemleri kullanma araştırmalarını teşvik etmiştir. Bu tez, temel olarak arazi yardımcı ve görüntü yardımcı navigasyon tekniklerine ve bütünsel navigasyon çözümlerini bir simülasyon ortamında test etmek ve doğrulamak için sentetik ataletsel ölçüm birimi verisi üretimine odaklanır. Arazi yardımcı navigasyon için özel olarak tasarlanmış yeni bir çok hipotezli filtre tanıtılmakta, bu filtre çeşitli arazi şekillerine ve önemli başlangıç hatalarına karşı dayanıklılık sunmaktadır. Ayrıca, çeşitli sensör hatalarını ve titreşim etkilerini içerecek şekilde sentetik ataletsel ölçüm birimi verisi üretimi için bir yöntem de önerilmektedir. Son olarak, doğrudan doğrusal dönüşüm tabanlı mutlak görsel konumlandırma yöntemi, çeşitli görsel özellik çıkarma yöntemleri ile birlikte gerçek bir ortofoto veri seti kullanılarak pozisyonlama doğruluğunu değerlendirmek için test edilmiştir.

Anahtar Kelimeler: Atalatsel Navigasyon, Arazi ve Görüntü Destekli Navigasyon, İz şekillendirme, Sentetik AÖB Veri Üretimi.

ACKNOWLEDGEMENTS

First and foremost, I extend my deepest gratitude to my mother and father, Nilüfer and Yücel, as well as to my wife, Nilüfer, for their unwavering support and unconditional love. Their presence and encouragement have been the foundation upon which I built my aspirations. Furthermore, I would like to express my sincere thanks to my thesis advisor, Assist. Prof. Köksal Hocaoglu, whose guidance has been invaluable throughout my academic journey, from my undergraduate degree to my doctoral studies. Finally, special thanks to Hashüs, who not only followed me to Aselsan for the shared laughter but also provided steadfast support, proving that true friendship endures through every chapter of our lives.



TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	vi
ÖZET	vii
ACKNOWLEDGEMENT	viii
TABLE of CONTENTS	xi
LIST of ABBREVIATIONS and ACRONYMS	xii
LIST of FIGURES	xv
LIST of TABLES	xvi
1. INTRODUCTION	1
1.1. Motivation and Scope	2
1.2. Contributions	2
1.3. Thesis Outline	3
2. BASICS OF INERTIAL NAVIGATION	4
2.1. Reference Frames	4
2.1.1. Earth-Centered Earth-Fixed Frame (ECEF)	4
2.1.2. Inertial Frame(I)	4
2.1.3. Local Navigation Frame	4
2.1.4. Wander Azimuth Frame	5
2.1.5. Body Frame	5
2.2. Basics of Coordinate Frame Mathematics	5
2.2.1. Kinematic Notation	6
2.2.2. Euler Representation	6
2.2.3. Direction Cosine Matrix (DCM) Representation	8
2.2.4. Basics of Motion Involving Multiple Frames	8
2.3. Inertial Navigation Mechanization Equations	12
2.3.1. Velocity Mechanization In Inertial Frame	12
2.3.2. Velocity Mechanization In Navigation Frame	13
2.4. Attitude Mechanization Equation	14
2.5. Earth Curvatures, Gravity and Position Calculation	14
2.6. Discrete Implementation of Inertial Navigation Equations	15
2.6.1. Attitude Calculation	16
2.6.2. Velocity Calculation	17
2.6.3. Position Calculation	17
2.7. Derivation of Error Dynamics	18
2.7.1. Position Error Dynamics	18
2.7.2. Velocity Error Dynamics	19
2.7.3. Attitude Error Dynamics	21

2.8.	State-space Demonstration of Error Dynamics Equations	22
2.9.	Augmenting the Error State	23
2.10.	Simplified Error Analysis	24
2.10.1.	Simplified Lateral Error Dynamics	24
2.10.2.	Simplified Vertical Error Dynamics	27
3.	TRAJECTORY AND IMU DATA GENERATION	28
3.1.	Trajectory Generation	28
3.1.1.	Polynomial Fitting for a Dynamic Segment	30
3.2.	IMU Data Generation via Inverse Kinematics	31
3.3.	Deterministic and Stochastic IMU Errors	34
3.3.1.	Bias Error	34
3.3.1.1.	Bias Repeatability Implementation	35
3.3.1.2.	Bias Instability Implementation	35
3.3.2.	Scale Factor Error	36
3.3.3.	Misalignment Error	37
3.3.4.	Angle & Velocity Random Walk Errors	38
3.3.5.	Other Error Types	39
3.4.	Data generation for Master-Slave Platforms	39
3.5.	Data generation for Gimballed Platforms	40
3.5.1.	Tracking a Point with a FLIR	42
3.6.	Vibration	46
3.6.1.	Power Spectral Density and Fourier Transform Relationship	46
3.6.2.	Synthetic Vibration Generation	50
4.	GNSS AIDED NAVIGATION	52
4.1.	Basics of GNSS	52
4.2.	Position Estimation Using Pseudorange	53
4.3.	Kalman Filter	56
4.4.	Loosely Coupled INS-GNSS Fusion	58
4.5.	Simulation Results	62
4.5.1.	Scenario-1	62
4.5.2.	Scenario-2	65
5.	TERRAIN AIDED NAVIGATION	69
5.1.	Literature Overview	69
5.1.1.	TERCOM	71
5.1.2.	SITAN	73
5.2.	A Novel Multi-Hypothesis Filter	74
5.2.1.	The Notation	75
5.2.2.	Multi-Hypothesis Structure In TAN	78
5.2.2.1.	The Definition of Hypothesis & multi-hypothesis	79
5.2.2.2.	Hypothesis Set Propagation	80
5.2.2.3.	Measurement Generation	81
5.2.2.4.	Gridding	83
5.2.3.	Clustering	84
5.2.3.1.	Hypothesis Set Correction	86
5.2.4.	Hypothesis Pruning and Merging	88
5.2.4.1.	Ad-Hoc Measures to Prevent Divergence	90

5.2.5.	Hybrid Output	93
5.2.6.	Simulation Results	94
5.2.7.	Discussion	96
6.	VISUAL AIDED NAVIGATION	99
6.1.	Relative Visual Positioning	99
6.2.	Absolute Visual Positioning	101
6.3.	Pinhole Camera Model & Projection Matrix	104
6.4.	Estimation of Camera Projection Matrix	106
6.4.1.	Decomposition of P	108
6.5.	Simulations	108
6.5.1.	Synthetic Camera Image Generation	108
6.6.	Discussion	111
7.	CONCLUSIONS AND FUTURE WORK	115
7.1.	Conclusions	115
7.2.	Future Works	115
7.2.1.	Terrain Aided Navigation	116
7.2.2.	Vision Aided Navigation	116
	REFERENCES	118
	BIOGRAPHY	125
	PUBLICATIONS WITHIN the SCOPE of THESIS STUDY	126

LIST of ABBREVIATIONS and ACRONYMS

2D	: 2-Dimensional
3D	: 3-Dimensional
AKAZE	: Accelerated KAZE
AI	: Artificial Intelligence
ARW	: Angle Random Walk
AVP	: Absolute Visual Positioning
CENSURE	: Center Surround Extremas
CS	: Cauchy-Schwarz
CSD	: Cauchy-Schwarz Divergence
DCM	: Direction Cosine Matrix
DEM	: Digital Elevation Model
DFT	: Discrete Fourier Transform
DLT	: Direct Linear Transform
DoF	: Degrees of Freedom
DTED	: Digital Terrain Elevation Data
ECEF	: Earth Centered Earth Fixed
EKF	: Extended Kalman Filter
ENU	: East-North-Up Frame
FAST	: Features from Accelerated Segment Test
FLIR	: Forward Looking Infrared
FT	: Fourier Transform
GAN	: Generative Adversarial Network
GM	: Gaussian Mixture
GNSS	: Global Navigation Satellite System
GPS	: Global Positioning System
ICP	: Iterative Closest Point
IWEKF	: Interacting Weighted Ensemble Kalman Filter
ITRF	: International Terrestrial Reference Frame
IMU	: Inertial Measurement Unit
INS	: Inertial Navigation System
KF	: Kalman Filter
KLD	: Kullback-Leibler Divergence
LOS	: Line of Sight
MAP	: Maximum a Posteriori
MAD	: Mean Absolute Difference
MEMS	: Micro Electro Mechanical Systems
MHF	: Multiple Hypothesis Filter
MSD	: Mean Squared Difference
MSL	: Mean Sea Level
NEES	: Normalized Estimation Error Squared
NED	: North-East-Down Frame
PF	: Particle Filter
PSD	: Power Spectrum Density
PVA	: Position, Velocity, Attitude
PVT	: Position, Velocity, Timing

RANSAC	:	Random Sample Consensus
RVP	:	Relative Visual Positioning
SLAM	:	Simultaneous Localization and Mapping
SIFT	:	Scale Invariant Feature Transform
SINS	:	Strapdown Inertial Navigation System
SITAN	:	Sandia Inertial Terrain Aided Navigation
SSD	:	Sum of Squared Difference
SURF	:	Speeded Up Robust Features
TAN	:	Terrain Aided Navigation
TERCOM	:	Terrain Contour Matching
UAV	:	Unmanned Aerial Vehicle
UKF	:	Unscented Kalman Filter
VAN	:	Visual Aided Navigation
VO	:	Visual Odometry
VRW	:	Velocity Random Walk



LIST OF FIGURES

	<u>Page</u>
Figure 2.1: Body frame.	5
Figure 2.2: ECEF and NED frame.	6
Figure 2.3: Consecutive Euler rotations: yaw (left), pitch(middle), roll(right)	7
Figure 2.4: A point moving relative to an observer at the O frame origin.	9
Figure 2.5: A point moves in the $\hat{I}$ direction, while M frame is translating.	10
Figure 2.6: A point moves as M frame is translating & rotating.	11
Figure 2.7: X-Z Plane.	24
Figure 2.8: Altitude and vertical velocity error graph.	26
Figure 3.1: Velocity profile.	31
Figure 3.2: Latitude-longitude graph.	32
Figure 3.3: Velocity graph.	33
Figure 3.4: Specific force graph.	33
Figure 3.5: Body rates.	34
Figure 3.6: Fixed bias error.	36
Figure 3.7: Scale factor error.	37
Figure 3.8: Misalignment error.	38
Figure 3.9: Master-Slave demonstration.	40
Figure 3.10: Master-slave (on a gimbal plate) demonstration.	41
Figure 3.11: Trajectory of a slave INS (on rotating a FLIR).	43
Figure 3.12: Body rates of the slave INS.	43
Figure 3.13: Specific forces of the slave INS.	44
Figure 3.14: Velocity of the slave INS (resolved in NED frame).	44
Figure 3.15: LOS vectors from FLIR to a stationary target(2D).	46
Figure 3.16: LOS vectors from FLIR to a stationary target(3D).	47
Figure 3.17: Gyroscope output of the slave system.	47
Figure 3.18: Accelerometer output of the slave system.	48
Figure 3.19: Random vibration PSD Spectrum.	50
Figure 3.20: Generated random vibration.	51
Figure 3.21: Estimated PSD of the generated data vs PSD Profile.	51
Figure 4.1: Illustration of GNSS satellites orbiting earth.	53
Figure 4.2: Open-loop fusion scheme.	59
Figure 4.3: Closed-loop fusion scheme.	60
Figure 4.4: Critical timings of GNSS Integrated Navigation System.	60
Figure 4.5: 2D trajectory.	63
Figure 4.6: Euler angles.	63
Figure 4.7: Position errors standard deviations.	64
Figure 4.8: Velocity errors standard deviations.	64
Figure 4.9: Attitude standard deviations.	65
Figure 4.10: 2D trajectory.	66
Figure 4.11: Euler angles.	66
Figure 4.12: Position errors standard deviations.	67
Figure 4.13: Velocity errors standard deviations.	67
Figure 4.14: Attitude errors standard deviations.	68
Figure 5.1: Aircraft altitude measurements: direct vs. pressure-based.	69

Figure 5.2: Flight trajectory and 3D terrain map.	70
Figure 5.3: Illustration of DTED.	70
Figure 5.4: Illustration of correlation lines.	72
Figure 5.5: Flowchart of the proposed Multi-Hypothesis TAN algorithm.	79
Figure 5.6: Demonstration of Υ_1^ξ and Υ_0^ξ .	85
Figure 5.7: Display of measurements and their uncertainty ellipses.	86
Figure 5.8: 2D illustration of merging: before(left), after(right)	90
Figure 5.9: The demonstration of hypothesis merging/pruning.	91
Figure 5.10: Visualization of covariance expansion effect.	94
Figure 5.11: The bird's eye view of flight path.	95
Figure 5.12: NEES plot.	96
Figure 5.13: Single-run true velocity versus estimated velocity.	97
Figure 5.14: Single-run true heading versus estimated heading.	97
Figure 5.15: Average number of hypotheses.	98
Figure 6.1: A satellite image illustration.	101
Figure 6.2: Examples of key feature detectors in aerial images.	103
Figure 6.3: Pinhole camera model (3D).	104
Figure 6.4: Pinhole camera model (side view).	104
Figure 6.5: Generalized camera configuration.	106
Figure 6.6: 100 MP ortophoto downloaded from USGS.	110
Figure 6.7: Synthetically produced camera image.	111
Figure 6.8: Camera locations.	112
Figure 6.9: DLT positioning error - (SIFT).	112
Figure 6.10: DLT positioning error - (AKAZE).	113
Figure 6.11: DLT positioning error - (ORB).	113
Figure 6.12: DLT positioning error - (FAST).	113
Figure 6.13: A valid feature association example.	114

LIST OF TABLES

	<u>Page</u>
Table 3.1: Straight motion segment.	28
Table 3.2: Rotate motion segment.	29
Table 4.1: Attributes of INS.	59
Table 4.2: Attributes of GNSS.	59
Table 6.1: Comparison of feature point detectors. C.D.: <i>Corner Detector</i> , B.D.: <i>Blob Detector</i> , R.I.: <i>Rotation Invariant</i> , S.I.: <i>Scale Invariant</i> , A.I.: <i>Affine Invariant</i> , Rep: <i>Reproducibility</i> , Loc. Accu.: <i>Localization Accuracy</i> , Rob.: <i>Robustness</i> , Eff.: <i>Efficiency</i> , courtesy of Fraundorfer & Scaramuzza [1].	102
Table 6.2: Camera Extrinsic.	109
Table 6.3: Camera Intrinsic.	109

1. INTRODUCTION

The Inertial Navigation Systems (INS) determine the Position, Velocity, and Attitude (PVA) of a vehicle through integration of the specific force and rotation rate outputs from the Inertial Measurement Unit (IMU) [2]. The INS systems can be classified into two categories, namely the Gimballed INS and the Strapdown INS (SINS). The former utilizes a stable platform to isolate the IMU from the rotational motion of the vehicle, while the latter directly mounts the IMU on the rigid body of the vehicle. Despite the simpler mechanical design, the Strapdown INS is more commonly utilized in modern navigation systems due to its cost, size, and weight efficiency, making it ideal for implementation on various platforms such as guided missiles and land vehicles. The major disadvantage of the SINS lies in its computational complexity; unlike gimballed INS, attitude is updated numerically via gyroscope outputs. However, the computational burden of this issue has been alleviated with the advent of high-performance computing technologies. INSs possess inherent self-sufficiency, allowing them to calculate the PVA without the need for external inputs once they are initialized. Therefore, in contrast to Global Navigation Satellite Systems (GNSS), INSs exhibit immunity to jamming and spoofing [3, 4]. However, the various errors present in IMU measurements lead to a degradation in the PVA solution in INS. The magnitude of errors in an INS is a crucial factor in determining the grade of the INS and influencing the size and cost of the system. The cost of maintaining an accurate PVA solution exclusively through INS over extended navigation durations becomes prohibitively high. Hence, external systems like GNSS are often integrated with INS to address this issue. In contrast to INS, GNSS offers a low-bandwidth, worldwide accurate Position, Velocity, and Timing (PVT) solution that does not deteriorate over time. However, GNSS systems are susceptible to jamming and spoofing, and their signals can be blocked depending on the environment [4]. This situation jeopardizes the reliability and continuous availability of navigation solutions in scenarios where GNSS signals are compromised or entirely unavailable. To address these vulnerabilities, the integration of diverse system outputs installed on the platform is investigated to support INS. This multi-sensor approach not only improves the accuracy and reliability of the navigation system but also ensures its operational integrity in GNSS-denied environments.

1.1. Motivation and Scope

The utilization of information extracted from external sensors, such as cameras, distance measuring devices, and terrain maps, has become increasingly prevalent in recent years as a means of limiting or reducing PVA errors in INS. This thesis investigates advanced navigation techniques suitable for use in GNSS-denied environments with a focus on vision and terrain-aided navigation systems, particularly emphasizing terrain-aided navigation. Moreover, considerable effort is dedicated to the generation of synthetic IMU data.

1.2. Contributions

The dissertation presents several novel contributions to the field of navigation. First, a comprehensive trajectory shaping and IMU data generation tool has been developed. The trajectory shaping tool generates ground truth kinematic data that closely resemble real-world scenarios. The designed trajectories are then fed into an IMU data generation tool, which yields accelerometer and gyroscope measurements. This versatile data generation tool is designed not only to test standalone INS but also for master-slave platform configurations allowing to test transfer alignment algorithms. Furthermore, a multi-hypothesis based terrain aided navigation algorithm has been proposed, demonstrating improved performance compared to conventional terrain-aided navigation methods. Finally, vision-based localization methods are explored. Various visual feature descriptor methods are evaluated for their effectiveness in orthophoto datasets. In short, the contributions of this thesis are as follows:

- A methodology is proposed for generating accurate ground truth PVA trajectories and corresponding synthetic IMU data for an aircraft's INS. Additionally, it covers the generation of similar outputs for a secondary, or 'slave', INS system mounted on the same aircraft platform. This process also incorporates simulation of environmental effects, including vibrations, to ensure that the synthetic data closely mirror real-world conditions.
- A novel Multi-Hypothesis Filter (MHF) for terrain-aided navigation is proposed. These hypotheses can be considered as imaginary INS that are propagated through IMU measurements. Additionally, 2D position measurements are generated

within each hypothesis' uncertainty region, which are then fused with appropriate hypotheses. The main innovation of the proposed MHF framework lies in converting terrain height information into 2D horizontal measurements and then employing multiple Kalman filters that utilize these measurements. The simulation results indicate that the proposed MHF is robust to multi-modal topography and large initialization errors, and also enables navigation using sub-tactical grade IMUs.

- Various visual feature extraction methods are evaluated in the Absolute Visual Positioning (AVP) problem. Four distinct types of visual features are extracted from synthetically generated camera images, along with a reference orthophoto image. Subsequently, these features are matched, and the feature pixels from the camera images, along with their corresponding geodetic coordinates derived from the orthophoto correspondences, are input into the Direct Linear Transform (DLT) to estimate the camera's position. The positioning accuracy of various visual feature extraction methods are compared.

1.3. Thesis Outline

This thesis is structured as follows: Section-2 delves into crucial definitions and the mathematical fundamentals of inertial navigation. Section-3 concentrates on trajectory shaping and the generation of synthetic IMU data, including various definitions of motion segments, different types of IMU errors, and details related to their implementation. Section-4 briefly discusses GNSS-aided inertial navigation and presents simulation results of loosely coupled INS-GNSS fusion for two different scenarios. Section-5 is dedicated to terrain-aided navigation, beginning with traditional methods before focusing on the core contribution of the dissertation: a MHF specifically designed for terrain-aided navigation. Section-6 of the thesis is dedicated to vision-based navigation, primarily focusing on the concept and application of AVP.

2. BASICS OF INERTIAL NAVIGATION

2.1. Reference Frames

A frame consists of an origin point and a set of three orthogonal, right-handed axes. Frames are utilized in representing an object's position, velocity, and orientation and can be used interchangeably with the concept of a reference frame. An orthogonal coordinate system possesses six parameters of freedom, encompassing the position of the origin (o) and the orientation of the axes (x , y , and z). These parameters must be expressed with respect to another frame for proper definition. Every navigation problem thus requires the consideration of at least two coordinate frames.

2.1.1. Earth-Centered Earth-Fixed Frame (ECEF)

The Earth-Centered, Earth-Fixed (ECEF) frame has its origin situated at the center of the Earth. The x -axis of this frame is collocated with the intersection of the equatorial plane and the Prime Meridian, while its z -axis aligns with the earth's rotational axis, directed towards the North Pole. The y -axis is determined by employing the right-hand rule. ECEF frame is denoted by the letter e .

2.1.2. Inertial Frame(I)

The inertial frame is a frame of reference that does not rotate or accelerate relative to the universe. Its z -axis is aligned with the earth's rotational axis and its x -axis points towards fixed celestial objects. This frame can also be conceptualized as a non-rotating ECEF frame.

2.1.3. Local Navigation Frame

The origin of the local navigation frame is positioned at the center of the navigating platform. The orientation of its axes can be defined in various ways. One widely used convention is the North-East-Down (NED) frame, where the x -axis points towards true north, the y -axis towards the east, and the z -axis towards the local vertical direction. In

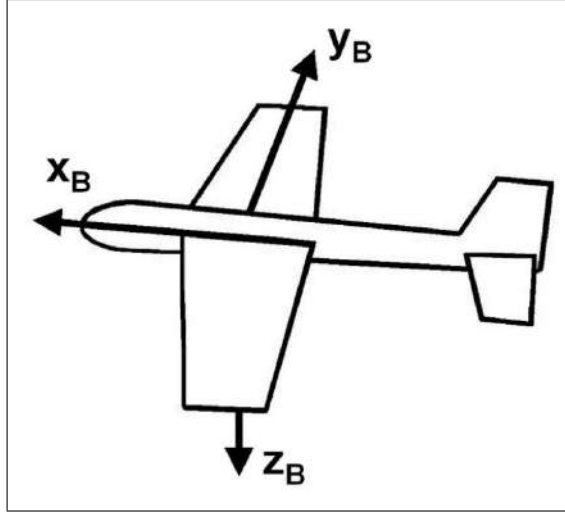


Figure 2.1: Body frame.

this thesis, the NED frame is denoted by the letter n .

2.1.4. Wander Azimuth Frame

The utilization of local navigation reference frames in navigation computations may result in imprecise results when operating at high latitudes, particularly when traversing the North Pole, where the transport rate becomes undefined. To mitigate such singularities, the wander azimuth frame is often employed for the calculation of mechanization equations. This frame can also be regarded as a local navigation frame with a fixed z -axis.

2.1.5. Body Frame

The body frame, denoted by b , has its origin at the vehicle's center, with its coordinate axes aligned with the vehicle's roll, pitch, and yaw axes. An illustration of the body frame is provided in Fig. 2.1. Within the context of this thesis, it is assumed that the frame representing the sensor triad of the IMU coincides with the body frame.

2.2. Basics of Coordinate Frame Mathematics

To estimate the PVA of a platform using measurements from an IMU, it is necessary to establish the kinematic relationship between the aforementioned frames, as well as

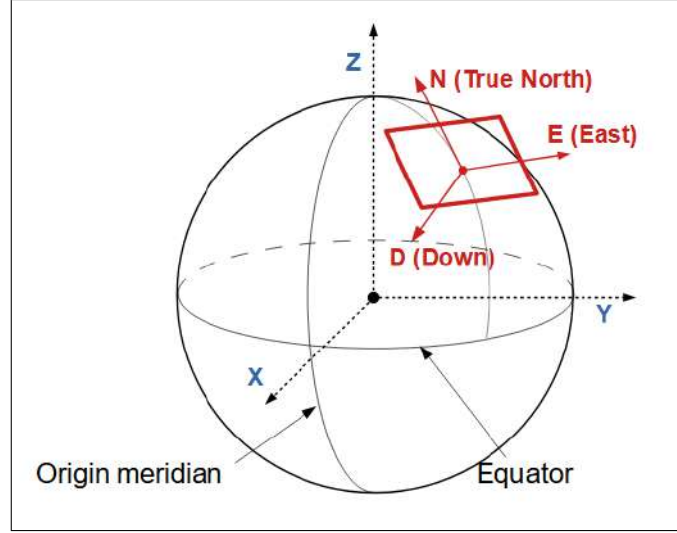


Figure 2.2: ECEF and NED frame.

the dynamic of this relationship. In the following section, we examine the different approaches to representing the attitude of an object and their respective time derivatives.

2.2.1. Kinematic Notation

In navigation, the dynamics of a reference frame are characterized relative to another frame. To capture the kinematic behavior, it is often necessary to employ a set of three coordinate frames to specify the position, velocity, and angular rate [5].

$$\mathbf{x}_{\beta\alpha}^{\gamma} \quad (2.1)$$

In (2.1), the object frame, denoted by α , is used to describe the motion of the frame of interest. The reference frame, denoted by β , represents the frame which object frame motion is relative to. The resolving frame, γ , specifies the coordinate frame in which the components of vector $\mathbf{x}$ are expressed. It is a fundamental requirement that the object and reference frames be distinct in order to capture motion. The choice of the resolving frame can be either the object or reference frame, or any other frame of reference.

2.2.2. Euler Representation

There are three distinct approaches to represent the attitude of one frame with respect to another. The Euler representation is the most intuitive, wherein the attitude is decomposed into three successive rotations as demonstrated in Fig. 2.3. The first

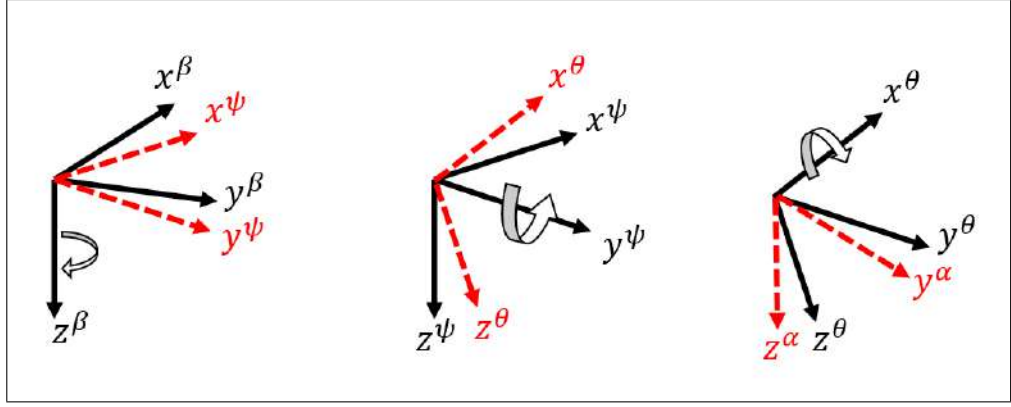


Figure 2.3: Consecutive Euler rotations: yaw (left), pitch (middle), roll (right)

rotation is performed around the z -axis of a reference frame in order to align the x -axis of the reference frame with the object frame. The resultant frame components following the initial rotation may be derived according to (2.2).

$$\begin{aligned} x^\psi &= x^\beta \cos \psi_{\beta\alpha} + y^\beta \sin \psi_{\beta\alpha} \\ y^\psi &= -x^\beta \sin \psi_{\beta\alpha} + y^\beta \cos \psi_{\beta\alpha} \\ z^\psi &= z^\beta \end{aligned} \quad (2.2)$$

Applying a rotation of angle $\theta_{\beta\alpha}$ to the y^ψ axis in a right-handed manner produces the "pitch" rotation, which is the second rotation in the sequence.

$$\begin{aligned} x^\theta &= x^\psi \cos \theta_{\beta\alpha} - z^\psi \sin \theta_{\beta\alpha} \\ y^\theta &= y^\psi \\ z^\theta &= x^\psi \sin \theta_{\beta\alpha} + z^\psi \cos \theta_{\beta\alpha} \end{aligned} \quad (2.3)$$

The final rotation in the sequence is the "roll" rotation, which involves a rotation of the x^θ axis by an angle of $\phi_{\beta\alpha}$.

$$\begin{aligned} x^\alpha &= x^\theta \\ y^\alpha &= y^\theta \cos \phi_{\beta\alpha} + z^\theta \sin \phi_{\beta\alpha} \\ z^\alpha &= -y^\theta \sin \phi_{\beta\alpha} + z^\theta \cos \phi_{\beta\alpha} \end{aligned} \quad (2.4)$$

2.2.3. Direction Cosine Matrix (DCM) Representation

The direction cosine matrix, denoted by C_{β}^{α} , is an orthogonal 3×3 matrix that transforms resolution of any vector from frame β to frame α . In (2.5), superscript α and β indicates frame in which the vector r is resolved.

$$\mathbf{r}^{\alpha} = C_{\beta}^{\alpha} \mathbf{r}^{\beta} \quad (2.5)$$

In (2.6) to (2.8) some of the properties of the DCM matrices are given

$$I_{3 \times 3} = C_{\beta}^{\alpha} C_{\alpha}^{\beta} \quad (2.6)$$

$$C_{\alpha}^{\beta} = (C_{\beta}^{\alpha})^{-1} \quad (2.7)$$

$$C_{\alpha}^{\beta} = C_{\gamma}^{\beta} C_{\alpha}^{\gamma} \quad (2.8)$$

The time derivative of a DCM matrix can be expressed as given in (2.9)

$$\dot{C}_{\alpha}^{\beta} = C_{\alpha}^{\beta} \Omega_{\beta\alpha}^{\alpha} \quad (2.9)$$

where $\Omega_{\beta\alpha}^{\alpha}$ is the skew-symmetric form of the angular rate vector $\omega_{\beta\alpha}^{\alpha}$ which is the instantaneous rate vector of frame β with respect to α .

$$\Omega_{\beta\alpha}^{\alpha} = S \left(\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \right) = \begin{bmatrix} 0 & -\omega_1 & +\omega_2 \\ \omega_1 & 0 & -\omega_3 \\ -\omega_2 & \omega_3 & 0 \end{bmatrix} \quad (2.10)$$

By considering the consecutive Euler rotations as described in equations (2.6), (2.7), and (2.8), it is possible to compute the transformation matrix from the β frame to the α frame using the following procedure.

2.2.4. Basics of Motion Involving Multiple Frames

The position of a point is denoted by the vector r that extends from the center of a reference frame origin to the point. When differentiating this vector, it becomes imperative to delineate the frame of reference, as the perceived motion of the point can vary based on the perspective from which it is observed.

- Motion involving a single fixed frame:

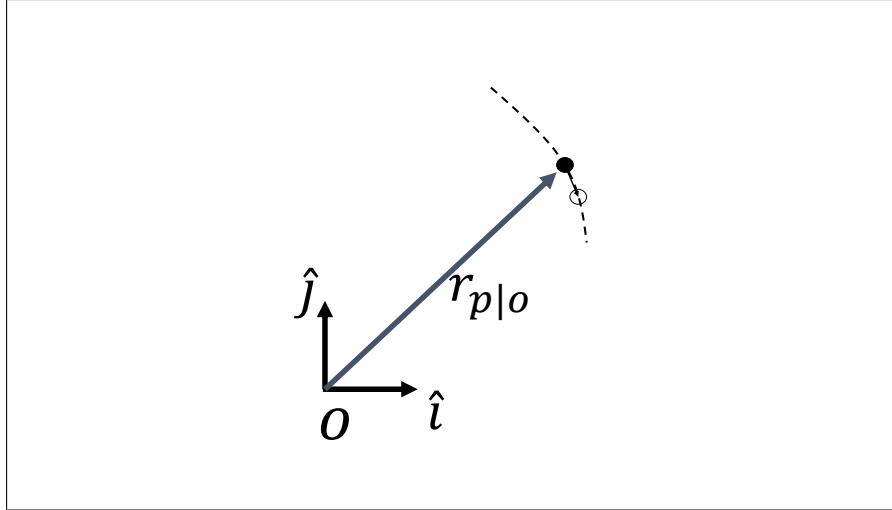


Figure 2.4: A point moving relative to an observer at the O frame origin.

If there is only one frame involved, the velocity of a point is evaluated by the time derivative of each components of position vector.

$$\begin{aligned} \left. \frac{d}{dt} \mathbf{r}_{p|O} \right|_O &= \left. \frac{d}{dt} \{x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}\} \right|_O \\ &= \dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}} + \dot{z}\hat{\mathbf{k}} \end{aligned} \quad (2.11)$$

In (2.11), O is an arbitrary fixed frame and $\left|_O\right.$ indicates that the derivative is taken with respect to O frame. Since there is only the O frame, the time derivative of the basis vectors with respect to the O frame is zero, as stated in (2.12).

$$\left. \frac{d}{dt} \hat{\mathbf{i}} \right|_O = \left. \frac{d}{dt} \hat{\mathbf{j}} \right|_O = \left. \frac{d}{dt} \hat{\mathbf{k}} \right|_O = 0 \quad (2.12)$$

- Motion involving a fixed and non-rotating moving frame:

If the point p is located on a moving (but not rotating) object and moves in a direction with respect to this moving frame, then the total velocity of the point p with respect to the frame O is the sum of the velocity of the frame M and the velocity of the point p with respect to the frame M . The Fig. 2.5. illustrates a translating object frame(M) that has a point located at $s\hat{\mathbf{I}}$ which also moves in the $\hat{\mathbf{I}}$ direction.

The position vector of the point p with respect to frame O can be written as the sum of the position of the frame M with respect to frame O and the position of

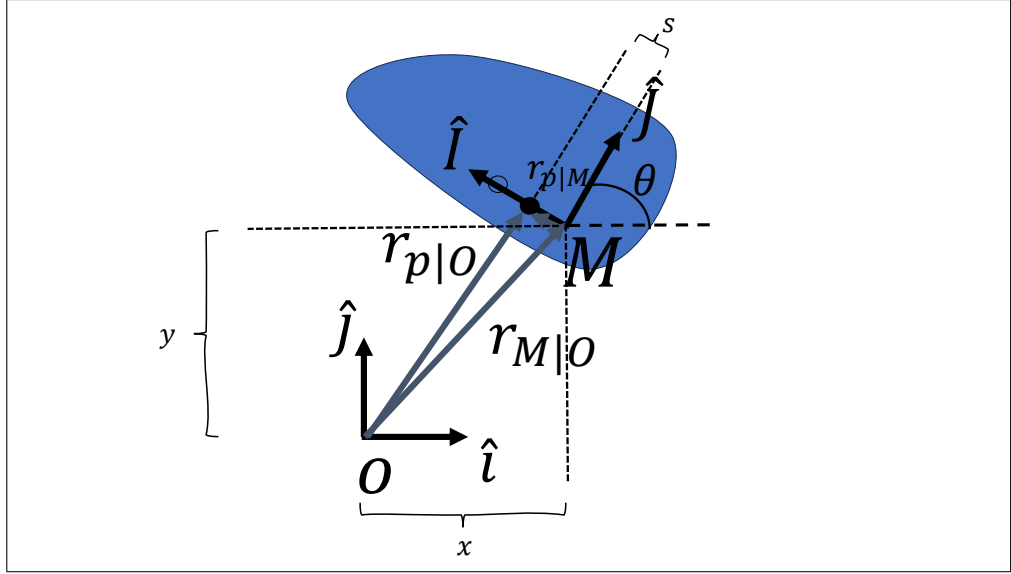


Figure 2.5: A point moves in the $\hat{I}$ direction, while M frame is translating.

the point p with respect to frame M .

$$\begin{aligned} \mathbf{r}_{p|O} &= \mathbf{r}_{M|O} + \mathbf{r}_{p|M} \\ &= x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + s\hat{\mathbf{I}} \end{aligned} \quad (2.13)$$

where the basis vectors of the M frame ($\hat{\mathbf{I}}$ and $\hat{\mathbf{J}}$) can be represented by the basis vectors of the O frame as given in (2.14-2.15).

$$\hat{\mathbf{I}} = \hat{\mathbf{j}}\cos(\theta) - \hat{\mathbf{i}}\sin(\theta) \quad (2.14)$$

$$\hat{\mathbf{J}} = \hat{\mathbf{j}}\sin(\theta) + \hat{\mathbf{i}}\cos(\theta) \quad (2.15)$$

The base vectors of M frame does not change with respect to O frame as the relative orientation stays the same. Thus, the time derivative of the base vector components of M frame with respect to O frame is also zero.

$$\left. \frac{d}{dt} \mathbf{r}_{p|O} \right|_O = \dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}} + \dot{s}\hat{\mathbf{I}} \quad (2.16)$$

- Motion involving a fixed and a rotating moving frame:

In this situation, an additional velocity is registered. This arises when the basis vectors of frame M undergo a change in relation to the basis vectors of frame

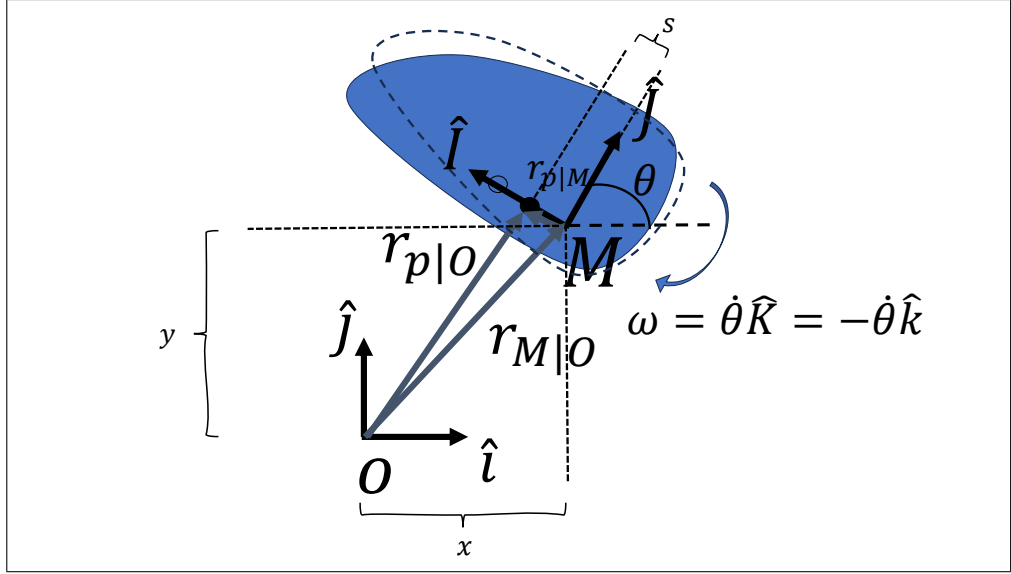


Figure 2.6: A point moves as M frame is translating & rotating.

O , primarily attributed to the rotation of frame M . In the case of a rotating M frame, the velocity of the point p with respect to the frame O can be expressed as follows:

$$\left. \frac{d}{dt} \mathbf{r}_{p|O} \right|_O = \dot{x} \hat{\mathbf{i}} + \dot{y} \hat{\mathbf{j}} + \dot{s} \hat{\mathbf{I}} + s \left. \frac{d}{dt} \hat{\mathbf{I}} \right|_O \quad (2.17)$$

where,

$$\begin{aligned} \left. \frac{d}{dt} \hat{\mathbf{I}} \right|_O &= -\hat{\mathbf{j}} \sin(\theta) \dot{\theta} - \hat{\mathbf{i}} \cos(\theta) \dot{\theta} \\ &= -\dot{\theta} \hat{\mathbf{k}} \times \hat{\mathbf{I}} \end{aligned} \quad (2.18)$$

Substituting (2.18) into (2.17);

$$\left. \frac{d}{dt} \mathbf{r}_{p|O} \right|_O = \dot{x} \hat{\mathbf{i}} + \dot{y} \hat{\mathbf{j}} + \dot{s} \hat{\mathbf{I}} - s \dot{\theta} \hat{\mathbf{k}} \times \hat{\mathbf{I}} \quad (2.19)$$

In (2.19), the expression $\dot{x} \hat{\mathbf{i}} + \dot{y} \hat{\mathbf{j}}$ denotes the velocity of the frame M as perceived from the frame O . Similarly, $\dot{s} \hat{\mathbf{I}}$ represents the velocity of point p when observed from frame M . Furthermore, $s \dot{\theta} \hat{\mathbf{k}} \times \hat{\mathbf{I}}$ elucidates the velocity of point p that arises due to the rotational movement of frame M relative to frame O . The subsequent equation, referred to as the 'Transport Theorem', provides a formulation for the

differentiation of any vector $\mathbf{R}$ within the frame of reference O [6].

$$\left. \frac{d}{dt} \mathbf{R} \right|_O = \left. \frac{d}{dt} \mathbf{R} \right|_M + \boldsymbol{\omega}_{M|O} \times \mathbf{R} \quad (2.20)$$

2.3. Inertial Navigation Mechanization Equations

In this subsection, we derive the continuous navigation equations for both the inertial frame and the navigation frame mechanization. Subsequently, a discrete implementation of these navigation mechanization equations is presented. Before deriving the mechanization equations, the navigation equation is provided as given in (2.21) [2].

$$\left. \frac{d^2}{dt^2} \mathbf{r} \right|_i = \mathbf{f} + \mathbf{g} \quad (2.21)$$

where $\mathbf{f}$ represents the specific force as measured by the accelerometer triad, while $\mathbf{g}$ denotes the gravitational vector arising from mass attraction. The term $\left. \frac{d^2}{dt^2} \mathbf{r} \right|_i$ represents the total acceleration of the point as it navigates in the vicinity of the Earth.

2.3.1. Velocity Mechanization In Inertial Frame

Utilizing the transport theorem as outlined in (2.20), one can express the derivative of the position vector $\mathbf{r}$ with respect to inertial frame as follows:

$$\left. \frac{d}{dt} \mathbf{r} \right|_i = \left. \frac{d}{dt} \mathbf{r} \right|_e + \boldsymbol{\omega}_{ie} \times \mathbf{r} \quad (2.22)$$

where $\mathbf{v}_e \triangleq \left. \frac{d}{dt} \mathbf{r} \right|_e$ which corresponds to the velocity vector with respect to ground. Taking the derivative of both sides of (2.22) with respect to inertial frame;

$$\left. \frac{d^2}{dt^2} \mathbf{r} \right|_i = \left. \frac{d}{dt} \mathbf{v}_e \right|_i + \left. \{ \boldsymbol{\omega}_{ie} \times \mathbf{r} \} \right|_i \quad (2.23)$$

$$= \left. \frac{d}{dt} \mathbf{v}_e \right|_i + \boldsymbol{\omega}_{ie} \times \left. \frac{d}{dt} \mathbf{r} \right|_i \quad (2.24)$$

substituting (2.21) and (2.22) into (2.24),

$$\left. \frac{d}{dt} \mathbf{v}_e \right|_i = \mathbf{f} - \boldsymbol{\omega}_{ie} \times \mathbf{v}_e - \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{r} + \mathbf{g} \quad (2.25)$$

In (2.25), the term $\boldsymbol{\omega}_{ie} \times \mathbf{v}_e$ represents the Coriolis acceleration, while $\boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{r}$ denotes the centripetal acceleration arising due to Earth's rotation. Notably, the mass

attraction acceleration and the centripetal acceleration are indistinguishable. Consequently, their combined effect, often referred to as the 'plumb bob', is employed to characterize the local gravity vector (2.26).

$$\mathbf{g}_l = \mathbf{g} - \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{r} \quad (2.26)$$

2.3.2. Velocity Mechanization In Navigation Frame

The navigation frame velocity mechanization equation can be obtained follows:

$$\left. \frac{d}{dt} \mathbf{v}_e \right|_n = \left. \frac{d}{dt} \mathbf{v}_e \right|_i - \boldsymbol{\omega}_{in} \times \mathbf{v}_e \quad (2.27)$$

where

$$\boldsymbol{\omega}_{in} = \boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{en} \quad (2.28)$$

Substituting (2.21) and (2.28) into (2.27);

$$\begin{aligned} \left. \frac{d}{dt} \mathbf{v}_e \right|_n &= \mathbf{f} - \boldsymbol{\omega}_{ie} \times \mathbf{v}_e - \boldsymbol{\omega}_{ie} \times \boldsymbol{\omega}_{ie} \times \mathbf{r} + \mathbf{g} - (\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{en}) \times \mathbf{v}_e \\ &= \mathbf{f} - (2\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{en}) \times \mathbf{v}_e + \mathbf{g}_l \end{aligned} \quad (2.29)$$

Given that the specific force is measured in the body frame, we can represent the velocity dynamics as illustrated in the subsequent equation, with all vectors resolved in the navigation frame.

$$\dot{\mathbf{v}}_e^n = C_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_e^n + \mathbf{g}_l^n \quad (2.30)$$

In (2.30), the vector $\boldsymbol{\omega}_{ie}^n$ signifies the Earth frame's rotation relative to the inertial frame. Similarly, $\boldsymbol{\omega}_{en}^n$ represents the rotation of the navigation frame with respect to the Earth frame. Evaluation of both of these vectors are given in (2.31) and (2.32)

$$\boldsymbol{\omega}_{ie}^n = \begin{bmatrix} \omega_{earth} \cos(l) \\ 0 \\ -\omega_{earth} \sin(l) \end{bmatrix} \quad (2.31)$$

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \frac{v_e}{R_e + h} \\ -\frac{v_n}{R_n + h} \\ \frac{v_e \tan(l)}{R_e + h} \end{bmatrix} \quad (2.32)$$

where ω_{earth} represents the Earth's rotation rate, that is 7.2921×10^{-5} rad/s. Here, l denotes the latitude and h signifies the altitude of the platform. The velocities v_n , v_e , and v_d correspond to the north, east, and down directions, respectively. Lastly, R_n and R_e are the meridian and traverse radii of curvature for the earth whose values are provided in (2.35) and (2.36).

2.4. Attitude Mechanization Equation

Since the sensitive axis of the IMU changes relative to the reference frame when the platform changes its attitude, updating the C_b^n matrix prior to integrating the velocity is essential. The dynamics of the C_b^n matrix are described by (2.33).

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b = C_b^n \Omega_{ib}^b - \Omega_{in}^n C_b^n \quad (2.33)$$

where Ω_{ib}^b and Ω_{in}^n are the skew-symmetric matrix forms of ω_{ib}^b and ω_{in}^n , respectively. ω_{ib}^b represents the gyroscope readings, while ω_{in}^n can be calculated using the following expression.

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n \quad (2.34)$$

2.5. Earth Curvatures, Gravity and Position Calculation

The ECEF coordinate system allows navigation with respect to Earth's center. However, for many navigation tasks, one's position relative to the Earth's uneven surface is essential. Given the complexities of capturing its true form, the surface is often simplified to a standard shape aligned with the mean sea level. Satellite navigation introduced the ability to determine positions on Earth based on a shared reference: the satellite constellation. This has resulted in the creation of universal ellipsoid models, primarily the World Geodetic System 1984 (WGS 84) [7] and the International Terrestrial Reference Frame (ITRF) [8]. Relative positioning on Earth's surface is represented using three perpendicular coordinates, corresponding to the local navigation frame's axes. The vertical distance from an object to the surface is termed height or altitude. Latitude represents the north-south coordinate where the surface's normal intersects, while longitude defines its east-west axis counterpart. The vector, which consists of latitude, longitude, and height components, is referred to as the 'curvilinear position' or 'geodetic position'. It's important to explain the ellipsoid's radii of curvature at this point. The

radius of curvature for moving from north to south is called the meridian radius of curvature, R_n . This radius shows how the geodetic latitude changes when moving along a meridian. Simply, it's the bend of the ellipsoid's slice in the north-down direction where we're looking. For movement from east to west, there's a radius of curvature called the transverse radius of curvature, R_e . This radius shows how the longitude changes on the surface perpendicular to a meridian, which isn't the same as moving along a line of constant latitude. The expressions for R_n and R_e are provided below.

$$R_n = \frac{R_0(1 - e^2)}{(1 - e^2 \sin(l))^{1.5}} \quad (2.35)$$

$$R_e = \frac{R_0}{\sqrt{1 - e^2 \sin^2(l)}} \quad (2.36)$$

where $e = 0.0818191908425$ and $R_0 = 6378137$ meters. The simplest gravity calculation model is the 'somigliana model' from the WGS 84 datum [7]. In this model, the acceleration due to gravity depends on latitude and can be described by the subsequent formula.

$$g(l) \approx 9.78032 \times \frac{1 + 0.001932 \sin^2(l)}{\sqrt{1 - e^2 \sin^2(l)}} m/s^2 \quad (2.37)$$

There is also more accurate spherical harmonics model for gravity calculation which utilizes EGM 96 gravitational model [9]. Latitude, longitude and altitude dynamics are given by the following equations:

$$\dot{l} = \frac{v_n}{R_n + h} \quad (2.38)$$

$$\dot{L} = \frac{v_e}{(R_e + h) \cos(l)} \quad (2.39)$$

$$\dot{h} = -v_d \quad (2.40)$$

2.6. Discrete Implementation of Inertial Navigation Equations

In the previous section, we presented the continuous forms of PVA dynamics. However, for digital environments, their discrete versions must be implemented. Early attempts to produce a strapdown INS were limited, due to the computer technologies available at the time. Apart from the physical size of early computers that delayed the development of strapdown systems, particularly for airborne applications, the lack of computing

speed, or throughput, was a major obstacle to the achievement of fast and accurate attitude computation. Initial efforts to create a strapdown INS faced challenges due to the computer technologies back then. Not only was the large size of early computers a setback for developing strapdown INS, especially for airborne use, but the slow computing speed also made it tough to quickly and accurately calculate attitude [2, 5].

2.6.1. Attitude Calculation

In this thesis, DCM is utilized to describe the orientation relationship between the body-fixed frame and the navigation frame. The attitude update consists of two steps. First, the change in the body frame due to the platform's maneuver is assessed using the gyroscope measurement. Second, the change in the navigation frame is evaluated. Because the rotation in the navigation frame is significantly smaller than the change in the body frame, the update rate for the navigation frame could be set to be slower than that for the body frame [10]. However, in this dissertation, the update rates for both the navigation and body frames are set to be identical. Before diving into the attitude calculation formulas, it is beneficial to introduce nomenclature related to the attitude equations. Local navigation and the body frame are denoted by n and b respectively. ω_{ib}^b in (2.43) represents the rate vector measured by the IMU's gyros. k is computer cycle index. In (2.41) discrete attitude update formula is given.

$$C_{b(k)}^{n(k)} = C_{n(k-1)}^{n(k)} C_{b(k-1)}^{n(k-1)} C_{b(k)}^{b(k-1)} \quad (2.41)$$

where $C_{b(k)}^{b(k-1)}$ is the DCM matrix that accounts for the body frame rotation in relation to its previous orientation, while $C_{n(k-1)}^{n(k)}$ represents the DCM matrix that accounts for the navigation frame's rotation from the prior cycle. $C_{b(k)}^{b(k-1)}$ can be evaluated as given by the following formula:

$$C_{b(k)}^{b(k-1)} = I + \frac{\sin(\phi_k)}{\phi_k} S(\phi_k) + \frac{1 - \cos(\phi_k)}{\phi_k^2} S(\phi_k) S(\phi_k) \quad (2.42)$$

where ϕ_k is the rotation vector evaluated by the integral formula given in (2.43) and ϕ_k is the norm of ϕ_k .

$$\phi_k = \int_{t_{k-1}}^{t_k} [\omega_{ib}^b(t) + 0.5(\alpha(t) \times \omega_{ib}^b(t))] dt \quad (2.43)$$

where $\alpha(t)$ is the integral of $\omega_{ib}^b(t)$ from t_{k-1} to t . In (2.43) $0.5(\alpha(t) \times \omega_{ib}^b(t))$ part corresponds to attitude change caused by the coning motion[11, 12]. The phenomenon

of coning motion is observed when there's a directional shift in the rotation vector as the body undergoes rotations. The rate-integrating gyroscopes typically produce rotation vectors that are compensated for coning effects by reducing the output rate and implementing sophisticated coning correction methodologies. To calculate $C_{n(k-1)}^{n(k)}$ (2.44) can be used

$$C_{n(k-1)}^{n(k)} = I - \frac{\sin(\zeta_k)}{\zeta_k} S(\zeta_k) + \frac{1 - \cos(\zeta_k)}{\zeta_k^2} S(\zeta_k) S(\zeta_k) \quad (2.44)$$

where

$$\zeta_k = \int_{t_{k-1}}^{t_k} \omega_{en}^n dt \approx \begin{bmatrix} \frac{v_e}{R_e+h} \\ -\frac{v_n}{R_n+h} \\ \frac{v_e \tan(l)}{R_e+h} \end{bmatrix} \Delta t \quad (2.45)$$

In (2.45) Δt corresponds to IMU sampling period.

2.6.2. Velocity Calculation

The continuous form of velocity dynamics equation is presented in (2.30). To transition to its discrete formulation, one can employ (2.46).

$$\mathbf{v}^n(k) = \mathbf{v}^n(k-1) + \Delta \mathbf{v}_{sf} + \Delta \mathbf{v}_{g/cor} \quad (2.46)$$

where

$$\Delta \mathbf{v}_{sf} = C_{n(k-1)}^{n(k)} C_{b(k-1)}^{n(k-1)} \left(\beta_k + \frac{1}{2} (\alpha_k \times \beta_k) \right) \quad (2.47)$$

$$\Delta \mathbf{v}_{g/cor} \approx \{ \mathbf{g}_l^n(k-1) - [2\omega_{ie}^n(k-1) + \omega_{en}^n(k-1)] \times \mathbf{v}^n(k-1) \} \Delta t \quad (2.48)$$

In (2.47), β_k is the integrated accelerometer output at time k while $\frac{1}{2}(\alpha_k \times \beta_k)$ is the sculling correction term.

2.6.3. Position Calculation

The latitude, longitude and altitude update equations in the discrete form are given in (2.51)(2.50)(2.51), respectively.

$$l(k) = l(k-1) + \frac{v_n(k-1/2)}{R_n+h} \Delta t \quad (2.49)$$

$$L(k) = L(k-1) + \frac{v_e(k-1/2)}{(R_e+h)\cos(l(k))} \Delta t \quad (2.50)$$

$$h(k) = h(k-1) - v_d(k-1/2) \Delta t \quad (2.51)$$

In the equations above, $(k - 1/2)$ represents the average value between the k and $k - 1$ indices.

2.7. Derivation of Error Dynamics

The errors in PVA within the INS deteriorate over time, regardless of sensor quality. The evolution of these navigation solution errors can be predicted using the PVA error dynamics equations. Subsequent subsections derive the position, velocity, and attitude error dynamics equations. In these derivations, we use $\hat{\cdot}$ symbol to denote the quantity is the computed variable in the navigation computer.

2.7.1. Position Error Dynamics

Let $\mathbf{p} = [l, L, h]^T$ denotes the position vector composed of latitude, longitude, and altitude. We define the error in position as given in (2.52)

$$\delta \mathbf{p} \triangleq \hat{\mathbf{p}} - \mathbf{p} \quad (2.52)$$

taking the time derivative of (2.52):

$$\dot{\delta \mathbf{p}} \triangleq \dot{\hat{\mathbf{p}}} - \dot{\mathbf{p}} \quad (2.53)$$

where $\dot{\hat{\mathbf{p}}}$ and $\dot{\mathbf{p}}$ are both functions of position and the velocity vector. They can be expressed as:

$$\dot{\delta \mathbf{p}} \triangleq \dot{\hat{\mathbf{p}}}(\hat{\mathbf{p}}, \hat{\mathbf{v}}) - \dot{\mathbf{p}}(\mathbf{p}, \mathbf{v}) \quad (2.54)$$

Using (2.52), $\dot{\mathbf{p}}(\mathbf{p}, \mathbf{v})$ in (2.54) can be rewritten as:

$$\dot{\mathbf{p}}(\mathbf{p}, \mathbf{v}) = \dot{\mathbf{p}}(\hat{\mathbf{p}} - \delta \mathbf{p}, \hat{\mathbf{v}} - \delta \mathbf{v}) \quad (2.55)$$

Ignoring the higher-order terms, the expression $\dot{\mathbf{p}}(\hat{\mathbf{p}} - \delta \mathbf{p}, \hat{\mathbf{v}} - \delta \mathbf{v})$ can be expanded using the Taylor series, which is presented as:

$$\dot{\mathbf{p}}(\hat{\mathbf{p}} - \delta \mathbf{p}, \hat{\mathbf{v}} - \delta \mathbf{v}) = \dot{\mathbf{p}}(\hat{\mathbf{p}}, \hat{\mathbf{v}}) - \left. \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{p}} \right|_{\hat{\mathbf{p}}, \hat{\mathbf{v}}} \delta \mathbf{p} - \left. \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}} \right|_{\hat{\mathbf{p}}, \hat{\mathbf{v}}} \delta \mathbf{v} \quad (2.56)$$

Substituting (2.56) into (2.54):

$$\dot{\delta \mathbf{p}} = \left. \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{p}} \right|_{\hat{\mathbf{p}}, \hat{\mathbf{v}}} \delta \mathbf{p} + \left. \frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}} \right|_{\hat{\mathbf{p}}, \hat{\mathbf{v}}} \delta \mathbf{v} \quad (2.57)$$

where, $\frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{p}}$ and $\frac{\partial \dot{\mathbf{p}}}{\partial \mathbf{v}}$ are given in (2.58) and (2.59).

$$\frac{\partial}{\partial \mathbf{p}} \dot{\mathbf{p}}(\mathbf{p}, \mathbf{v}) = \begin{bmatrix} \frac{\partial \dot{l}}{\partial l} & \frac{\partial \dot{l}}{\partial L} & \frac{\partial \dot{l}}{\partial h} \\ \frac{\partial \dot{L}}{\partial l} & \frac{\partial \dot{L}}{\partial L} & \frac{\partial \dot{L}}{\partial h} \\ \frac{\partial \dot{h}}{\partial l} & \frac{\partial \dot{h}}{\partial L} & \frac{\partial \dot{h}}{\partial h} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{v_n}{(R_n+h)^2} \\ \frac{v_e \sin(l)}{(R_e+h) \cos^2(l)} & 0 & -\frac{v_e \sec(l)}{(R_e+h)^2} \\ 0 & 0 & 0 \end{bmatrix} \quad (2.58)$$

$$\frac{\partial}{\partial \mathbf{v}} \dot{\mathbf{p}}(\mathbf{p}, \mathbf{v}) = \begin{bmatrix} \frac{\partial \dot{l}}{\partial v_n} & \frac{\partial \dot{l}}{\partial v_e} & \frac{\partial \dot{l}}{\partial v_d} \\ \frac{\partial \dot{L}}{\partial v_n} & \frac{\partial \dot{L}}{\partial v_e} & \frac{\partial \dot{L}}{\partial v_d} \\ \frac{\partial \dot{h}}{\partial v_n} & \frac{\partial \dot{h}}{\partial v_e} & \frac{\partial \dot{h}}{\partial v_d} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{\sec(l)}{R_e+h} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.59)$$

substituting (2.58) and (2.59) into (2.57),

$$\begin{aligned} \delta \dot{\mathbf{p}} &= \begin{bmatrix} 0 & 0 & -\frac{v_n}{(R_n+h)^2} \\ \frac{v_e \sin(l)}{(R_e+h) \cos^2(l)} & 0 & -\frac{v_e \sec(l)}{(R_e+h)^2} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta l \\ \delta L \\ \delta h \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{\sec(l)}{R_e+h} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_d \end{bmatrix} \end{aligned} \quad (2.60)$$

2.7.2. Velocity Error Dynamics

The velocity error dynamics equation is derived using a methodology analogous to that used for the position error dynamics equation. Let $\delta \mathbf{v}$ represent the error in the velocity vector, as detailed in the subsequent expression:

$$\delta \mathbf{v} \triangleq \hat{\mathbf{v}} - \mathbf{v} \quad (2.61)$$

Taking the time derivative of (2.61), the following expression is obtained.

$$\dot{\delta \mathbf{v}} \triangleq \dot{\hat{\mathbf{v}}} - \dot{\mathbf{v}} \quad (2.62)$$

where $\dot{\mathbf{v}}$ is given in (2.30) and $\dot{\hat{\mathbf{v}}}$ is the computed version of $\dot{\mathbf{v}}$ as given in (2.63)

$$\dot{\hat{\mathbf{v}}}^n = \hat{C}_b^n \dot{\hat{\mathbf{f}}}^b - (2\hat{\omega}_{ie}^n + \hat{\omega}_{en}^n) \times \hat{\mathbf{v}}^n + \hat{\mathbf{g}}_l^n \quad (2.63)$$

where,

$$\hat{C}_b^n = (I - S(\psi))C_b^n \quad (2.64)$$

By substituting (2.64) into (2.63) and then subtracting (2.30) from (2.63), we obtain:

$$\delta \dot{\mathbf{v}}^n = S(C_B^n \mathbf{f}^b) \psi + C_B^n \delta \mathbf{f}^b - (2\delta \omega_{ie}^n + \delta \omega_{en}^n) \times \mathbf{v}^n - (2\omega_{ie}^n + \omega_{en}^n) \times \delta \mathbf{v}^n + \delta \mathbf{g}^n \quad (2.65)$$

where $\delta \omega_{ie}^n$ and $\delta \omega_{en}^n$ are given in (2.66),(2.67).

$$\begin{aligned} \delta \omega_{ie}^n &= \frac{\partial}{\partial \mathbf{p}} \omega_{ie}^n \delta \mathbf{p} \\ &= \begin{bmatrix} -\sin(l)\Omega_{ie} \\ 0 \\ -\cos(l)\Omega_{ie} \end{bmatrix} \delta l \end{aligned} \quad (2.66)$$

$$\begin{aligned} \delta \omega_{en}^n &= \frac{\partial}{\partial \mathbf{p}} \omega_{en}^n \delta \mathbf{p} + \frac{\partial}{\partial \mathbf{v}} \omega_{en}^n \delta \mathbf{v} \\ \delta \omega_{EN}^n &= \begin{bmatrix} 0 & 0 & -\frac{v_e}{(R_e+h)^2} \\ 0 & 0 & \frac{v_n}{(R_n+h)^2} \\ -\frac{v_e \sec^2(l)}{R_e+h} & 0 & \frac{v_e \tan(l)}{(R_e+h)^2} \end{bmatrix} \begin{bmatrix} \delta l \\ \delta L \\ \delta h \end{bmatrix} \\ &+ \begin{bmatrix} 0 & \frac{1}{R_e+h} & 0 \\ -\frac{1}{R_n+h} & 0 & 0 \\ 0 & -\frac{\tan(l)}{R_e+h} & 0 \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_d \end{bmatrix} \end{aligned} \quad (2.67)$$

The computation of $\delta \mathbf{g}^n$ is outlined in the following part. The down component of the gravity vector, as a function of both latitude and altitude, can be found in (2.68).

$$g_z(l, h) \approx \frac{9.78(1 + 5.3 \times 10^{-3} \sin^2(l) - 5.9 \times 10^{-6} \sin^2(2l))}{(1 + \frac{h}{R})^2} \quad (2.68)$$

and plumb bob gravity vector is expressed as:

$$\mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ g_z(l, h) \end{bmatrix} + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \quad (2.69)$$

where $[\eta_x, \eta_y, \eta_z]^T$ is the gravity anomaly vector. The error in gravity can be articulated as:

$$\delta \mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ \delta g_z(l, h) \end{bmatrix} \quad (2.70)$$

where

$$\begin{aligned}
\delta g_z(l, h) &= \frac{\partial}{\partial l} g_z(l, h) \delta l + \frac{\partial}{\partial h} g_z(l, h) \delta h \\
&\approx \frac{\partial}{\partial h} g_z(l, h) \delta h \\
&\approx -2 \times \frac{9.78}{R} \delta h
\end{aligned} \tag{2.71}$$

2.7.3. Attitude Error Dynamics

In (2.33), the continuous form of the attitude dynamics equation is presented. The attitude error can be represented by the vector ψ , which encompasses small tilt errors around the navigation axes. The relationship between the true C_b^n and the computed $\hat{C}_b^n$ is provided in the subsequent expression.

$$\hat{C}_b^n = (I - S(\psi)) C_b^n \tag{2.72}$$

multiplying both sides with C_n^b then, $S(\psi)$ can be obtained as follows

$$S(\psi) = I - \hat{C}_b^n C_n^b \tag{2.73}$$

taking the derivative of (2.73),

$$S(\dot{\psi}) = -\dot{\hat{C}}_b^n (C_b^n)^T - \hat{C}_b^n (\dot{C}_b^n)^T \tag{2.74}$$

where

$$\dot{\hat{C}}_b^n = \hat{C}_b^n \hat{\Omega}_{ib}^b - \hat{\Omega}_{in}^n \hat{C}_b^n \tag{2.75}$$

the following expressions can be obtained by substituting (2.75) and (2.72) into (2.74),

$$\begin{aligned}
S(\dot{\psi}) &= -\hat{C}_b^n \hat{\Omega}_{ib}^b (C_b^n)^T + \hat{\Omega}_{in}^n \hat{C}_b^n (C_b^n)^T - \hat{C}_b^n (-\Omega_{ib}^b (C_b^n)^T + (C_b^n)^T \Omega_{in}^n) \\
&= -\hat{C}_b^n \delta \Omega_{ib}^b (C_b^n)^T + (\Omega_{in}^n + \delta \Omega_{in}^n) (I - S(\psi)) - (I - S(\psi)) \Omega_{in}^n \\
&\approx -C_b^n \delta \omega_{ib}^b C_b^{nT} + \delta \omega_{in}^n + S(\psi \times \omega_{in}^n)
\end{aligned} \tag{2.76}$$

taking the inverse skew-symmetric form of the final expression in (2.76) gives the following equation for the attitude error dynamics.

$$\dot{\psi} = -C_b^n \delta \omega_{ib}^b + \delta \omega_{in}^n - \omega_{in}^n \times \psi \tag{2.77}$$

2.8. State-space Demonstration of Error Dynamics Equations

In preceding sections, the error dynamics equations for PVA were derived. These equations can be consolidated into a unified state-space matrix representation as illustrated below.

$$\begin{bmatrix} \dot{\psi} \\ \delta \mathbf{v}^n \\ \delta \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} F_{\psi\psi} & F_{\psi V} & F_{\psi p} \\ F_{V\psi} & F_{VV} & F_{Vp} \\ F_{p\psi} & F_{pV} & F_{pp} \end{bmatrix} \begin{bmatrix} \psi \\ \delta \mathbf{v}^n \\ \delta \mathbf{p} \end{bmatrix} + \begin{bmatrix} -C_b^n & 0 & 0 \\ 0 & C_b^n & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{\omega}_{ib}^b \\ \delta \mathbf{f}^b \\ 0 \end{bmatrix} \quad (2.78)$$

where the individual F submatrices are provided in equations (2.79) through (2.86).

$$F_{\psi\psi} = \begin{bmatrix} 0 & \cos(l)\omega_{ie} - \frac{V_e \tan(l)}{R_e+h} & \frac{V_n}{R_n+h} \\ -\cos(l)\omega_{ie} + \frac{V_e \tan(l)}{R_e+h} & 0 & -\sin(l)\omega_{ie} + \frac{V_e}{R_e+h} \\ -\frac{V_n}{R_n+h} & \sin(l)\omega_{ie} - \frac{V_e}{R_e+h} & 0 \end{bmatrix} \quad (2.79)$$

$$F_{\psi V} = \begin{bmatrix} 0 & \frac{1}{R_e+h} & 0 \\ -\frac{1}{R_n+h} & 0 & 0 \\ 0 & -\frac{\tan(l)}{R_e+h} & 0 \end{bmatrix} \quad (2.80)$$

$$F_{\psi p} = \begin{bmatrix} 0 & 0 & -\frac{V_e}{(R_e+h)^2} \\ 0 & 0 & \frac{V_n}{(R_n+h)^2} \\ -\frac{V_e \sec^2(l)}{R_e+h} & 0 & \frac{V_e \tan(l)}{(R_e+h)^2} \end{bmatrix} \quad (2.81)$$

$$F_{V\psi} = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix} \quad (2.82)$$

$$F_{Vp} = \begin{bmatrix} 0 & -V_D & V_E \\ V_D & 0 & -V_N \\ -V_E & V_N & 0 \end{bmatrix} \begin{bmatrix} -2\sin(l)\Omega_{ie} & \frac{1}{R_e+h} & 0 \\ -\frac{1}{R_n+h} & 0 & 0 \\ -2\cos(l)\Omega_{ie} & -\frac{\tan(l)}{R_e+h} & 0 \end{bmatrix} + G_p \quad (2.83)$$

$$F_{p\psi} = \begin{bmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{\dot{L}}{V_e} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.84)$$

$$F_{pp} = \begin{bmatrix} 0 & 0 & -\frac{\dot{L}}{R_n+h} \\ \dot{L} \tan(l) & 0 & -\frac{\dot{L}}{R_e+h} \\ 0 & 0 & 0 \end{bmatrix} \quad (2.85)$$

$$F_{p\psi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad G_p = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2\frac{g}{R} \end{bmatrix} \quad (2.86)$$

2.9. Augmenting the Error State

Although the state-space representation of errors in (2.78) is suitable for many basic navigation problems where $\delta\omega_{ib}^b$ and $\delta\mathbf{f}^b$ are modeled as random constants, it is important to recognize that IMU errors can be diverse. Most of these errors do not follow a zero-mean Gaussian distribution. Therefore, for accurate error propagation, they should be modeled in (2.78) through state augmentation. For example, bias instability errors in the accelerometer ($\mathbf{f}_{Bins}$) and gyroscope (ω_{Bins}) are modeled as state variables in (2.87).

$$\begin{bmatrix} \dot{\psi} \\ \delta\dot{\mathbf{V}}^n \\ \delta\dot{\mathbf{p}} \\ \omega_{Bins} \\ \mathbf{f}_{Bins} \end{bmatrix} = \begin{bmatrix} F_{\psi\psi} & F_{\psi V} & F_{\psi p} & \mathbf{0} & \mathbf{0} \\ F_{V\psi} & F_{VV} & F_{Vp} & \mathbf{0} & \mathbf{0} \\ F_{p\psi} & F_{pV} & F_{pp} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -M_\omega & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -M_f \end{bmatrix} \begin{bmatrix} \psi \\ \delta\mathbf{V}^n \\ \delta\mathbf{p} \\ \omega_{Bins} \\ \mathbf{f}_{Bins} \end{bmatrix} \quad (2.87)$$

$$+ \begin{bmatrix} -C_b^n G_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_b^n G_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & G_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & G_4 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \end{bmatrix}$$

Where M_ω and M_f are diagonal matrices, with each diagonal entry representing the inverse of the correlation time as depicted in (2.88). The vectors $\mathbf{u}_1$ to $\mathbf{u}_4$ serve as process noise vectors, characterized by a multivariate Gaussian probability density function $N(\mathbf{0}, I)$. The matrices G_1 to G_4 are noise gain matrices [13]. The elements of G_1 and G_2 are determined by the inherent stochastic error characteristics associated with the IMU measurements, while G_3 and G_4 determine the driving noises of the bias

instability terms for the gyroscope and accelerometer, respectively.

$$M_\omega = \begin{bmatrix} \frac{1}{T_{\omega_1}} & 0 & 0 \\ 0 & \frac{1}{T_{\omega_2}} & 0 \\ 0 & 0 & \frac{1}{T_{\omega_3}} \end{bmatrix}, \quad M_f = \begin{bmatrix} \frac{1}{T_{f_1}} & 0 & 0 \\ 0 & \frac{1}{T_{f_2}} & 0 \\ 0 & 0 & \frac{1}{T_{f_3}} \end{bmatrix} \quad (2.88)$$

2.10. Simplified Error Analysis

Over short and mid-term durations, the vertical and horizontal channels of the errors decouple, allowing for independent analysis of each channel [3]. In Fig. 2.7, a platform moving in the north direction is depicted on the X-Z plane. The solutions to the simplified lateral and vertical error dynamics are derived in the following subsections.

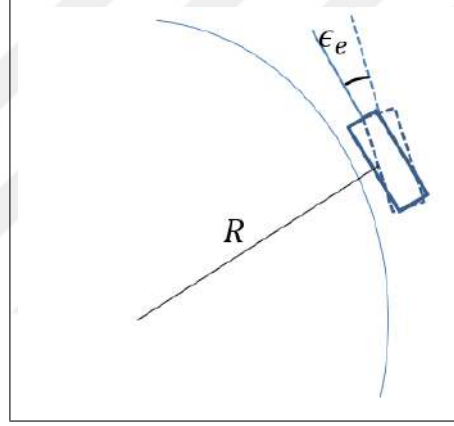


Figure 2.7: X-Z Plane.

2.10.1. Simplified Lateral Error Dynamics

In simplified error analysis to comprehend the deterministic nature of error dynamics, biases, δf_n and $\delta \omega_e$, are treated as constant rather than stochastic processes. Furthermore, effects on the north velocity error from sources other than pitch error are disregarded in (2.78). Under these assumptions, the dynamics of the north velocity error (δv_e) and the east tilt error (ϵ_e) can be described as presented in (2.89).

$$\begin{bmatrix} \dot{\delta v_e} \\ \dot{\epsilon_e} \end{bmatrix} = \begin{bmatrix} 0 & g \\ -\frac{1}{R} & 0 \end{bmatrix} \begin{bmatrix} \delta v_e \\ \epsilon_e \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta f_n \\ \delta \omega_e \end{bmatrix} \quad (2.89)$$

where g is the gravitational acceleration and R is the mean radius of curvature. The equation above takes the form of the following linear dynamics equation:

$$\dot{\mathbf{X}} = A\mathbf{X} + B\mathbf{u} \quad (2.90)$$

The solution of (2.90) is given by (2.91) [14]

$$\mathbf{X}(t) = e^{A(t-t_0)}\mathbf{X}(t_0) + \int_{t_0}^t e^{A(t-\tau)}B(\tau)\mathbf{u}(\tau)d\tau \quad (2.91)$$

Given the initial condition $t_0 = 0$, the formulation e^{At} can be determined in the subsequent formula.

$$e^{At} = Ve^{Dt}V^{-1} \quad (2.92)$$

where V is the matrix composed of the eigenvectors of A , and D is the diagonal matrix composed of the eigenvalues of A . In the case of A is set as given in (2.89), V and D matrices are as follows.

$$V = \begin{bmatrix} -i\sqrt{Rg} & i\sqrt{Rg} \\ 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} i\sqrt{\frac{g}{R}} & 0 \\ 0 & -i\sqrt{\frac{g}{R}} \end{bmatrix} \quad (2.93)$$

Substituting V and D in (2.92) into (2.92):

$$\begin{aligned} e^{At} &= \begin{bmatrix} -i\sqrt{Rg} & i\sqrt{Rg} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} e^{i\sqrt{\frac{g}{R}}t} & 0 \\ 0 & e^{-i\sqrt{\frac{g}{R}}t} \end{bmatrix} \begin{bmatrix} -\frac{1}{2i\sqrt{Rg}} & \frac{1}{2} \\ \frac{1}{2i\sqrt{Rg}} & \frac{1}{2} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\omega_s t) & \sqrt{Rg} \sin(\omega_s t) \\ -\frac{1}{\sqrt{Rg}} \sin(\omega_s t) & \cos(\omega_s t) \end{bmatrix} \end{aligned} \quad (2.94)$$

where, ω_s is known as the Shuler Frequency:

$$\omega_s = \sqrt{\frac{g}{R}} \quad (2.95)$$

By substituting (2.94) into (2.91), the solution for the dynamic system presented in (2.89) can be determined as follows:

$$\begin{aligned} \begin{bmatrix} \delta v_n(t) \\ \epsilon_e(t) \end{bmatrix} &= \begin{bmatrix} \cos(\omega_s t) & \sqrt{Rg} \sin(\omega_s t) \\ -\frac{1}{\sqrt{Rg}} \sin(\omega_s t) & \cos(\omega_s t) \end{bmatrix} \begin{bmatrix} \delta v_n(0) \\ \epsilon_e(0) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{\omega_s} \sin(\omega_s t) & -R(1 - \cos(\omega_s t)) \\ -\frac{1}{g}(1 - \cos(\omega_s t)) & \frac{1}{\omega_s} \sin(\omega_s t) \end{bmatrix} \begin{bmatrix} \delta f_n \\ \delta \omega_e \end{bmatrix} \end{aligned} \quad (2.96)$$

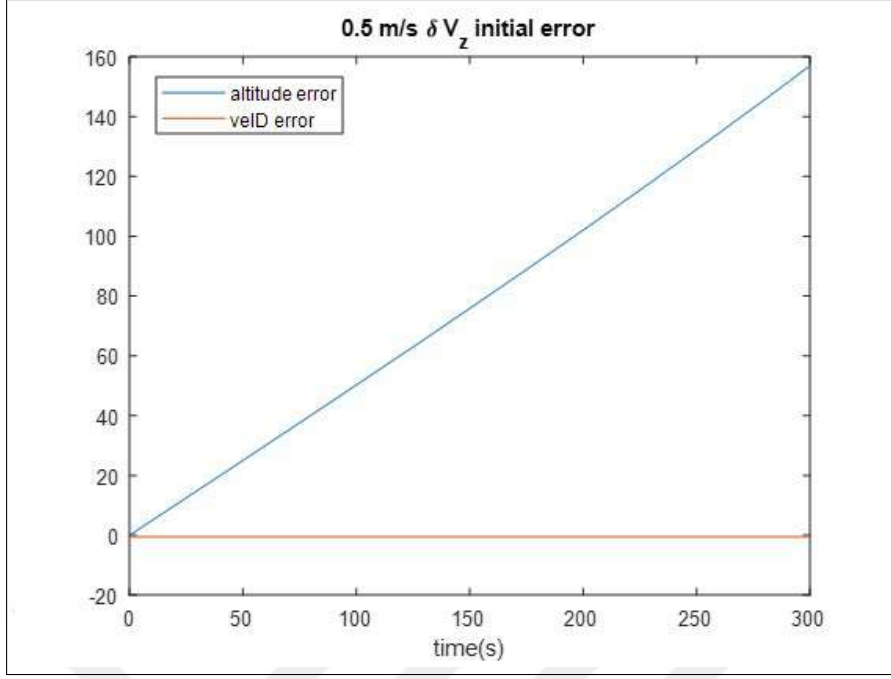


Figure 2.8: Altitude and vertical velocity error graph.

In this simplified model, when we specifically consider the velocity error, it can be articulated as shown in (2.97).

$$\delta v_n(t) = \cos(\omega_s t) \delta v_n(0) + \sqrt{Rg} \sin(\omega_s t) \epsilon_e(0) + \frac{1}{\omega_s} \sin(\omega_s t) \delta f_n + R(\cos(\omega_s t) - 1) \delta \omega_e \quad (2.97)$$

The northing position error equation $\delta p_n(t)$ is obtained by taking the integral of (2.97).

$$\begin{aligned} \delta p_n(t) = & \frac{\sin(\omega_s t)}{\omega_s} \delta v_n(0) + R(1 - \cos(\omega_s t)) \epsilon_e(0) \\ & + \frac{1}{\omega_s^2} (1 - \cos(\omega_s t)) \delta f_n + R \left(\frac{\sin(\omega_s t)}{\omega_s} - t \right) \delta \omega_e \end{aligned} \quad (2.98)$$

The simplified lateral error model reveals that initial velocity, tilt, and accelerometer bias errors induce position error oscillations at the Schuler frequency. For instance, a 0.1 milli-G accelerometer bias results in a position error oscillation of approximately 1.2 km peak to peak. On the other hand, gyroscope bias leads to a linear growth in the position error.

2.10.2. Simplified Vertical Error Dynamics

The gravitational force stabilizes lateral errors, as discussed in the previous subsection. However, it has a destabilizing effect on the vertical channel, as demonstrated below. The altitude error (δh) and vertical velocity error (δv) dynamics can be expressed as demonstrated in (2.99), (2.100) respectively.

$$\delta \dot{h} = -\delta v_d \quad (2.99)$$

$$\delta \dot{v} = \frac{v_n^2 + v_e^2}{R^2} \delta h + \delta g_z \quad (2.100)$$

by substituting δg_z from (2.71) into (2.100) and ignoring the $\frac{v_n^2 + v_e^2}{R^2}$ term, both (2.99) and (2.100) can be expressed in the state-space form as shown in (2.101).

$$\begin{bmatrix} \delta \dot{h} \\ \delta \dot{v}_d \end{bmatrix} \approx \begin{bmatrix} 0 & -1 \\ -\frac{2g}{R} & 0 \end{bmatrix} \begin{bmatrix} \delta h \\ \delta v_d \end{bmatrix} \quad (2.101)$$

The eigenvalues of the state transition matrix in (2.101) are $\pm \sqrt{\frac{2g}{R}}$. Due to the negative eigenvalue, the vertical channel is unstable, and errors grow exponentially if not stabilized by altitude information obtained from an external device, such as a pressure altimeter [3]. In Fig. 2.8, the altitude and vertical velocity errors with respect to time are illustrated for the case with a 0.5 m/s initial downward velocity error.

3. TRAJECTORY AND IMU DATA GENERATION

Testing and validation are critical in the design process of an INS. Generating synthetic data in a simulation environment allows for a quick assessment of algorithm performance, eliminating the time-consuming process of collecting data from a real-time system. Numerous studies in the literature address synthetic IMU data generation and trajectory shaping. Costley *et al.* [15] proposes a technique to produce position, attitude, and velocity states for an aircraft based on a smoothed trajectory derived from waypoints, considering system limitations. Additionally, it employs the trajectory and curvilinear motion theory to simulate accelerometer and gyro measurements. Giroux *et al.* [16] introduce a modular, Simulink-based INS simulator for rapid prototyping and real-time testing. The simulator's validation and performance are evaluated, suggesting its utility in designing inertial navigation algorithms and confirming its effectiveness compared to existing integration schemes. Similarly, Gonzales [17] introduces NaveGo, an open-source simulation tool for low-cost integrated navigation systems. In [18], study introduces an improved IMU data generator for Strapdown inertial navigation simulations, significantly reducing discretization errors.

3.1. Trajectory Generation

Trajectory generation involves creating the ground truth kinematic history of the platform to which the INS is attached. This ground truth kinematic history must adhere to the platform's dynamic limitations and be meaningful in terms of maneuvering dynamics. In this thesis, three distinct motion profiles are defined: straight, rotate, and coordinated turn. In the straight mode, the velocity and attitude profile remain constant over time, but the velocity can increase, decrease, or remain stationary within the profile.

Table 3.1: Straight motion segment.

Velocity Profile	Attitude Profile
$\dot{v}^b = 0, < 0, > 0$	$\dot{r}, \dot{p}, \dot{y} = 0$

In the rotate mode, the body velocity remains constant, while one of the Euler angle

rates is non-zero but constant.

Table 3.2: Rotate motion segment.

Velocity Profile	Attitude Profile
$\dot{v}^b = 0$	$\dot{r}$ or $\dot{p}$ or $\dot{y} \neq 0$

In the coordinated turn maneuver, we aim to simulate more realistic aircraft behaviors. In this move, the aircraft adjusts its roll at a specific rate until it achieves a particular g-force. Simultaneously, its yaw angle changes in proportion to the present roll angle. In (3.1) and (3.2) load factor and the yaw rate expression are given.

$$n = \frac{1}{\cos(r)} \quad (3.1)$$

$$\dot{y} = \frac{g}{v} \sin(r) \cos(r) \quad (3.2)$$

Where n represents the g-force (also known as the load factor), r is the roll angle, and y is the yaw angle. To generate the attitude history for the coordinated turn maneuver, the rate vector representing the rate of the body frame relative to the geodetic frame (ω_{nb}^n) can be computed using the following equation. In this equation, $c(\cdot)$ and $s(\cdot)$ denote the cosine and sine functions, respectively, and r , p , and y represent the roll, pitch, and yaw angles, respectively.

$$\begin{pmatrix} \omega_{nb}^n(x) \\ \omega_{nb}^n(y) \\ \omega_{nb}^n(z) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dot{y} \end{pmatrix} + \begin{pmatrix} c(y) & -s(y) & 0 \\ s(y) & c(y) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ \dot{p} \\ 0 \end{pmatrix} + \begin{pmatrix} c(y) & -s(y) & 0 \\ s(y) & c(y) & 0 \\ 0 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} c(p) & 0 & s(p) \\ 0 & 1 & 0 \\ -s(p) & 0 & c(p) \end{pmatrix} \begin{pmatrix} \dot{r} \\ 0 \\ 0 \end{pmatrix} \right) \quad (3.3)$$

Once the instantaneous body rates are calculated, C_b^n matrix can be updated as follows,

$$C_b^m(k+1) \approx C_b^m(k) \exp(\Omega_{nb}^b \Delta T) \quad (3.4)$$

where,

$$\Omega_{nb}^b = C_n^b \Omega_{nb}^n C_b^n \quad (3.5)$$

3.1.1. Polynomial Fitting for a Dynamic Segment

A typical flight trajectory comprises a sequence of various motion segments. To ensure smooth error-free IMU data, both the velocity and attitude profiles must be continuous and differentiable. To achieve this, a fifth-order polynomial is fitted to the non-stationary motion segments. Each non-stationary segment, in terms of either velocity or attitude, has initial and final values, and the intermediate values are determined using interpolation.

Let us assume that the velocity resolved in body axes is a function of time t within the interval $\in [t_0, t_1]$ and is represented as a fifth-order polynomial as given in (3.6)

$$v(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (3.6)$$

Taking the first and second derivative of (3.6),

$$\dot{v}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \quad (3.7)$$

$$\ddot{v}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3 \quad (3.8)$$

The fifth-order selection is deliberate. To determine the coefficients a_0 to a_5 , at least 6 different linearly independent equations are required. Assuming $\dot{v}(t)$ and $\ddot{v}(t)$ are zero at the boundary points ensures smoothness in rate and acceleration at the segment junctions, yielding four additional equations. Therefore, knowing only the boundary values of the polynomial is sufficient to determine the coefficients.

$$\begin{bmatrix} v(t_0) \\ v(t_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (3.9)$$

So, it is straightforward to obtain coefficients using (3.10).

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \end{bmatrix}^{-1} \begin{bmatrix} v(t_0) \\ v(t_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.10)$$

For illustration, in Fig. 3.1, the graph of $v(t)$ is shown where the initial velocity at $t = 0$ is '0' and the final velocity is 200 m/s at $t = 5$. The same method can be utilized for the Euler angles, representing the vehicle's attitude changes during a maneuver.

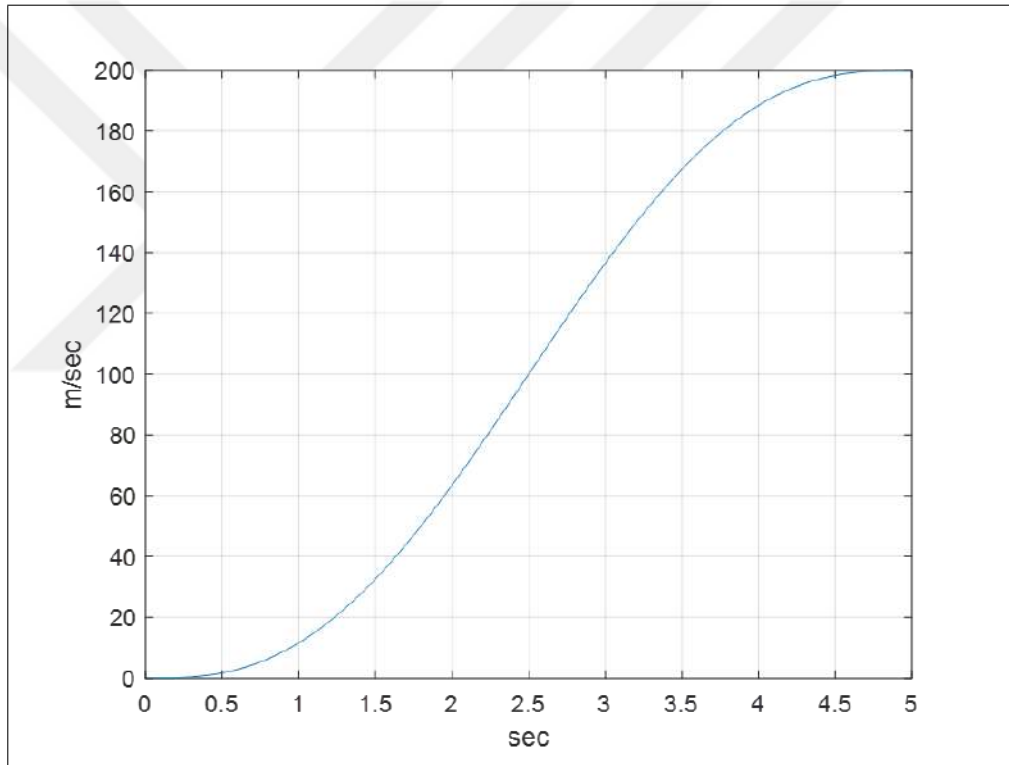


Figure 3.1: Velocity profile.

3.2. IMU Data Generation via Inverse Kinematics

Once a trajectory is set (history of C_b^n , v^b), the inverse procedure of navigation mechanization equations can be applied to generate groundtruth IMU specific force and turn rate measurements. While this 'inverting mechanization' approach might not provide

genuine IMU measurements due to its inherent connection to the discrete implementation method of the navigation mechanization equations and chosen gravity models, it is sufficiently accurate for use in offline simulations, especially when the primary objective is to evaluate hybrid navigation performance, such as INS-GNSS integration.

$$C_{b(k)}^{b(k-1)} = \left(C_{n(k-1)}^{n(k)} \right)^T \left(C_{b(k-1)}^{n(k-1)} \right)^T C_{b(k)}^{n(k)} \quad (3.11)$$

$C_{b(k+1)}^{b(k)}$ then can be converted to a rotation vector, which can be approximated as $\omega_{ib}^b \Delta T$ assuming that ω_{ib}^b is constant between t_k and t_{k+1} . Specific force, f^b , can be evaluated via following equation;

$$f^b = (C_b^n)^T \left[\frac{(v^n(k) - v^n(k-1))}{\Delta T} + (2\omega_{ie}^n + \omega_{en}^n(k)) \times v^n(k) - v^n(k) \right] \quad (3.12)$$

2D position, velocity, specific force, and body rate graphs for a sample trajectory where the aircraft follows a straight line for 30 seconds, then undertakes two consecutive 2g turns for 10 seconds each, and resumes straight flight for another 30 seconds are presented in Fig. 3.2 through Fig. 3.5

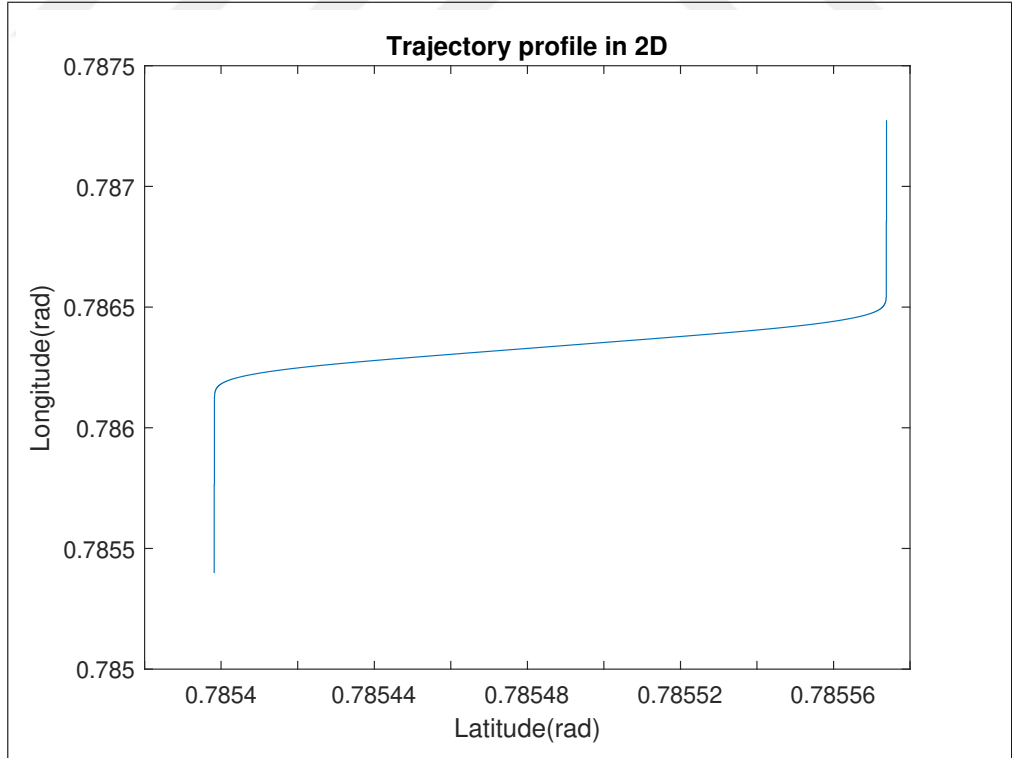


Figure 3.2: Latitude-longitude graph.

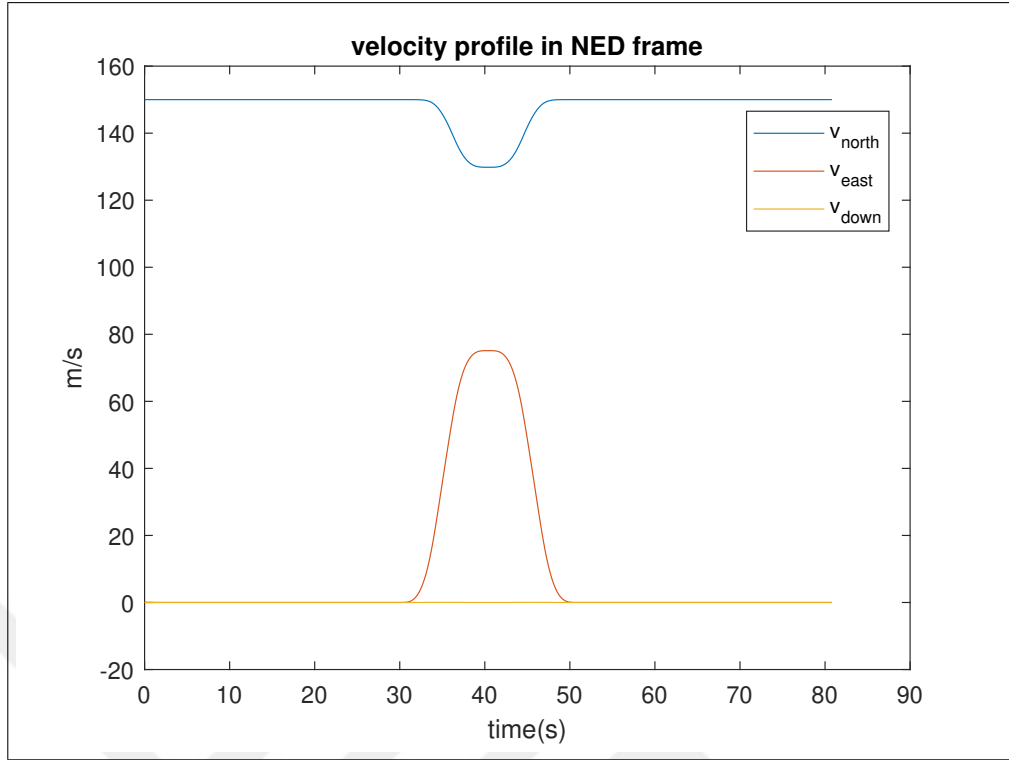


Figure 3.3: Velocity graph.

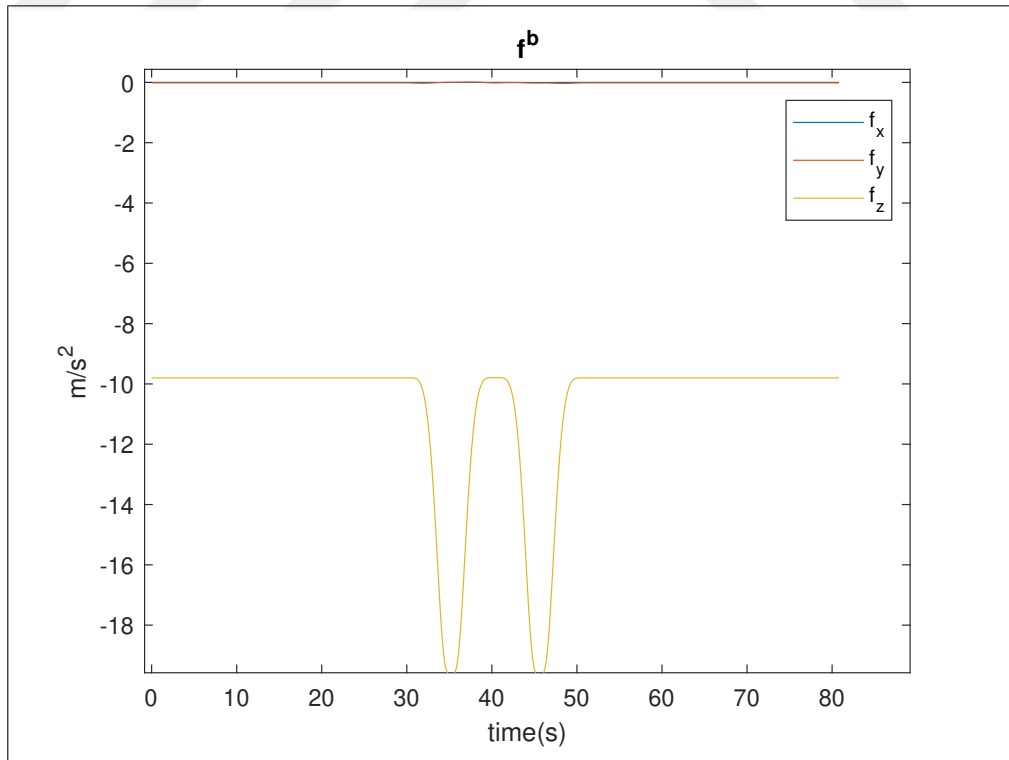


Figure 3.4: Specific force graph.

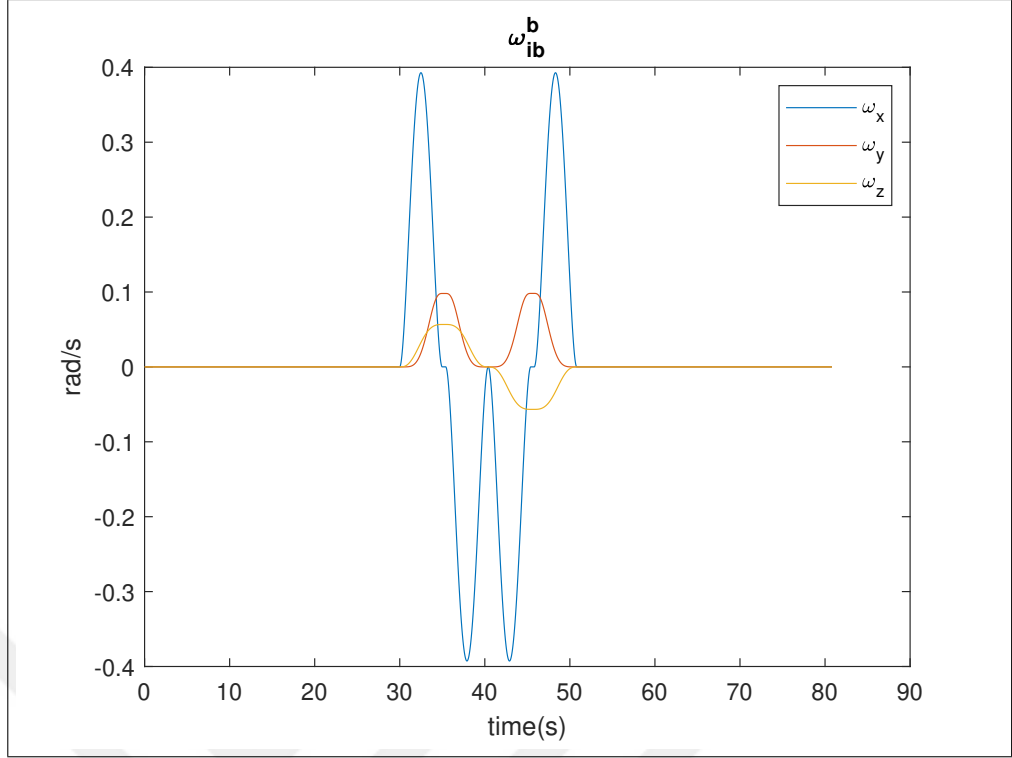


Figure 3.5: Body rates.

3.3. Deterministic and Stochastic IMU Errors

In the previous subsection, we discussed the method of obtaining ground truth IMU measurements by inverting the navigation mechanization equations. However, the two primary components of an IMU, gyroscopes and accelerometers, are subject to several sources of error. If we disregard initialization errors, these IMU errors become the primary reason for the gradual drift of the standalone PVA solution of an INS from the actual PVA [19], [20]. Therefore, it is crucial to model or generate these dominant errors for a realistic simulation. Regardless of the IMU component, these errors can be categorized into two main types: deterministic and stochastic errors.

3.3.1. Bias Error

Bias can be divided into two components: static and dynamic [21]. The static bias, also referred to as fixed, turn-on or repeatability bias, is constant during an IMU's operation but may differ across multiple runs. In contrast, the dynamic bias varies within short time frames, like minutes or hours. Fixed accelerometer biases are typically expressed

in milli-g (mg) or micro-g (μg), with 1g equivalent to 9.80665 m/s^2 . Fixed gyro biases are commonly denoted in degrees per hour ($^\circ/\text{hr}$), where $1^\circ/\text{hr}$ equals $4.85 \times 10^{-6} \text{ rad/s}$. Only lower-quality gyros use degrees per second as a measure.

3.3.1.1. Bias Repeatability Implementation

In simulation, the implementation of bias repeatability (fixed bias) error is direct and uncomplicated. Datasheets typically provide the standard deviation of turn-on biases for both accelerometers and gyroscopes. In a single simulation run, the bias value for either the accelerometer or gyroscope is randomly determined based on the 1 std value of the corresponding bias specification. These constant values are then added to the true rate and specific force values throughout the simulation.

3.3.1.2. Bias Instability Implementation

A bias instability error is defined as a bias error that changes over time. To model this error, various models have been suggested in the literature such as first-order Gauss-Markov process [22], [23] and higher-order autoregressive models [24]. In this thesis, our primary attention is directed towards the first order Gauss-Markov based modeling and the details of its implementation.

$$\dot{\epsilon}_{bist} = -\frac{1}{T_B}\epsilon_{bist} + \omega_B(t) \quad (3.13)$$

In equation (3.13), ϵ_{bist} represents the bias instability error, T_B denotes the correlation time of the process, and $\omega_B(t)$ is the white driving noise, with its power spectrum denoted by S_B as given by (3.14), where the B value can be estimated from the Allan variance graph. [25].

$$S_B = \frac{2B^2 \ln(2)}{\pi(0.4365)^2 T_B} \quad (3.14)$$

Once the values of T_B and S_B are determined, it becomes essential to derive the discrete version of the process presented in (3.13) to implement the bias instability error in a simulation environment. The discretized version of (3.13) is given in [3] as follows:

$$\epsilon_{bist}(k) = e^{-\frac{\Delta t}{T_B}} \epsilon_{bist}(k-1) + \left\{ \sqrt{\frac{S_B T_B}{2}} (1 - e^{-2\frac{\Delta t}{T_B}}) \right\} u(k) \quad (3.15)$$

Where $u(k)$ denotes the white noise sequence with a variance of 1, and Δt refers to the sampling period.

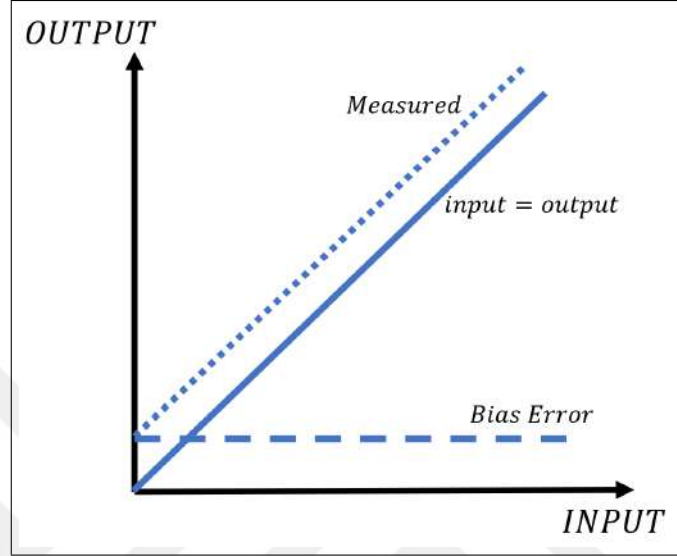


Figure 3.6: Fixed bias error.

3.3.2. Scale Factor Error

The scale factor error in an instrument represents a deviation from a one-to-one input-output relationship after unit conversion by the IMU. This means that the accelerometer's output error, resulting from the scale factor error, is tied to the true specific force on its sensitive axis. Similarly, the gyro's output error related to the scale factor error mirrors the true angular rate about its sensitive axis.

Similar to the bias repeatability error, the scale factor error is also modeled as a random constant. This implies that its value is randomly chosen based on a one-standard deviation specification and remains constant throughout the simulation. Once its value is determined, the erroneous measurement $\tilde{\epsilon}$ (rate for gyro or specific force for accelerom-

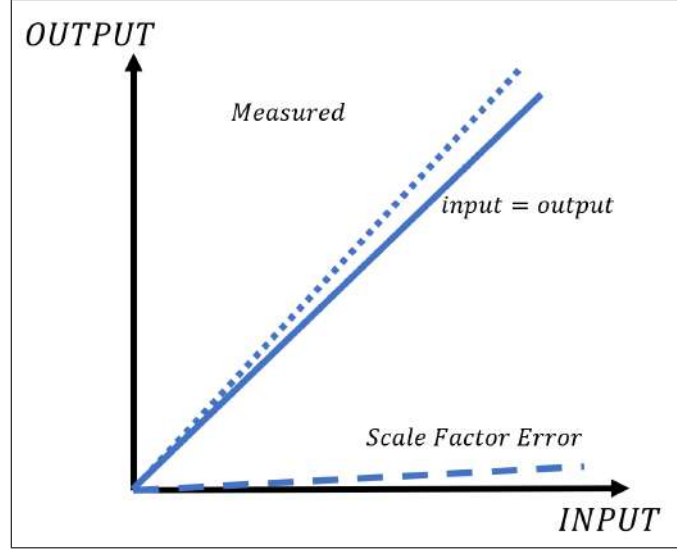


Figure 3.7: Scale factor error.

eter) is obtained from the true value ϵ as presented in (3.16)

$$\tilde{\epsilon} = (I + S)\epsilon, \quad S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \quad (3.16)$$

where s_x , s_y , s_z are the realized scale factor errors for each IMU axes x, y, and z, respectively.

3.3.3. Misalignment Error

Misalignment errors in IMUs result from misalignments between the sensitive axes of inertial sensors and the orthogonal body frame axes, often due to manufacturing constraints. They cause accelerometers to detect forces and gyros to detect angular rates not in line with their primary axes. Misalignment error is also modeled as a random constant. The relationship between an erroneous measurement $\tilde{\epsilon}$ and the true value ϵ is given in (3.17), where the off-diagonal elements in M indicate misalignment angles.

$$\tilde{\epsilon} = (I + M)\epsilon, \quad M = \begin{pmatrix} 0 & m_{xy} & m_{xz} \\ m_{yx} & 0 & m_{yz} \\ m_{zx} & m_{zy} & 0 \end{pmatrix} \quad (3.17)$$

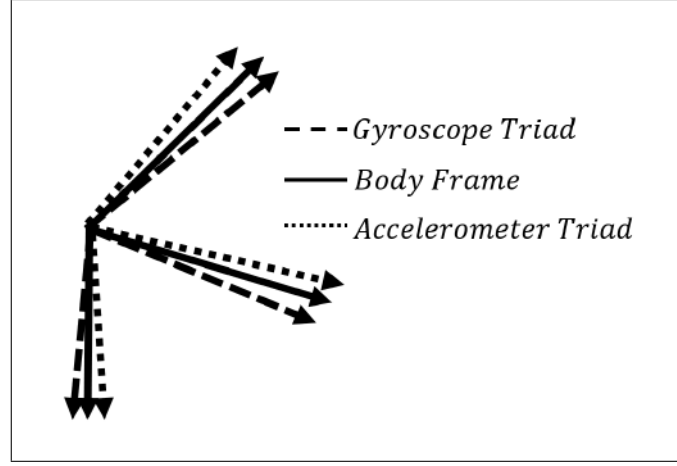


Figure 3.8: Misalignment error.

3.3.4. Angle & Velocity Random Walk Errors

Angle random walk (ARW) and velocity random walk (VRW) errors are characterized as independent and identically distributed random noise on gyroscope and accelerometer measurements, respectively. These errors manifest as random walks on angle and velocity [26], [27]. The unit of VRW is typically represented as $m/s/\sqrt{Hour}$ or $mg/\sqrt{Hz}$, whereas the unit of ARW is commonly given as $^{\circ}/\sqrt{Hour}$. To implement ARW or VRW in a simulation, one needs to evaluate the standard deviation specified as σ_d in (3.18) where $u(k)$ is the standard white noise sequence.

$$\tilde{\epsilon}(k) = \epsilon(k) + \sigma_d u(k) \quad (3.18)$$

where, ϵ is the true value (rate for gyroscope or specific force for accelerometer) and $\tilde{\epsilon}$ represents the erroneous value obtained by adding ARW or VRW on the true value. If ϵ is rate(rad/s), then σ_d can be calculated using (3.19) assuming the ARW error specification is $X^{\circ}/\sqrt{Hour}$.

$$\sigma_d = \frac{X(\pi/180)}{60\sqrt{\Delta t}} \quad (3.19)$$

If ϵ is a specific force, then σ_d can be evaluated from the given $X m/s/\sqrt{Hour}$ VRW specification as given in (3.20)

$$\sigma_d = \frac{X}{60\sqrt{\Delta t}} \quad (3.20)$$

where Δt is the sampling period of IMU in (3.20), (3.19).

3.3.5. Other Error Types

Apart from the errors discussed above, there are other types of IMU errors, including g-sensitive, square-sensitive, scale factor nonlinearity, rate, and accelerometer random walk errors, which are not covered in this thesis. Interested readers can refer to [2] and [21].

3.4. Data generation for Master-Slave Platforms

In some applications, multiple INSs are present on a platform. For instance, munitions attached under an aircraft pylon have a tactical grade INS that requires initialization from the PVA data provided by the main INS of the aircraft. IMU data for a slave system can be generated if the history of the lever-arm vector, which represents the vector from the master INS origin to the slave INS origin, and the orientation between the slave and master INS are known. As discussed in Section 3.2, synthetic IMU measurements require histories of both attitude and velocity information. In Fig 3.9, a configuration of a master (aircraft) INS and a slave (munition) INS is depicted. The vector $r_{m|s}^m$ is referred to as the lever-arm (or moment-arm), where m represents the master INS frame, and s represents the slave INS frame. The velocity of the slave system can be derived from the kinematic information of the master frame combined with the lever arm vector. $r_{e|s}$ vector can be expressed as given in (3.21).

$$\mathbf{r}_{e|s} = \mathbf{r}_{e|m} + \mathbf{r}_{m|s} \quad (3.21)$$

taking the time derivative of (3.21) with respect to e frame,

$$\left. \frac{d}{dt} \mathbf{r}_{e|s} \right|_e = \left. \frac{d}{dt} \mathbf{r}_{e|m} \right|_e + \left. \frac{d}{dt} \mathbf{r}_{m|s} \right|_e \quad (3.22)$$

$$\mathbf{v}_s \triangleq \left. \frac{d}{dt} \mathbf{r}_{e|s} \right|_e \quad \mathbf{v}_m \triangleq \left. \frac{d}{dt} \mathbf{r}_{e|m} \right|_e \quad (3.23)$$

where $\mathbf{v}_s$ and $\mathbf{v}_m$ represent the ground velocities of the slave and master systems, respectively, as outlined in (3.23). By employing the transport theorem given in (2.20), the expression for $\left. \frac{d}{dt} \mathbf{r}_{m|s} \right|_e$ can be written as:

$$\left. \frac{d}{dt} \mathbf{r}_{m|s} \right|_e = \left. \frac{d}{dt} \mathbf{r}_{m|s} \right|_m + \boldsymbol{\omega}_{em} \times \mathbf{r}_{m|s} \quad (3.24)$$

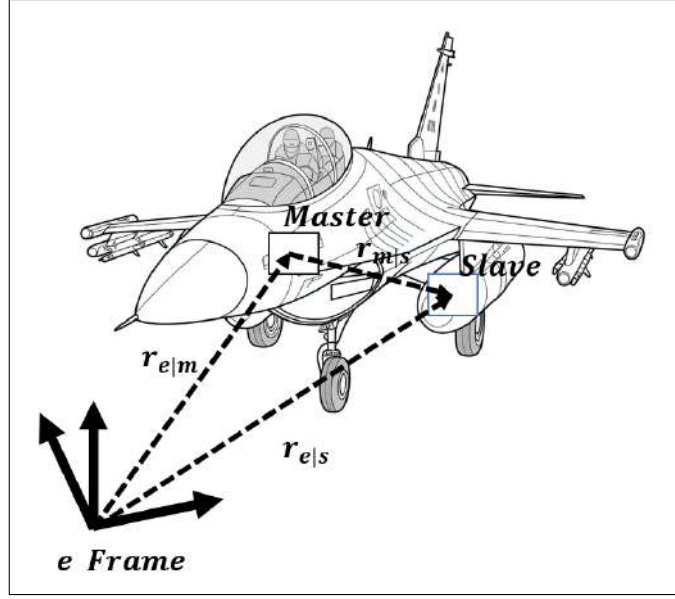


Figure 3.9: Master-Slave demonstration.

where $\left. \frac{d}{dt} \mathbf{r}_{m|s} \right|_m$ represents the time derivative of the lever-arm vector due to flexure, symbolized as $\dot{\mathbf{r}}_{m|s}$, and $\boldsymbol{\omega}_{em} = -\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{im}$. Substituting these values and inserting (3.24) into (3.22), we get:

$$\mathbf{v}_s = \mathbf{v}_m + \dot{\mathbf{r}}_{m|s} + (-\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{im}) \times \mathbf{r}_{m|s} \quad (3.25)$$

Attitude of the slave system can be obtained as demonstrated in (3.26).

$$\mathbf{C}_s^n = \mathbf{C}_m^m \mathbf{C}_s^m \quad (3.26)$$

where $\mathbf{C}_s^m$ represents the DCM matrix derived from the boresight angles between the master frame and the slave frame.

3.5. Data generation for Gimbaled Platforms

In forward-looking infrared (FLIR) applications, the slave INS is mounted on a plate within the gimbaled structure. This structure rotates in response to the operator's commands. Generating slave IMU data for these applications can aid in analyzing the sight-line estimation problem within a simulated environment.

Generating IMU data for a FLIR system necessitates a slight modification to the master-slave data generation procedure discussed in the previous subsection. Unlike the fixed

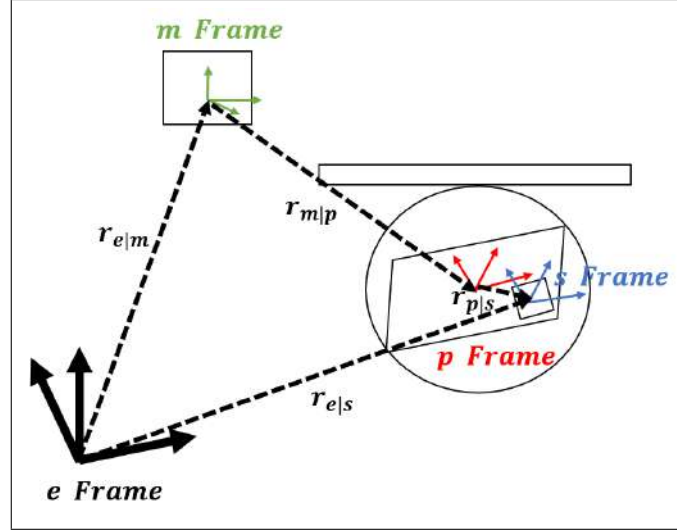


Figure 3.10: Master-slave (on a gimbal plate) demonstration.

lever arm ($\mathbf{r}_{m|s}$) and boresight (C_s^m) between the IMU, the FLIR systems require these values to vary due to gimbal movement.

In the subsequent section, the attitude and velocity equations for the slave INS mounted on the plate of a FLIR system are derived, as illustrated in Fig.3.10. Here, the m , e , p , and s frames represent the Master INS frame, ECEF frame, Plate frame, and Slave frame, respectively.

$$\mathbf{r}_{e|s} = \mathbf{r}_{e|m} + \mathbf{r}_{m|p} + \mathbf{r}_{p|s} \quad (3.27)$$

taking the time derivative of (3.27) with respect to earth frame:

$$\mathbf{v}_s = \mathbf{v}_m + \left. \frac{d}{dt} \mathbf{r}_{m|p} \right|_e + \left. \frac{d}{dt} \mathbf{r}_{p|s} \right|_e \quad (3.28)$$

Where $\mathbf{v}_s$ and $\mathbf{v}_m$ represent the velocity vectors of the slave and master systems with respect to the ground. Utilizing the transport theorem, the rest of the variables can be expressed as follows:

$$\left. \frac{d}{dt} \mathbf{r}_{m|p} \right|_e = \left. \frac{d}{dt} \mathbf{r}_{m|p} \right|_m + \boldsymbol{\omega}_{em} \times \mathbf{r}_{m|p} \quad (3.29)$$

$$\left. \frac{d}{dt} \mathbf{r}_{p|s} \right|_e = \left. \frac{d}{dt} \mathbf{r}_{p|s} \right|_p + \boldsymbol{\omega}_{ep} \times \mathbf{r}_{p|s} \quad (3.30)$$

Assuming that the lever-arm vector from the master to the plate frame center and from the plate center to the slave INS is constant, equation (3.28) can be rewritten as follows:

$$\mathbf{v}_s = \mathbf{v}_m + \boldsymbol{\omega}_{em} \times \mathbf{r}_{m|p} + \boldsymbol{\omega}_{ep} \times \mathbf{r}_{p|s} \quad (3.31)$$

where

$$\omega_{em} = \omega_{im} - \omega_{ie} \quad (3.32)$$

$$\omega_{ep} = \omega_{em} + \omega_{mp} \quad (3.33)$$

The process of obtaining the slave INS attitude with respect to the navigation frame N is demonstrated below.

$$C_s^m = C_m^m C_p^m C_s^p \quad (3.34)$$

Taking the derivative of equation (3.34) and assuming that the attitude of the slave INS remains constant with respect to the plate frame ($\dot{C}_s^p = 0$),

$$\dot{C}_s^n = (\dot{C}_m^n C_p^m + C_m^n \dot{C}_p^m) C_s^p \quad (3.35)$$

$$= C_m^n \Omega_{nm}^m C_p^m + C_m^n C_p^m \Omega_{mp}^p \quad (3.36)$$

where $\Omega_{mp}^p = S(\omega_{mp}^p)$, where ω_{mp}^p is the rate vector resulting from the rotation of the plate frame with respect to the master INS frame due to the gimbal command. In the subsequent section, a trajectory of a slave system attached to a rotating FLIR system with a speed of 100 rpm is showcased, accompanied by its IMU measurements in the following graphs.

True gyroscope and specific force graphs are given in Fig.3.12 and Fig.3.13 The velocity of the slave INS, resolved in the n frame (NED), is presented in Fig.3.14

3.5.1. Tracking a Point with a FLIR

One of the primary missions of FLIR systems is reconnaissance, wherein the object or target of interest is centered on the FLIR camera while the UAV is in motion. In this subsection, we discuss the details of the slave INS data generation when an imaginary camera, whose x-axis is aligned with the slave INS's x-axis, tracks a point on Earth. The primary objective is to ensure that the line of sight vector, ζ , delineated as the unit vector directed towards the target point from the slave INS, remains consistently aligned with the x-axis of the slave INS frame. It's crucial to note that the camera frame and the

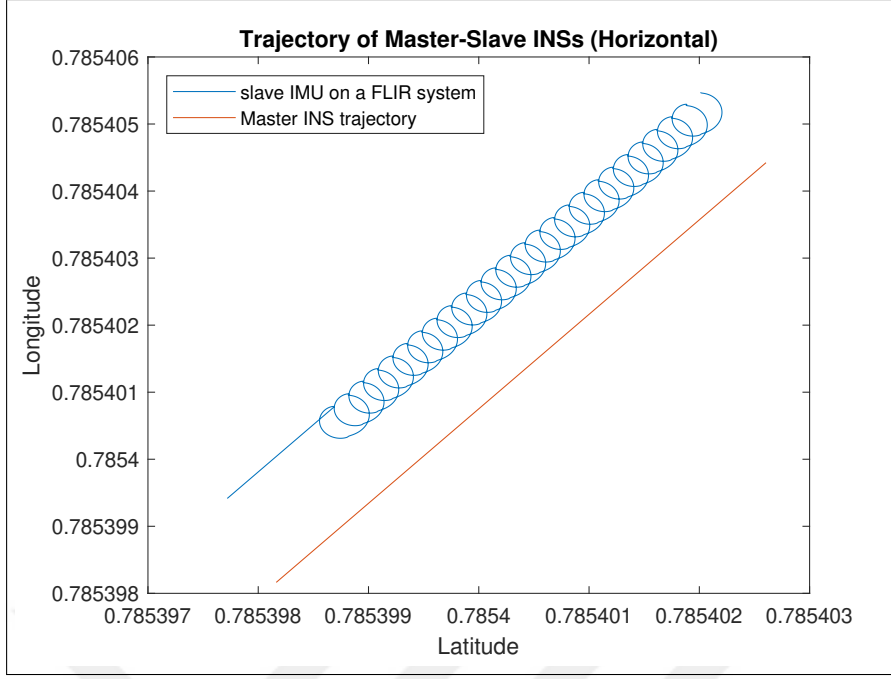


Figure 3.11: Trajectory of a slave INS (on rotating a FLIR).

slave INS frame are considered to be in perfect alignment and their centers overlap.

$$\zeta^s(t) = C_n^s(t)\zeta^n(t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \forall t \quad (3.37)$$

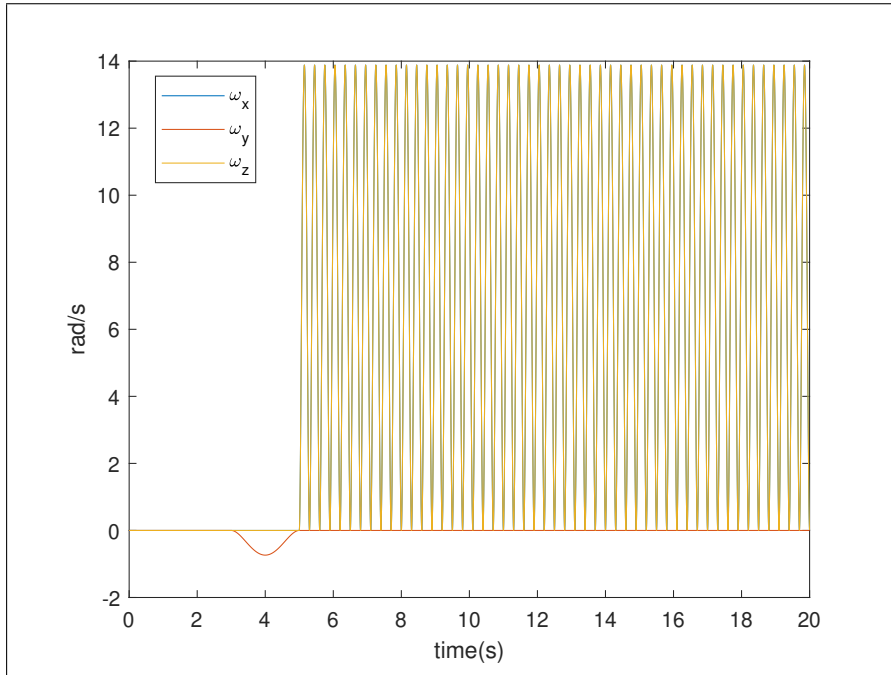


Figure 3.12: Body rates of the slave INS.

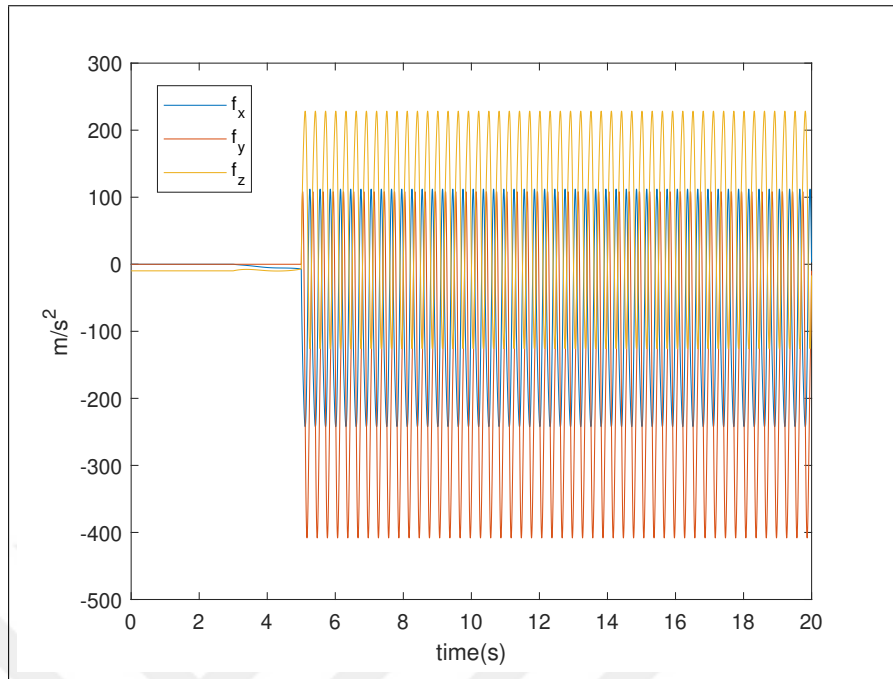


Figure 3.13: Specific forces of the slave INS.

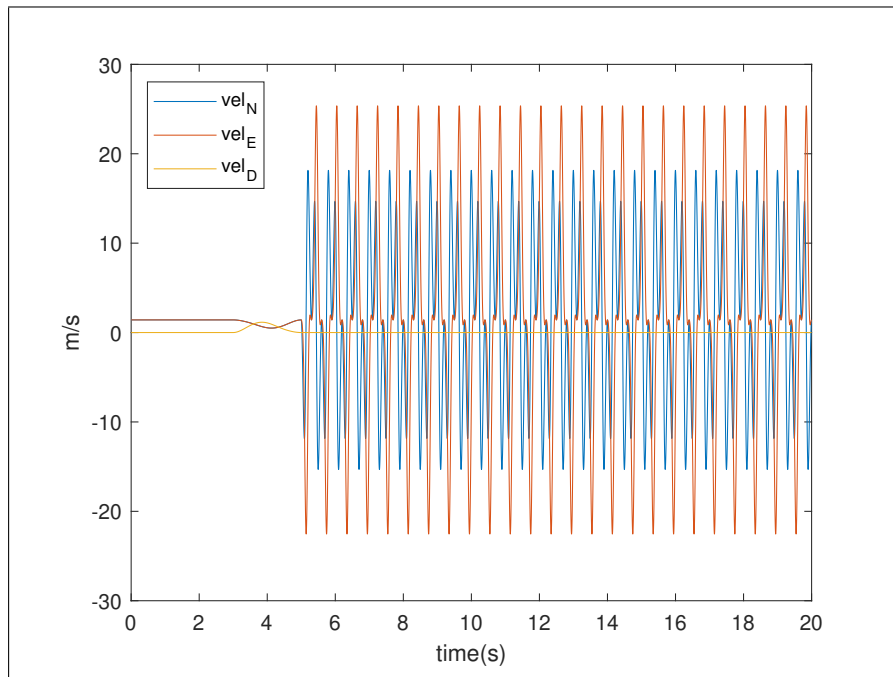


Figure 3.14: Velocity of the slave INS (resolved in NED frame).

where ζ^n can be approximated as

$$\zeta^n \approx \left\| \begin{bmatrix} (l_t - l_S)(R_n + h_S) \\ (L_t - L_S)(R_e + h_S)(\cos(l_S)) \\ -(h_t - h_S) \end{bmatrix} \right\| \quad (3.38)$$

where the symbols l , L , and h represent latitude, longitude, and altitude, respectively. Meanwhile, the subscripts S and t distinguish between the Slave INS and the target. The symbol $\|$ denotes the norm of the vector enclosed within it. C_n^s can be expanded as presented in (3.39)

$$C_n^s = C_p^s C_m^p(r=0, p, y) C_n^m \quad (3.39)$$

Here, $C_m^p(r=0, p, y)$ represents the DCM characterized by two degrees of freedom, specifically the pitch (p) and yaw angles (y), indicating that there is no rotation of the gimbal around the roll axis.

$$C_m^p(r=0, p, y) = \begin{bmatrix} c(p) & 0 & -s(p) \\ 0 & 1 & 0 \\ s(p) & 0 & c(p) \end{bmatrix} \begin{bmatrix} c(y) & s(y) & 0 \\ -s(y) & c(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.40)$$

Substituting the right hand side of equation (3.40) into equation (3.39) and pre-multiplying by C_s^p and post-multiplying by ζ both sides of (3.39), we obtain:

$$\begin{aligned} C_s^p \mathbf{1}_x &= \begin{bmatrix} c(p) & 0 & -s(p) \\ 0 & 1 & 0 \\ s(p) & 0 & c(p) \end{bmatrix} \begin{bmatrix} c(y) & s(y) & 0 \\ -s(y) & c(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \zeta^m \\ &= \begin{bmatrix} c(p)c(y) & c(p)s(y) & -s(p) \\ -s(y) & c(y) & 0 \\ s(p)c(y) & s(p)s(y) & c(p) \end{bmatrix} \zeta^m \end{aligned} \quad (3.41)$$

For every time step in the simulation, the values of r and p that fulfill equation (3.41) must be determined. To compute these gimbal angles, we employed MATLAB's 'fsolve' function. The following illustrations belong to slave INS that tracks a point. In this setup, the master INS advances in the north direction at a speed of 1 m/s. The lever-arm vector, extending from the master to the slave system, is given by $[0, 5, 1]^T m$. Additionally, a stationary target is positioned 15 meters from the FLIR system's starting point, equidistant in both the north and east directions. In Fig 3.15 and Fig 3.16, the

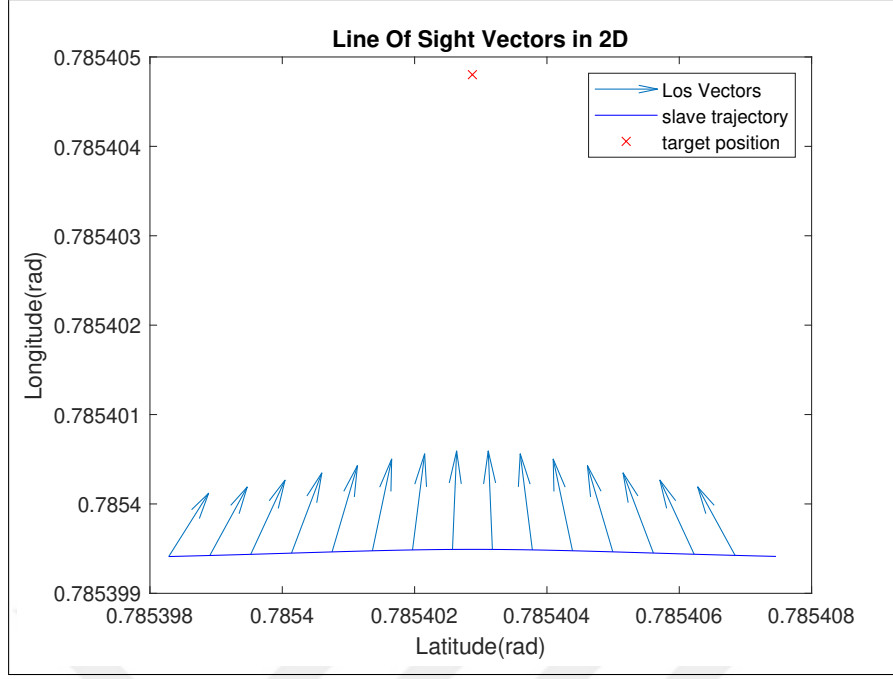


Figure 3.15: LOS vectors from FLIR to a stationary target(2D).

blue vectors depict the line of sight (LOS) vectors over the entire trajectory, with the FLIR system tracking the red 'x' mark shown in the diagrams. These LOS vectors align with the x-axis of the slave frame.

3.6. Vibration

The effect of vibration is unignorable in the design and performance assessment of inertial navigation systems. Vibrational effects can introduce additional errors in IMU readings, which can, in turn, degrade the overall system accuracy. In this part, synthetic vibration generation is discussed.

3.6.1. Power Spectral Density and Fourier Transform Relationship

The Fourier transform (FT) converts a time-domain signal into its corresponding frequency-domain representation. Understanding the FT becomes imperative, especially when examining system responses to different frequency inputs or disturbances. The analysis and synthesis equations of the discrete Fourier transform (DFT) are provided

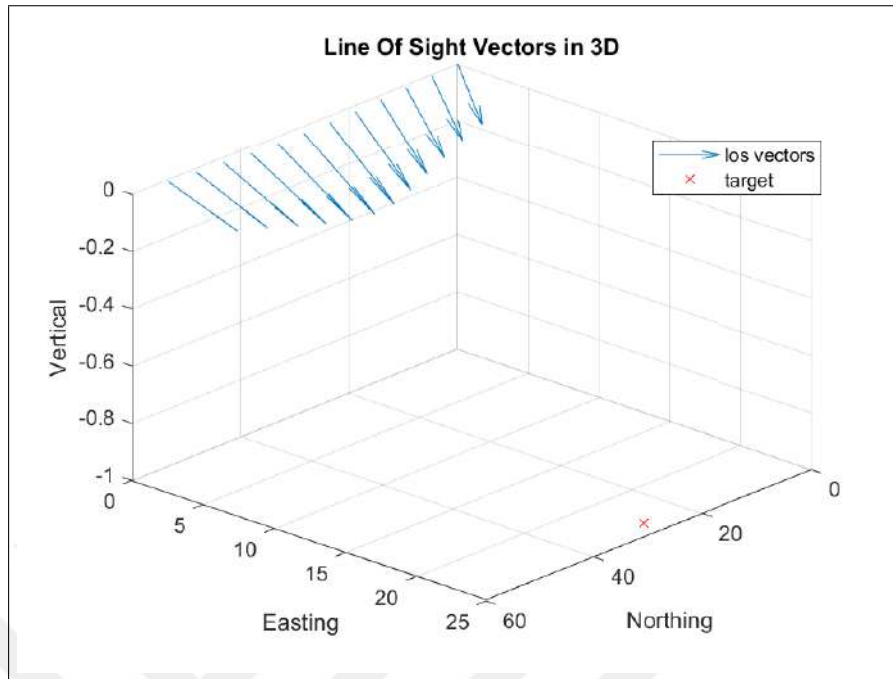


Figure 3.16: LOS vectors from FLIR to a stationary target(3D).

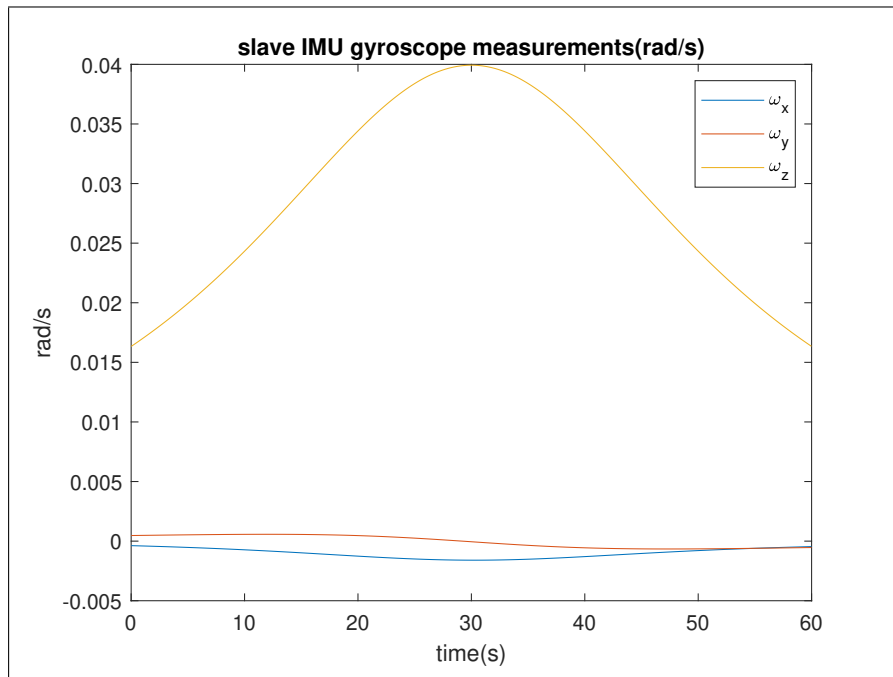


Figure 3.17: Gyroscope output of the slave system.

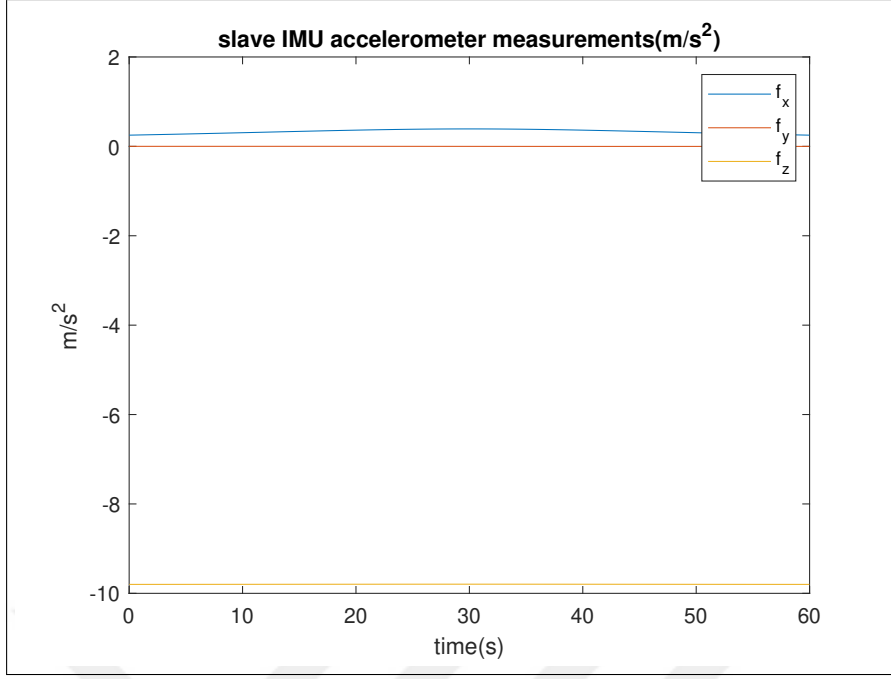


Figure 3.18: Accelerometer output of the slave system.

in equations (3.42) and (3.43), respectively [28].

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn} \quad (3.42)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{+j \frac{2\pi}{N} kn} \quad (3.43)$$

where N is the total number of samples of the discrete signal $x[n]$ and k denotes the frequency bin index whose frequency correspondence can be evaluated as $\omega_k = k \frac{2\pi}{N}$ thus for discrete time signals ω_k takes values between 0 to $(N - 1) \frac{2\pi}{N}$. However, $X[k]$ is symmetric around $\omega_k = \pi$ due to the inherent properties of the DFT. For real-valued signals, only half the DFT spectrum contains unique information. This is often why analysis is focused on the range of k values from 0 to $N/2$, corresponding to the frequency range from 0 to the π . If the sampling frequency, f_s , is known, then the frequency axes of the FT can be expressed in Hertz by multiplying the $\frac{f_s}{2\pi}$ by normalized frequency axis $[0 - \pi]$. FT of the discrete signal $x[k]$, obtained by sampling the continuous version $x(t)$ with the sampling frequency f_s , allows for the analysis of the frequency spectrum $[0, \frac{f_s}{2}]$ Hz. The upper limit value $\frac{f_s}{2}$ is also known as the "Nyquist frequency". Although FT is a powerful way to analyze deterministic type of signals, it may not be adequate for analyzing random vibrations. This inadequacy arises

because the magnitude of the Fourier representation depends on the number of spectral lines. To overcome this limitation, the Power Spectral Density (PSD) is often used for analyzing random vibrations. The PSD provides a more meaningful representation of random vibration by showing how the power (variance) of the vibration is distributed across the frequency spectrum. It normalizes the magnitude of the spectral lines by the frequency resolution, which provides a clearer picture of the energy distribution in the frequency domain irrespective of the number of spectral lines [29]. To establish the relationship between PSD and DFT, the Parseval theorem articulated in (3.44) can be utilized for stationary random signals.

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \quad (3.44)$$

Assuming that in (3.44) $x[n]$ is a random vibration modeled by Gaussian white noise process. Since this process is ergodic in variance, one can express $E[x[k]^2]$ as shown in (3.45).

$$E[x[n]^2] \approx \frac{1}{N} \sum_{n=0}^{N-1} x[n]^2 \quad N \rightarrow \infty \quad (3.45)$$

where $E[x[n]^2]$ denotes the variance, σ^2 , of the process, it can also be determined by computing the area beneath the PSD ($\mathcal{P}[k]$) of the process.

$$E[x[n]^2] = \sigma^2 = \sum_{k=0}^{N-1} \mathcal{P}[k] \Delta f \quad (3.46)$$

where,

$$\Delta f = \frac{f_s}{N} \quad (3.47)$$

In Equation (3.47), Δf denotes the frequency resolution. The left-hand side of (3.44) can be represented as $N E[x[n]^2]$ by employing (3.45). By substituting the right-hand side of (3.46) for $E[x[n]^2]$, the ensuing formulation is acquired:

$$N \sum_{k=0}^{N-1} \mathcal{P}[k] \frac{f_s}{N} = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \quad (3.48)$$

Thus, $\mathcal{P}[k]$ can be expressed as:

$$\mathcal{P}[k] = \frac{|X[k]|^2}{f_s N} \quad (3.49)$$

3.6.2. Synthetic Vibration Generation

Vibration profiles are usually characterized by PSD graph. In Fig 3.19 a random vibration spectrum profile is illustrated. This profile is widely used for manufacturing screening to identify defects in both parts and workmanship [30].

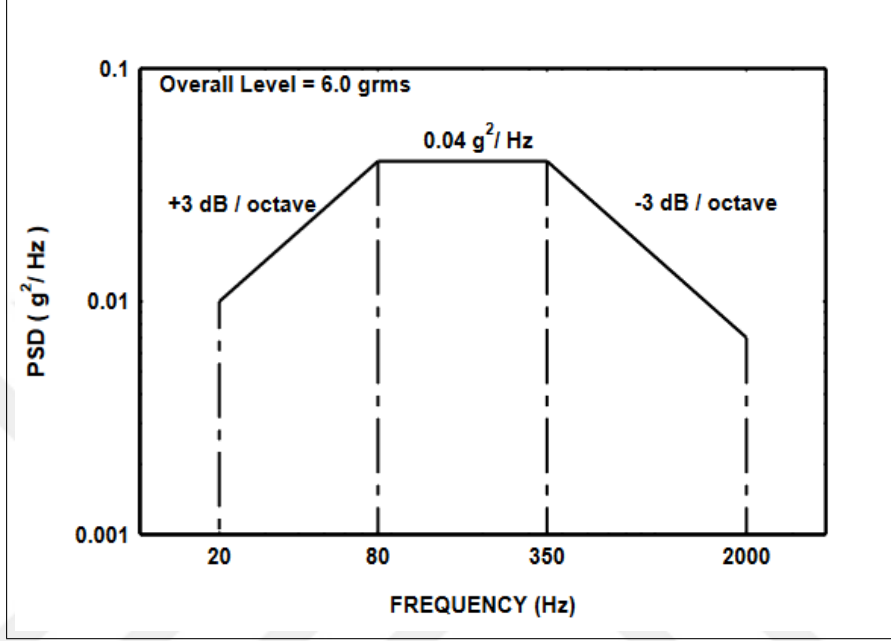


Figure 3.19: Random vibration PSD Spectrum.

One way to generate synthetic vibration data, whose PSD aligns with a given PSD profile, is to produce a transfer function $H_f(j\omega)$ that satisfies the subsequent property.

$$S_w(\omega)|H(j\omega)|^2 = \mathcal{P}(\omega) \quad (3.50)$$

Where the input to the frequency-shaping filter is a white noise process with a variance of unity, we have $S_w(\omega) = 1$. Therefore, the transfer function can be approximately designed using a filter designer so that its magnitude profile mirrors $\sqrt{\mathcal{P}(\omega)}$.

An alternative method for creating synthetic vibration data involves utilizing (3.49), where the magnitude $|X[k]|$ is derived from the PSD profile and the phase is assigned randomly. Subsequently, the inverse Fourier transform, as specified in (3.43), is computed to yield the discrete time sequence of random vibration.

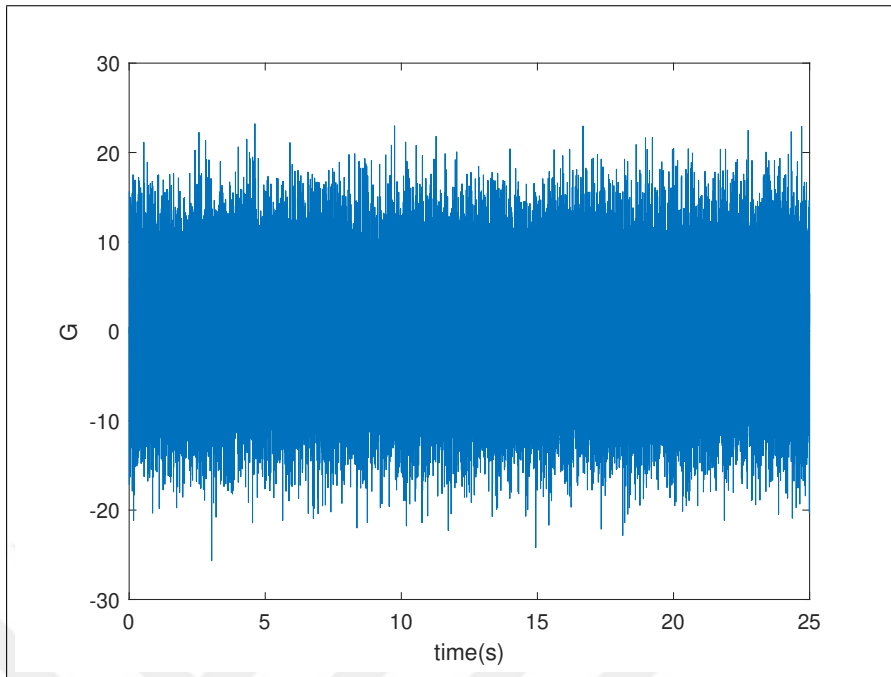


Figure 3.20: Generated random vibration.

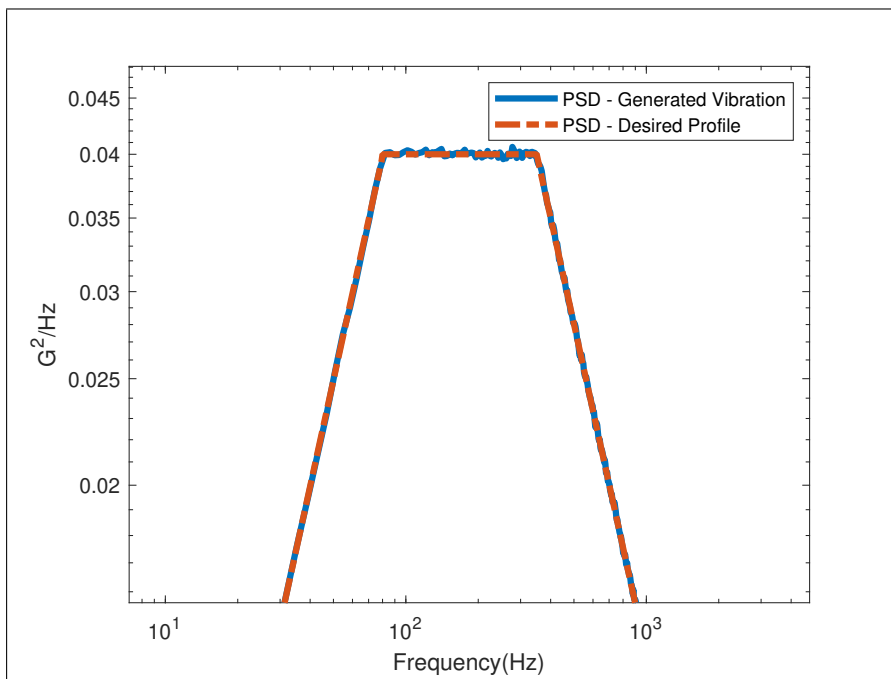


Figure 3.21: Estimated PSD of the generated data vs PSD Profile.

4. GNSS AIDED NAVIGATION

In Chapter 2.7, PVA error dynamics equations are derived from the INS mechanization equations. Although INS systems are self-contained and provide navigation solutions at a high output rate, errors in their PVA outputs accumulate over time due to initialization errors and the integration of the errors in IMU measurements. To mitigate this disadvantage, GNSS have been used together with INS. GNSS systems provide position and velocity information at consistent intervals, which are inherently absolute and not subject to cumulative drifts found in INS outputs [2].

4.1. Basics of GNSS

In satellite navigation, the GNSS is a critical component for accurate geospatial positioning and timing. GNSS employs a network of satellites in designated orbits, enabling ground receivers to determine their exact spatial coordinates and achieve time synchronization. Within GNSS, several key satellite navigation systems exist, each representing different geopolitical regions. These include the United States' Global Positioning System (GPS), Russia's GLONASS, the European Union's Galileo, and China's BeiDou. Although these subsystems operate on similar principles, they each have distinct frequencies, modulations, and error profiles. The GNSS infrastructure is divided into three principal segments: the space segment, which comprises the satellites; the control segment, responsible for system management and satellite monitoring; and the user segment, consisting of the receivers that utilize the system for navigation and timing [4].

- **Space Segment:** This segment encompasses the orbiting satellites themselves. These satellites continuously broadcast navigation signals and system data at specific frequencies. For a GNSS to provide global coverage, a substantial number of satellites are required. For instance, the GPS constellation typically consists of 24 operational satellites, ensuring that users worldwide can receive signals from at least four satellites simultaneously, which is crucial for accurate positioning.
- **Control Segment:** This comprises ground-based infrastructure responsible for

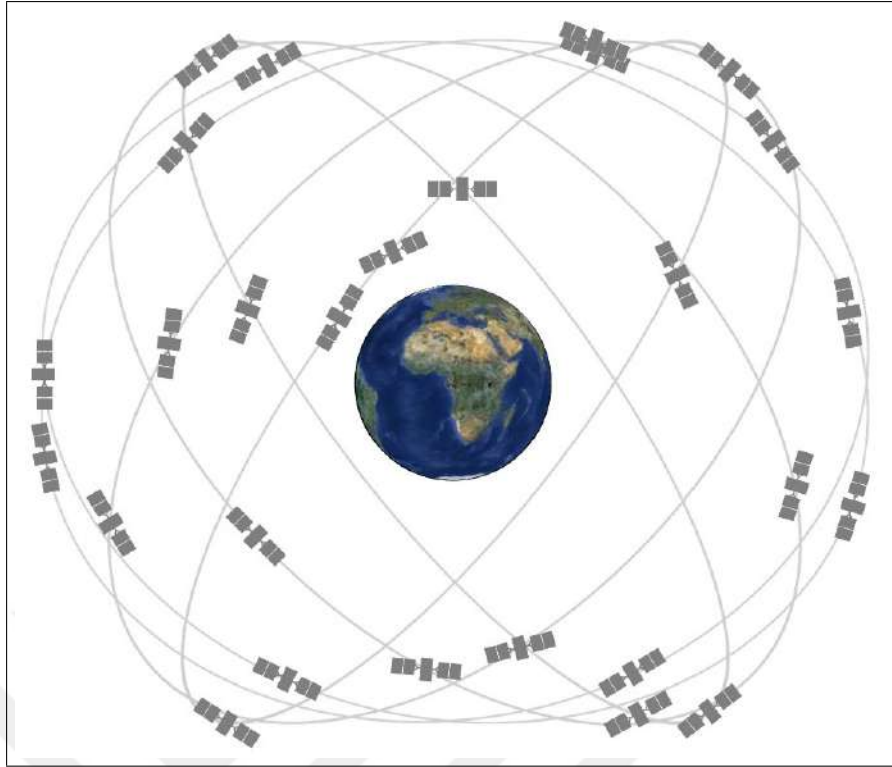


Figure 4.1: Illustration of GNSS satellites orbiting earth.

monitoring, managing, and maintaining the satellites in their proper orbits and ensuring the accuracy of their broadcast signals. The control segment stations track the satellites, update their onboard ephemeris data and monitor satellite clock accuracy. Adjustments and corrections are periodically sent to the satellites to maintain the integrity and accuracy of the system.

- **User Segment:** This segment consists of the GNSS receivers, which could be handheld devices, equipment mounted on vehicles, or integrated into various applications, such as aviation or maritime systems. These receivers intercept signals from multiple satellites, processing the information to compute the user's position, velocity, and time.

4.2. Position Estimation Using Pseudoranges

Pseudorange is defined as the measured distances between a satellite and a receiver. Each GPS satellite transmits a unique coded signal with an embedded timestamp. When the receiver captures this signal, it determines the time difference between signal transmission and reception. This time difference, when multiplied by the speed of light,

gives an initial pseudorange. This initial value is adjusted for various errors like satellite and receiver clock biases, atmospheric delays, and reflections. In (4.1) the formula of the measured pseudorange $\tilde{\rho}^{(i)}$ from the i -th satellite is given [14].

$$\begin{aligned} \tilde{\rho}^{(i)} = & \sqrt{(X^{(i)} - x)^2 + (Y^{(i)} - y)^2 + (Z^{(i)} - z)^2} \\ & + c\Delta t_r + c\Delta t_{sv}^{(i)} + c\Delta t_a^{(i)} + E^{(i)} + MP^{(i)} + \eta^{(i)} \end{aligned} \quad (4.1)$$

Where x , y , and z are the ECEF coordinates of the receiver; $X^{(i)}$, $Y^{(i)}$, and $Z^{(i)}$ are the ECEF coordinates of the i th satellite position; $c\Delta t_r$ represents the receiver clock bias error; $c\Delta t_{sv}^{(i)}$ is the satellite clock bias error; $E^{(i)}$ denotes the error in the ephemeris data; $\Delta t_a^{(i)}$ indicates the error caused by atmospheric delays; $MP^{(i)}$ signifies the multipath error; and $\eta^{(i)}$ represents the receiver tracking random errors. The superscript (i) denotes the satellite number. Let $\chi^{(i)}$ denote the sum of errors as shown in (4.2).

$$\chi^{(i)} = c\Delta t_{sv}^{(i)} + c\Delta t_a^{(i)} + E^{(i)} + MP^{(i)} + \eta^{(i)} \quad (4.2)$$

Substituting (4.2) into (4.1), the measured $\tilde{\rho}^{(i)}$ can be regarded as a function of receiver position and the receiver clock bias.

$$\tilde{\rho}^{(i)}(x, y, z, c\Delta t_r) = \sqrt{(X^{(i)} - x)^2 + (Y^{(i)} - y)^2 + (Z^{(i)} - z)^2} + c\Delta t_r + \chi^{(i)} \quad (4.3)$$

Ignoring the higher order terms, the Taylor expansion of (4.3) around an arbitrary position (x_0, y_0, z_0) , and receiver clock bias $c\Delta t_{r,0}$:

$$\begin{aligned} \tilde{\rho}^{(i)}(x, y, z, c\Delta t_r) = & \tilde{\rho}^{(i)}(x_0, y_0, z_0, c\Delta t_{r,0}) + \left. \frac{\partial \tilde{\rho}^{(i)}}{\partial x} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} (x - x_0) \\ & + \left. \frac{\partial \tilde{\rho}^{(i)}}{\partial y} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} (y - y_0) + \left. \frac{\partial \tilde{\rho}^{(i)}}{\partial z} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} (z - z_0) \\ & + \left. \frac{\partial \tilde{\rho}^{(i)}}{\partial c\Delta t_r} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} (c\Delta t_r - c\Delta t_{r,0}) + \chi^{(i)} \end{aligned} \quad (4.4)$$

where

$$\left. \frac{\partial \tilde{\rho}^{(i)}}{\partial x} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} = \frac{(X^{(i)} - x_0)}{D^i(x_0, y_0, z_0)} \quad (4.5)$$

$$\left. \frac{\partial \tilde{\rho}^{(i)}}{\partial y} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} = \frac{(Y^{(i)} - y_0)}{D^i(x_0, y_0, z_0)} \quad (4.6)$$

$$\left. \frac{\partial \tilde{\rho}^{(i)}}{\partial z} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} = \frac{(Z^{(i)} - z_0)}{D^i(x_0, y_0, z_0)} \quad (4.7)$$

$$\left. \frac{\partial \tilde{\rho}^{(i)}}{\partial c\Delta t_r} \right|_{x_0, y_0, z_0, c\Delta t_{r,0}} = 1 \quad (4.8)$$

where $D^i(x_0, y_0, z_0)$ represents the Euclidean distance between the i -th satellite and the point (x_0, y_0, z_0) , as evaluated according to (4.9).

$$D^i(x_0, y_0, z_0) = \sqrt{(X^{(i)} - x_0)^2 + (Y^{(i)} - y_0)^2 + (Z^{(i)} - z_0)^2} \quad (4.9)$$

The expressions given in (4.5), (4.6), and (4.7) represent the components of the unit vector pointing from the i th satellite to the receiver. As there are four unknowns, at least 4 pseudorange measurements are required to solve the position of the receiver. Given that $\tilde{\rho}^{(1)} \dots \tilde{\rho}^{(4)}$ are available, following linear system can be constructed.

$$\delta \mathbf{p} = H \delta \mathbf{x} + \boldsymbol{\chi} \quad (4.10)$$

where,

$$\delta \mathbf{x} = \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \\ c\Delta t_r - c\Delta t_{r,0} \end{bmatrix} \quad (4.11)$$

$$\delta \mathbf{p} = \begin{bmatrix} \tilde{\rho}^{(1)}(x, y, z, c\Delta t_r) - \tilde{\rho}^{(1)}(x_0, y_0, z_0, c\Delta t_{r,0}) \\ \tilde{\rho}^{(2)}(x, y, z, c\Delta t_r) - \tilde{\rho}^{(2)}(x_0, y_0, z_0, c\Delta t_{r,0}) \\ \tilde{\rho}^{(3)}(x, y, z, c\Delta t_r) - \tilde{\rho}^{(3)}(x_0, y_0, z_0, c\Delta t_{r,0}) \\ \tilde{\rho}^{(4)}(x, y, z, c\Delta t_r) - \tilde{\rho}^{(4)}(x_0, y_0, z_0, c\Delta t_{r,0}) \end{bmatrix} \quad (4.12)$$

$$H = \begin{bmatrix} \frac{\partial p^{(1)}}{\partial x} & \frac{\partial p^{(1)}}{\partial y} & \frac{\partial p^{(1)}}{\partial z} & \frac{\partial p^{(1)}}{\partial (c\Delta t_r)} \\ \frac{\partial p^{(2)}}{\partial x} & \frac{\partial p^{(2)}}{\partial y} & \frac{\partial p^{(2)}}{\partial z} & \frac{\partial p^{(2)}}{\partial (c\Delta t_r)} \\ \frac{\partial p^{(3)}}{\partial x} & \frac{\partial p^{(3)}}{\partial y} & \frac{\partial p^{(3)}}{\partial z} & \frac{\partial p^{(3)}}{\partial (c\Delta t_r)} \\ \frac{\partial p^{(4)}}{\partial x} & \frac{\partial p^{(4)}}{\partial y} & \frac{\partial p^{(4)}}{\partial z} & \frac{\partial p^{(4)}}{\partial (c\Delta t_r)} \end{bmatrix} \quad (4.13)$$

In (4.13) the last column of H is composed of ones. The remaining elements of H represent unit vectors; specifically, the first three columns of the i^{th} row correspond to the unit vector components in the x, y, and z directions, pointing from the i^{th} satellite to the receiver. Since χ is not known, the iterative least square algorithm [31] could be utilized to solve $\delta \mathbf{x}$.

$$\delta \mathbf{x} = H^{-1} \delta \mathbf{p} \quad (4.14)$$

If the system given by (4.10) is overdetermined, implying there are more than four pseudoranges, the pseudoinverse of H is evaluated as shown in (4.15).

$$\delta \mathbf{x} = (H^T H)^{-1} H^T \delta \mathbf{p} \quad (4.15)$$

4.3. Kalman Filter

This section provides a concise overview of the Kalman Filter (KF), an optimal state estimation technique for linear and Gaussian stochastic dynamic models. The history of Kalman Filter goes back to the early 1960s when Rudolf E. Kalman introduced this revolutionary approach [32]. The filter gained widespread recognition for its ability to predict the state of dynamic systems over time.

The objective of the KF is to estimate the state of a linear dynamic system utilizing the gathered measurement data. The process and measurement models of a linear dynamic system, in a discrete context, are presented in (4.16) and (4.17), respectively.

$$\mathbf{x}_k = \Phi_k \mathbf{x}_{k-1} + \mathbf{w}_k \quad (4.16)$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (4.17)$$

Here, $\mathbf{x}_k$ represents the state vector at a discrete time instance k , with dimensions $(n_x \times 1)$. The matrix Φ_k is the state transition matrix, describing the system's state evolution over time. H_k is the measurement matrix, and $\mathbf{z}_k$ is the measurement vector of size $(n_z \times 1)$, where $n_z \leq n_x$. The terms $\mathbf{w}_k$ and $\mathbf{v}_k$ represent Gaussian white noise processes with covariance matrices Q_k and R_k , respectively.

The KF algorithm consists of two main steps: prediction and update. In the prediction step, the state and state covariance are propagated as follows:

$$\mathbb{E}[\mathbf{x}_k | Z^{k-1}] \triangleq \hat{\mathbf{x}}_{k|k-1} = \Phi_k \hat{\mathbf{x}}_{k-1|k-1} \quad (4.18)$$

where Z^{k-1} denotes the set of all measurements up to and including time step $k-1$. Here, $\hat{\mathbf{x}}_{k-1|k-1}$ represents the estimated state at time $k-1$, conditioned on all measurements up to time $k-1$. The state covariance prediction step in estimation is performed as follows

$$\mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T | Z^{k-1}] \triangleq P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + Q_k \quad (4.19)$$

where $P_{k|k-1}$ is the predicted state covariance, Φ_k is the state transition matrix, $P_{k-1|k-1}$ is the previous state covariance, and Q_k is the process noise covariance matrix.

When a new measurement is received, the Kalman Filter update step is performed as follows:

1. Compute the Kalman Gain:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (4.20)$$

where K_k is the Kalman Gain, $P_{k|k-1}$ is the predicted state covariance, H_k is the measurement matrix, and S_k is the innovation covariance. The innovation covariance S_k is defined as:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (4.21)$$

2. Update the estimate with the new measurement:

$$E[\mathbf{x}_k | Z^{k-1}, \mathbf{z}_k] = \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1}) \quad (4.22)$$

where, $\hat{\mathbf{x}}_{k|k}$ is the updated state estimate.

3. Update the state covariance:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (4.23)$$

where $P_{k|k}$ is the updated state covariance.

A designed filter is said to be "consistent" if (4.24) and (4.25) are valid.

$$E[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}] \triangleq E[\tilde{\mathbf{x}}_{k|k}] = 0 \quad (4.24)$$

$$E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T] \triangleq E[\tilde{\mathbf{x}}_{k|k} \tilde{\mathbf{x}}_{k|k}^T] = P_{k|k} \quad (4.25)$$

where,

$$\tilde{\mathbf{x}}_{k|k} \triangleq \mathbf{x}_k - \hat{\mathbf{x}}_{k|k} \quad (4.26)$$

(4.24) confirms that the estimator is unbiased, as the expected value of the estimation error is zero. Equation (4.25) ensures covariance consistency, implying that the estimated covariance $P_{k|k}$ accurately reflects the true error covariance of the estimator. For offline estimation where true state values are available, the Normalized Estimation Error Squared (NEES) test serves as a useful tool for assessing both the accuracy of the estimated covariance and unbiasedness [13].

$$\epsilon(k) = \tilde{\mathbf{x}}_{k|k}^T P_{k|k}^{-1} \tilde{\mathbf{x}}_{k|k} \quad (4.27)$$

In (4.27), $\epsilon(k)$ follows a chi-squared distribution with n_x degrees of freedom, where n_x represents the dimensionality of the state vector. If the number of Monte Carlo runs is set to N , then one can estimate $\epsilon(k)$ as follows.

$$\bar{\epsilon}(k) = \frac{1}{N} \sum_{i=1}^N \epsilon_i(k) \quad (4.28)$$

from (4.28) it can be seen that $N\bar{\epsilon}(k)$ has chi-squared distribution with $N \times n_x$ degrees of freedom. By referring to a chi-squared distribution table and selecting an appropriate confidence level, one can determine the threshold value that $N\bar{\epsilon}(k)$ should not exceed to maintain that confidence level in the estimator's performance.

4.4. Loosely Coupled INS-GNSS Fusion

Providing absolute position and velocity solutions with limited errors makes GNSS a crucial system for mitigating the disadvantages of INS. Consequently, in nearly all air, land, and sea platforms, INS and GNSS systems are employed in tandem. In the table below, a comparison between INS and GNSS is presented. As evident from the tables above, the benefits of INS and GNSS are complementary. Consequently, the outputs of these two systems are typically fused to achieve cost-effective navigation over extended durations. INS-GNSS fusion is generally performed in two different framework: open-loop fusion versus closed-loop fusion. In the open-loop configuration, also known as the feedforward complementary structure, the estimated errors in position, velocity, and attitude are used to correct the navigation solution in a separate block; thus, they are not fed back to the INS.

Table 4.1: Attributes of INS.

Does not need external signals. Independent.
Position, speed, orientation errors increase over time.
Works in places where no signal can reach, like tunnels and underwater.
Update rate is high.
Initialization sensitivity greatly affects performance.

Table 4.2: Attributes of GNSS.

Requires signals from satellites.
Position, speed, orientation errors are limited. However, they can be blocked and mixed up.
Must have an unobstructed line of sight to satellites to operate.
Update rate is low.
Initialization sensitivity is less critical.

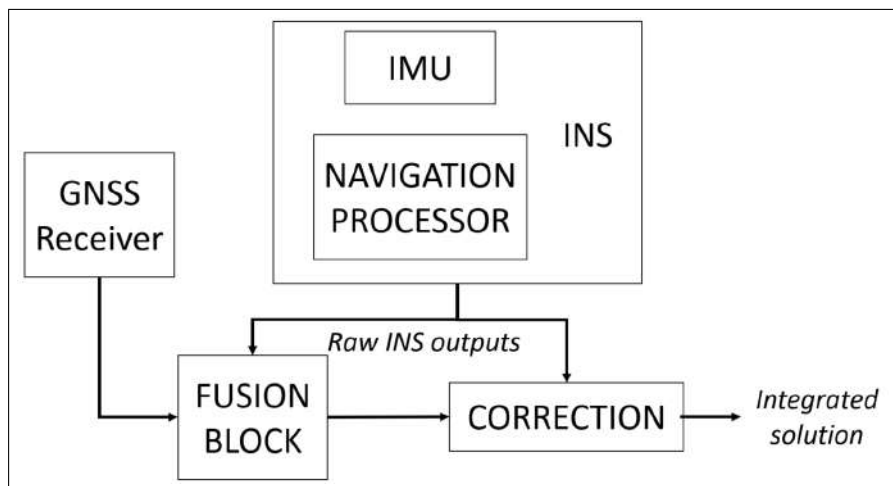


Figure 4.2: Open-loop fusion scheme.

In closed-loop design, often referred to as the feedback complementary filter, the derived errors in position, velocity, and attitude are feedback to the inertial navigation processing unit for direct correction of the INS solution. This feedback can be realized either during each iteration of the Kalman filter or at extended intervals. After every correction

feedback, the error estimates of position, velocity, and attitude in the Kalman filter are reset to zero, resulting in the absence of a standalone uncorrected INS solution. Since the error states are reset to zero after each Kalman correction step, the magnitude of the states remains controlled, thus mitigating the linearization errors inherent in the system model [5]. In Fig. 4.4, the timing specifics of the primary components of the integrated

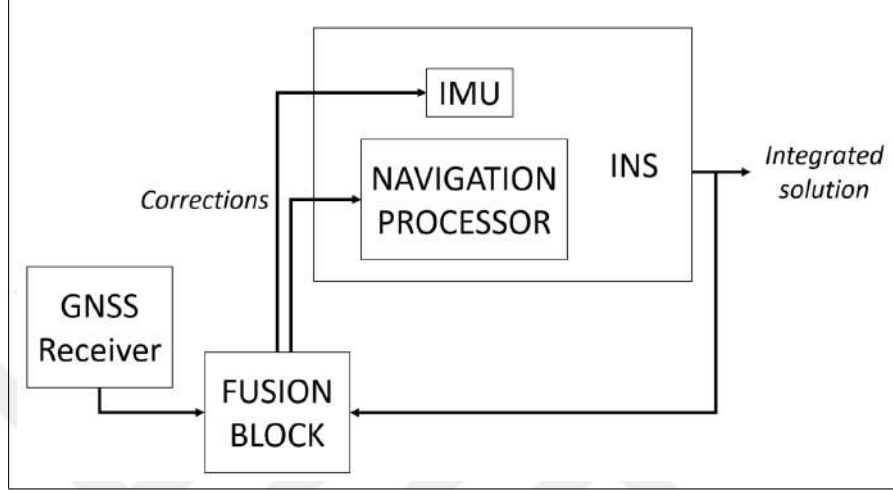


Figure 4.3: Closed-loop fusion scheme.

navigation system are displayed. Within this illustration, the smaller bars represent times of IMU data retrieval. During these periods, often referred to as the fast cycle, the PVA data from the INS is refreshed by executing the INS mechanization equations using corrected IMU measurements. The elongated bars colored in red, conversely, denote the times when GNSS position messages are received. At these junctures, commonly termed slow cycles, the INS solution is integrated with the GNSS measurements, typically employing the Kalman filter [33] or its variations [34].

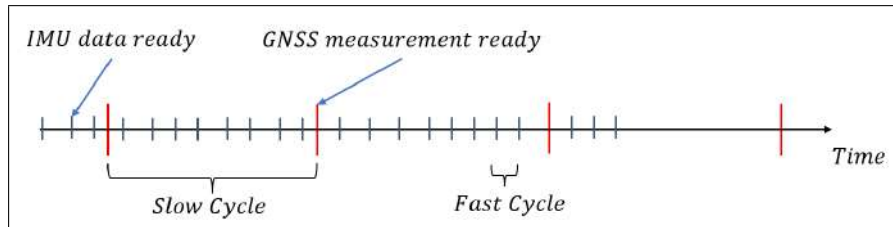


Figure 4.4: Critical timings of GNSS Integrated Navigation System.

Algorithm 1 Fast Cycle Pseudo code

Require: PVA, IMU and Estimated corrections at time t_k ,

Ensure: PVA at time t_{k+1}

- 1: **correct raw IMU measurements:**
 - 2: $\hat{\mathbf{f}}^b = \tilde{\mathbf{f}}^b - \Delta \mathbf{f}^b$
 - 3: $\hat{\boldsymbol{\omega}}_{ib} = \tilde{\boldsymbol{\omega}}_{ib} - \Delta \boldsymbol{\omega}_{ib}$
 - 4: **Perform INS mechanization using $\hat{\mathbf{f}}^b, \hat{\boldsymbol{\omega}}_{ib}$:**
 - 5: update attitude using 2.41
 - 6: update velocity using 2.46
 - 7: update position using 2.49, 2.50, 2.51
-

Algorithm 2 Slow Cycle Pseudo code

Require: PVA output of INS and GNSS measurement at t_k, R, H

Ensure: Corrected PVA at time t_k and IMU corrections $\Delta \boldsymbol{\omega}_{ib}, \Delta \mathbf{f}^b$

- 1: **Perform error covariance prediction:**
 - 2: Evaluate the state transition matrix F and Q matrices in (2.87)
 - 3: discretize F and Q matrices to get Φ and Q_d
 - 4: Evaluate predicted error covariance $P = \Phi P \Phi^T + Q_d$
 - 5: **Perform Kalman Filter Correction Step:**
 - 6: generate error measurement as given in 4.29 and 4.30
 - 7: Calculate Kalman gain: $K = PH^T(HPH^T + R)^{-1}$
 - 8: Calculate corrected error state : $\Delta \mathbf{X} = K \Delta \mathbf{z}$
 - 9: Calculate corrected error covariance: $(I - KH)P(I - KH)^T + K RK^T$
 - 10: **Feedback the estimated errors to inertial navigation solution:**
 - 11: Correct position using (4.31), (4.32), (4.33)
 - 12: Correct velocity using (4.34), (4.35), (4.36)
 - 13: Correct attitude using (4.37)
 - 14: Update IMU corrections extracting the related values of $\Delta \mathbf{X}$
-

$$\delta \mathbf{z}_{\text{pos}} = \begin{bmatrix} \text{lat}_{\text{nav}} - \text{lat}_{\text{gnss}} \\ \text{lon}_{\text{nav}} - \text{lon}_{\text{gnss}} \\ \text{alt}_{\text{nav}} - \text{alt}_{\text{gnss}} \end{bmatrix} \quad (4.29)$$

$$\delta \mathbf{z}_{\text{vel}} = \begin{bmatrix} \text{velNorth}_{\text{nav}} - \text{velNorth}_{\text{gnss}} \\ \text{velEast}_{\text{nav}} - \text{velEast}_{\text{gnss}} \\ \text{velDown}_{\text{nav}} - \text{velDown}_{\text{gnss}} \end{bmatrix} \quad (4.30)$$

Equations (4.29) and (4.30) represent the observed position and velocity errors, respectively, which are utilized as measurements in the error state Kalman filter. After the Kalman correction is applied, the estimated position and velocity errors are subtracted from the INS solution, as demonstrated in equations from (4.31) to (4.36). Furthermore, the DCM matrix, which represents the attitude of the body frame relative to the navigation frame, is updated using the estimated tilt angle vector, $\delta \hat{\boldsymbol{\psi}}$, as specified in (4.37).

$$\text{lat}^+ = \text{lat}^- - \delta \hat{\text{lat}} \quad (4.31)$$

$$\text{lon}^+ = \text{lon}^- - \delta \hat{\text{lon}} \quad (4.32)$$

$$\text{alt}^+ = \text{alt}^- - \delta \hat{\text{alt}} \quad (4.33)$$

$$\text{Vel}_N^+ = \text{Vel}_N^- - \delta \hat{\text{Vel}}_N \quad (4.34)$$

$$\text{Vel}_E^+ = \text{Vel}_E^- - \delta \hat{\text{Vel}}_E \quad (4.35)$$

$$\text{Vel}_D^+ = \text{Vel}_D^- - \delta \hat{\text{Vel}}_D \quad (4.36)$$

$$\mathbf{C}_b^m = (\mathbf{I}_{3 \times 3} - \boldsymbol{\psi} \delta \hat{\boldsymbol{\psi}})^{-1} \mathbf{C}_b^m \quad (4.37)$$

4.5. Simulation Results

In this section, we discuss the simulation results from two distinct INS-GNSS fusion scenarios.

4.5.1. Scenario-1

In the first scenario, the platform follows a straight path, then executes two consecutive coordinated turns before resuming straight flight. The platform's speed is set at 200 m/s, and both turns have an identical load factor of 1.8 G . GNSS measurements are received at 1 Hz for the whole trajectory. In Fig. 4.5, and Fig. 4.6 two dimensional (2D) trajectory of the flight path and euler angles of the platform are illustrated respectively.

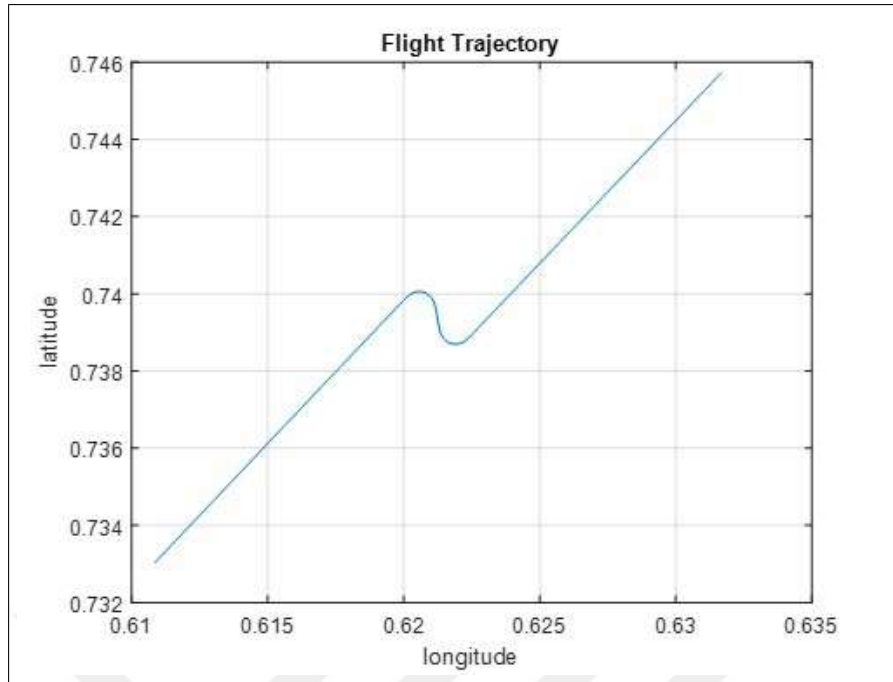


Figure 4.5: 2D trajectory.

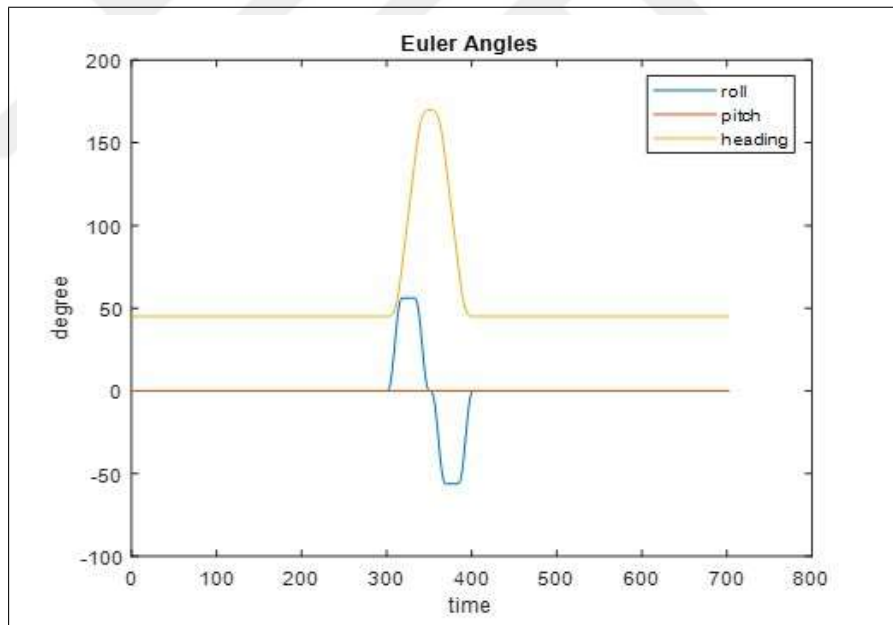


Figure 4.6: Euler angles.

In Figs 4.7, 4.8, and 4.9, the covariance analyses of position, velocity, and attitude are presented, respectively. Notably, the heading error variance increases as the platform maintains straight flight but reduces during maneuvers. This characteristic behavior of the heading error arises because the heading error and the east gyroscope bias become

indistinguishable in the absence of lateral acceleration [35]. So, it is necessary to make maneuver to reduce the heading angle error of the integrated INS system.

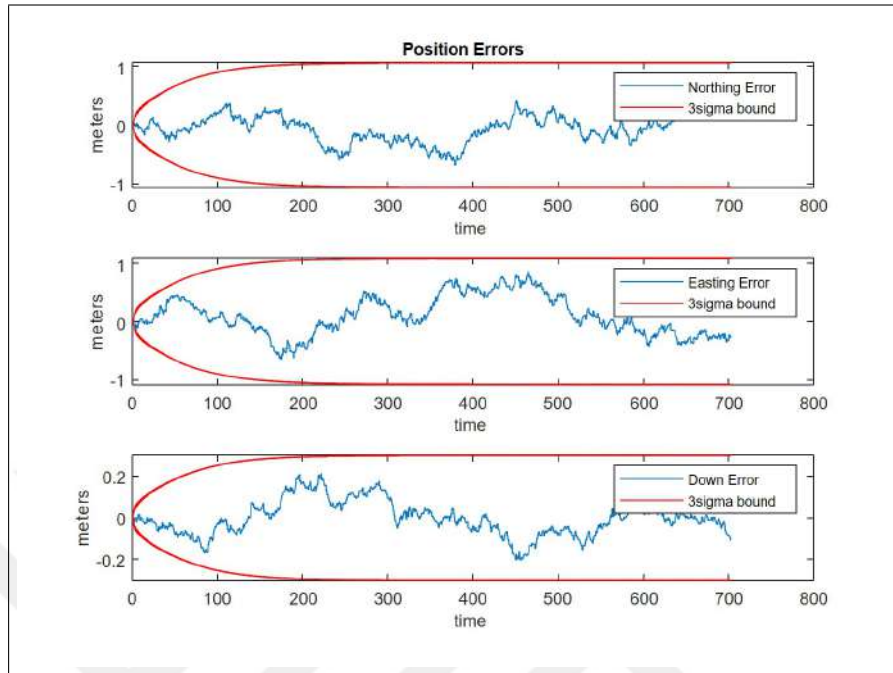


Figure 4.7: Position errors standard deviations.

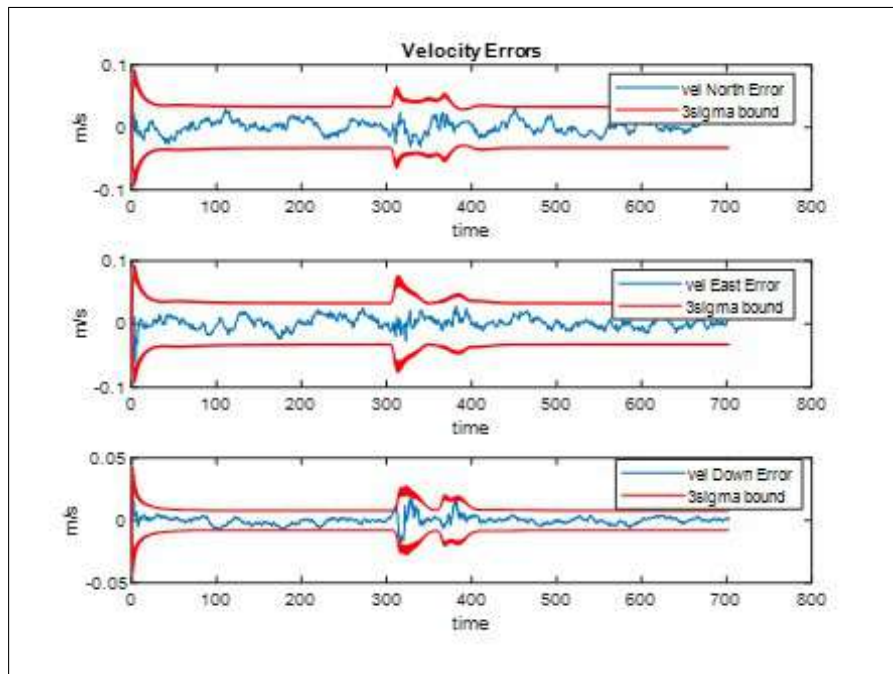


Figure 4.8: Velocity errors standard deviations.

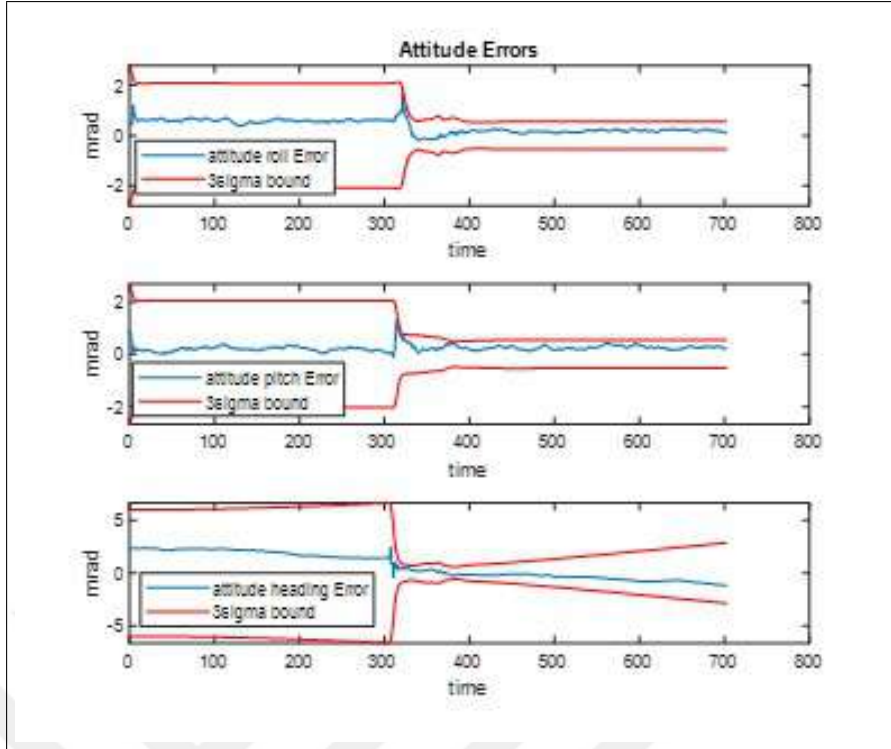


Figure 4.9: Attitude standard deviations.

4.5.2. Scenario-2

In the second scenario, the platform undergoes four consecutive straight flights, each followed by a 90° right turn, completing a loop. Figures 4.10 and 4.11 depict the 2D trajectory and the Euler angles of the platform, respectively. Unlike in the first scenario, there are some segments where GNSS measurements are invalid. In Fig. 4.10, the red-colored segments of the trajectory indicate these locations.

The covariance analyzes for the position, velocity, and attitude errors are depicted in Figures 4.12, 4.13, and 4.14, respectively. As evident from these figures, error variances grow during periods of GNSS outage.

As depicted in the covariance analysis graphs from the simulations, the combination of INS and GNSS creates a synergy that many contemporary air, land, and sea platforms have embraced. However, in GNSS-denied regions, this fusion is not possible. These challenging zones underscore the necessity for other aiding systems. These systems are supposed to offer accuracy in places where conventional GNSS signals are absent. Subsequent sections delve into these alternative systems and their significance in navigating

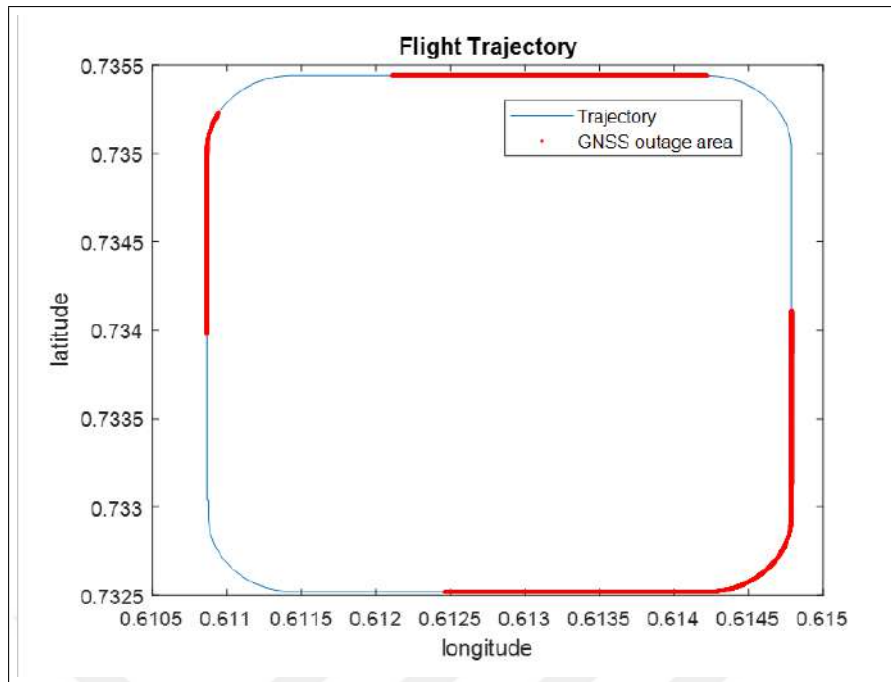


Figure 4.10: 2D trajectory.

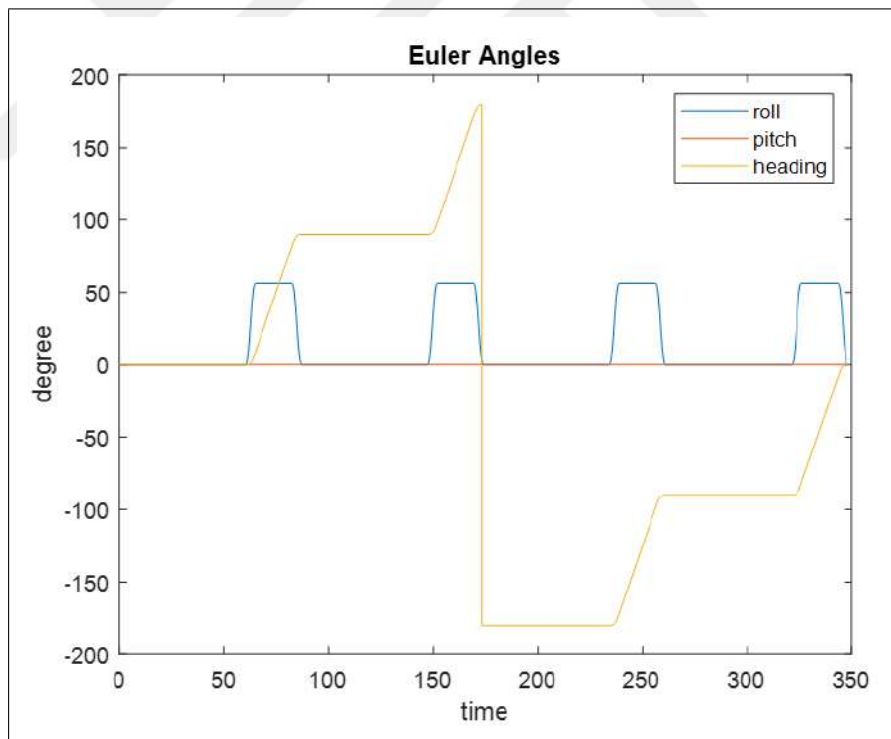


Figure 4.11: Euler angles.

GNSS-denied environments.

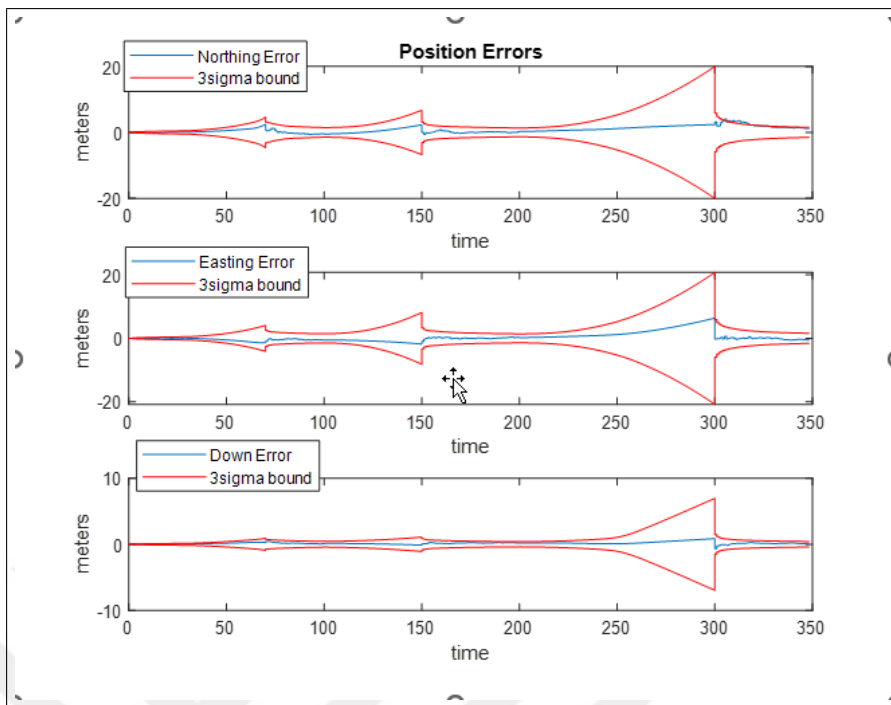


Figure 4.12: Position errors standard deviations.

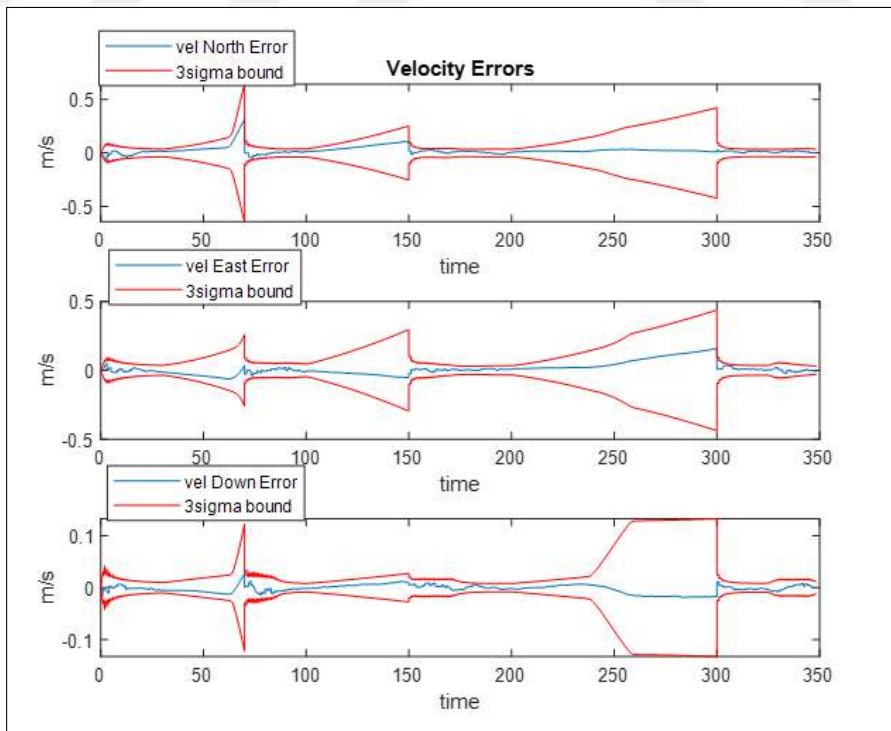


Figure 4.13: Velocity errors standard deviations.

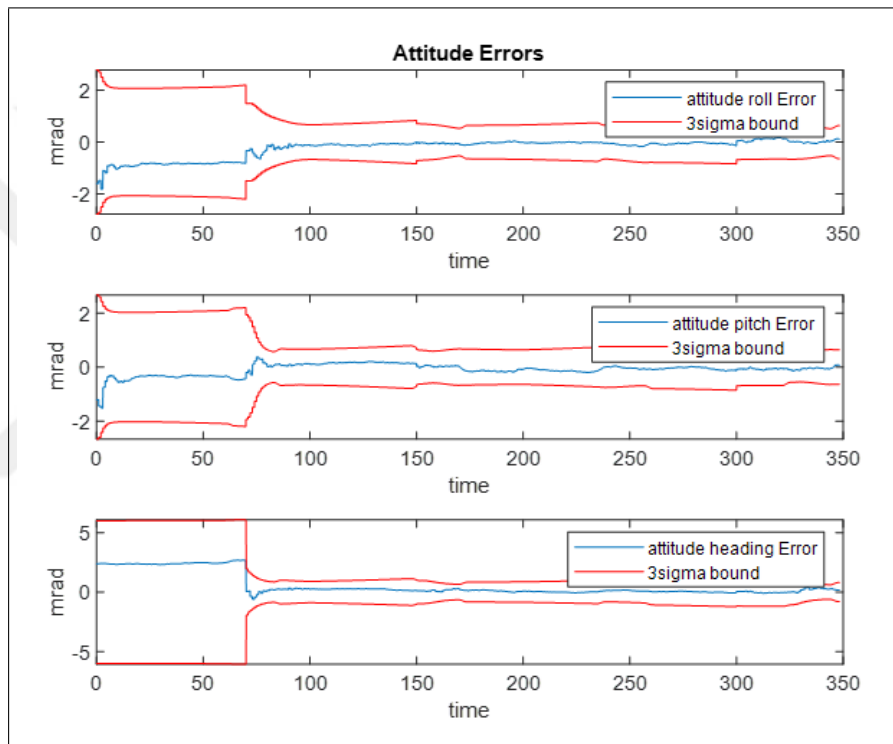


Figure 4.14: Attitude errors standard deviations.

5. TERRAIN AIDED NAVIGATION

TAN harnesses distance measuring sensors and Digital Terrain Elevation Data (DTED) to estimate a platform's location. Within this approach, an aircraft calculates its terrain profile beneath by employing distance-measuring devices like radar and pressure altimeters equipped in the aircraft. Then, the aim is to infer the position by utilizing the DTED map.

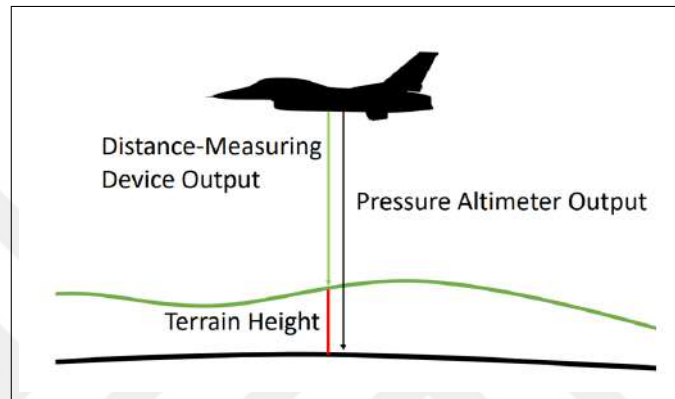


Figure 5.1: Aircraft altitude measurements: direct vs. pressure-based.

In Fig. 5.1, the green arrow represents the ground clearance, which is the distance between the terrain and the aircraft. This distance is gauged by distance-measuring devices, such as radar or laser altimeters. Conversely, the black arrow depicts the distance between the aircraft and the mean sea level (MSL), as determined by a pressure altimeter. By subtracting the output from the radar altimeter from that of the pressure altimeter, one can estimate the height of the terrain beneath the aircraft.

5.1. Literature Overview

TAN techniques fall broadly into two groups: batch and recursive. In the batch category, the Terrain Contour Matching (TERCOM) algorithm stands out in TAN literature, as it establishes position by drawing parallels between perceived terrain profiles and the DTED map [36, 37]. The method involves evaluating a cost function that contrasts terrain heights along a given trajectory at various offsets. The offset that has the minimum cost value is selected as a positional error, which is subsequently fed into

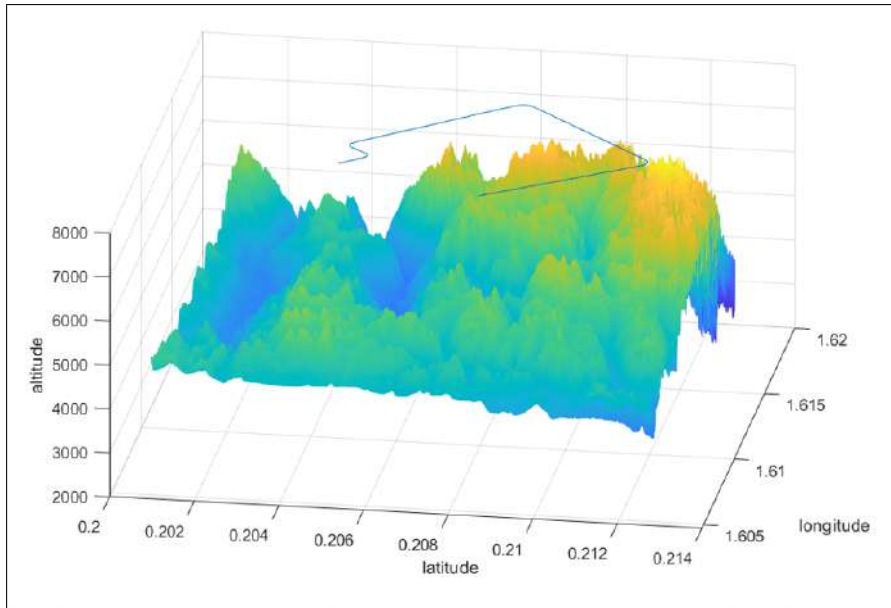


Figure 5.2: Flight trajectory and 3D terrain map.

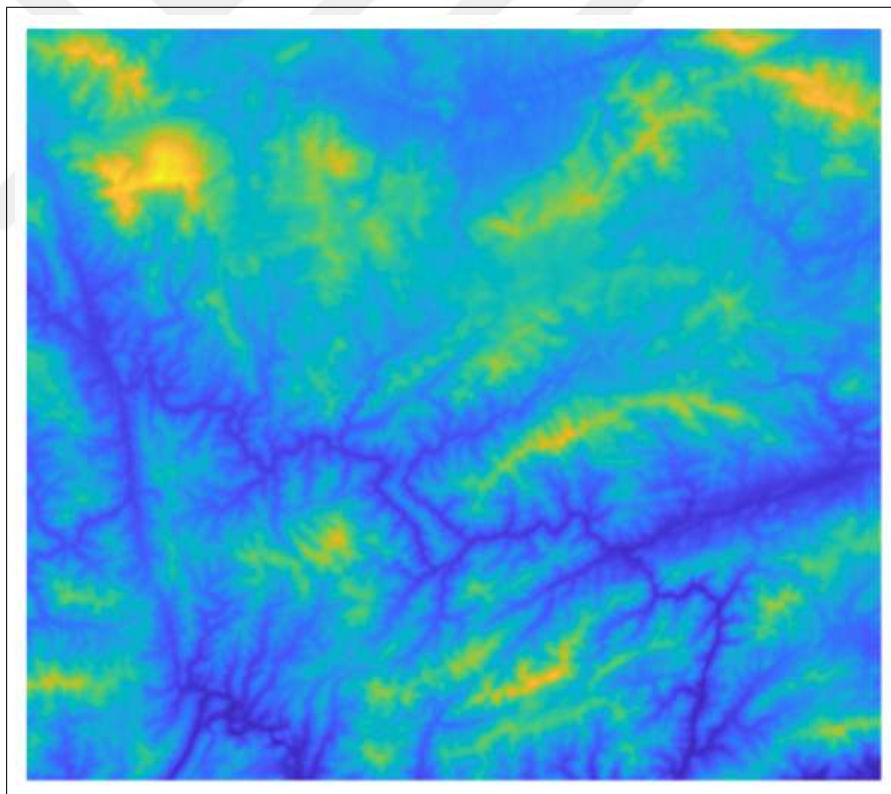


Figure 5.3: Illustration of DTED.

the INS for correction. On the recursive side, Sandia Inertial Terrain Aided Navigation (SITAN) takes the lead [38]. Here, error in terrain elevation are estimated via DTED gradients around the INS position solution. The difference between the actual and

predicted height error is utilized as an innovation in the KF.

Multiple studies have proposed variations of the TERCOM and SITAN methods. Some have paired TERCOM with different filtering techniques [39], others have customized it for land vehicles [40], and some have intertwined it with probabilistic methods to fine-tune its precision [41–46].

In the TAN literature, non-linear estimation methods also feature prominently. Enns and Morrell advanced a method based on the maximum posteriori (MAP) [47]. Meanwhile, Bergman *et al.* introduced a Bayesian model that utilizes a point mass filter, as detailed in their studies [48, 49]. Gustafsson *et al.* developed the particle filter (PF) approach specifically for TAN [50]. Follow-up PF-focused based TAN methods delved into varied aspects from resampling strategies to confidence modeling [51–56]. Many PF-based TAN techniques, by relying on simple motion models, tend to overlook the significance of IMU errors. This oversight can adversely impact performance and may even result in divergence during real-time implementations. In their work, Nordlund and Gustafsson [57] tackled this issue by integrating linearized INS error models into their PF-based TAN approach, thereby accounting for the above-mentioned IMU errors to a certain extent.

Although PFs have become popular among non-linear estimation problems like TAN, they bring some challenges as well. As state numbers increase, computational demands rise [58]. Multi-modal posterior distributions, exacerbated by terrains with similar topographies, pose another predicament. Solutions like the Interacting Weighted Ensemble Kalman Filter (IWEKF) seek to tackle this but often remain rooted in basic motion models and neglect comprehensive IMU errors [59, 60].

5.1.1. TERCOM

TERCOM [36] stands as one of the earliest and most renowned batch methods in TAN. TERCOM operates by gauging a similarity measure between the terrain height measurements gathered and those derived from DTED for various positional offsets, as showcased in Fig. 5.4. The positional offset that results in the greatest similarity between terrain heights is selected as platform's position error and is consequently fed back into the INS for position fix. As illustrated in Fig. 5.4, The grey lines show possible paths that

have been created by adding fixed offsets in position to the navigation position record denoted by the black line. In contrast, the red line denotes the true flight trajectory of the platform. The original TERCOM algorithm employed the Mean Absolute Difference (MAD) or Mean Squared Difference (MSD) as metrics to assess the disparity between the acquired terrain height array and the proposed terrain height array. The mathematical formulations for these metrics are provided in (5.1) and (5.2)

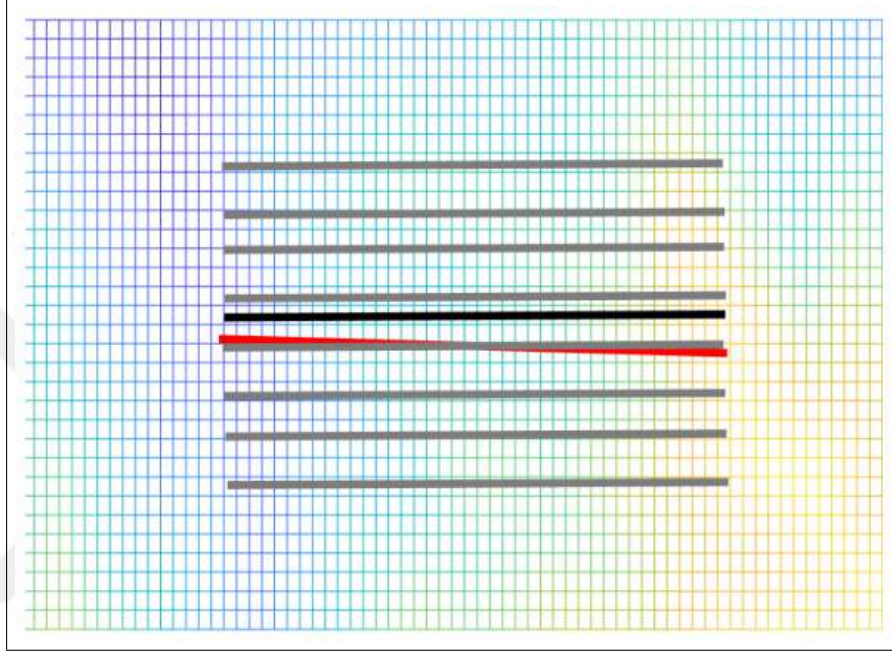


Figure 5.4: Illustration of correlation lines.

$$MAD(\delta_x, \delta_y) = \frac{1}{K} \sum_{t=1}^{t_k} |h_{TER}(t) - h_{DTED}(l_{NAV}(t) + \delta_x, L_{NAV}(t) + \delta_y)| \quad (5.1)$$

$$MSD(\delta_x, \delta_y) = \frac{1}{K} \sum_{t=1}^{t_k} (h_{TER}(t) - h_{DTED}(l_{NAV}(t) + \delta_x, L_{NAV}(t) + \delta_y))^2 \quad (5.2)$$

where δ_x and δ_y are the shift amount in latitude and longitude, $h_{TER}(t)$ is the terrain height calculated as given in (5.3), $h_{DTED}(a, b)$ is the computed terrain height at the latitude a and longitude b using DTED.

$$h_{TER}(t) = h_{BARO}(t) - h_{DME}(t) \quad (5.3)$$

Here, $h_{BARO}(t)$ represents the output from the pressure altimeter, while $h_{DME}(t)$ indicates the reading from distance measuring equipment such as radar or laser altimeter at

time t . One limitation of the TERCOM algorithm is that any error in velocity during the acquisition mode can distort the navigation trajectory relative to the actual path known as cross-track and down-track errors. Consequently, the shift amount that minimizes the values in (5.1) or (5.2) might not necessarily reflect the real position error equivalent to that shift. This underscores the necessity for TERCOM to rely on navigation-grade INS, ensuring that velocity errors are negligible [61].

5.1.2. SITAN

SITAN stands out as one of the most recognized recursive TAN methods [38]. The essence of SITAN is to harness the gradient or slope information of the terrain in proximity to the navigation solution, which is employed in the measurement model. The following part of this section provides a detailed derivation of the measurement model. In ideal (error-free) case, (5.4) would be valid.

$$0 = h_{BARO} - (h_{DTED}(l, L) + h_{GC}) \quad (5.4)$$

where, h_{BARO} denotes the altitude relative to MSL; $h_{DTED}(l, L)$ represents the undistorted terrain height evaluated at the authentic geodetic position coordinates l (latitude) and L (longitude). Meanwhile, h_{GC} is the actual ground clearance distance, gauged using either radar or laser altimeter. Given that the readings from these instruments are prone to errors and the actual position remains undetermined, the subsequent equation holds true.

$$z = \hat{h}_{BARO} - (\hat{h}_{DTED}(\hat{l}, \hat{L}) + \hat{h}_{GC}) \quad (5.5)$$

where,

$$\hat{h}_{BARO} = h_{BARO} + \delta h \quad (5.6)$$

$$\hat{l} = l + \delta l \quad (5.7)$$

$$\hat{L} = L + \delta L \quad (5.8)$$

$$\hat{h}_{GC} = h_{GC} + \delta h_{RA} \quad (5.9)$$

$$\hat{h}_{DTED}(\hat{l}, \hat{L}) = h_{DTED}(l + \delta l, L + \delta L) + \delta h_{DTED} \quad (5.10)$$

Subtracting (5.4) from (5.5), the following expression is acquired.

$$z = \delta h_{BARO} + h_{DTED}(l + \delta l, L + \delta L) + \delta h_{DTED} - h_{DTED}(l, L) + \delta h_{GC} \quad (5.11)$$

Using Taylor series expansion, $h_{DTED}(l + \delta l, L + \delta L)$ in (5.11) can be approximated around the points l and L , as shown in the following expression.

$$h_{DTED}(l + \delta l, L + \delta L) = h_{DTED}(l, L) + \frac{\partial}{\partial l} h_{DTED}(l, L) \delta l + \frac{\partial}{\partial L} h_{DTED}(l, L) \delta L \quad (5.12)$$

substituting (5.12) into (5.11)

$$z = +\delta h_{BARO} - \left(\frac{\partial}{\partial l} h_{DTED}(l, L) \delta l + \frac{\partial}{\partial L} h_{DTED}(l, L) \delta L + \delta h_{GC} + +\delta h_{DTED} \right) \quad (5.13)$$

Thus, the measurement equation in SITAN can be written in a state-space form as given in (5.14).

$$z = \begin{bmatrix} -\frac{\partial h_{DTED}(l, L)}{\partial l} & -\frac{\partial h_{DTED}(l, L)}{\partial L} \end{bmatrix} \begin{bmatrix} \delta l \\ \delta L \\ \delta h \end{bmatrix} + \delta h_{DTED, GC} \quad (5.14)$$

In the SITAN algorithm, the terms δl , δL , and δh denote the errors in latitude, longitude, and altitude, respectively. The error in the sum of the DTED and radar altimeter readings is given by $\delta h_{DTED, GC}$. The partial derivatives $\frac{\partial h_{DTED}(l, L)}{\partial l}$ and $\frac{\partial h_{DTED}(l, L)}{\partial L}$ quantify the terrain's slope along the north and east directions at the true position. Accurately calculating these gradients is a complex aspect of the SITAN approach. Although, Hollowell [38] suggests a planar fit method to estimate these gradients in the initial SITAN description, the literature also describes alternative methods that employ higher-order polynomial fits [62]. The accuracy of the estimated gradients decreases as the positional uncertainty of the INS increases.

5.2. A Novel Multi-Hypothesis Filter

Within the scope of this thesis, a new approach to TAN is developed [63], grounded in a multi-hypothesis filter-based framework. The proposed method employs multiple navigation hypotheses, propagating them with IMU measurements and correcting them using pressure and radar altimeter measurements. This multi-hypothesis approach allows for a more flexible and robust response to the varying characteristics of different terrains, enhancing the overall effectiveness and reliability of the navigation system.

5.2.1. The Notation

Before delving into the specifics of the proposed TAN method, it is essential to clarify the notation used throughout this work, particularly those related to inertial navigation. In this context, the geodetic position vector is denoted as $\mathbf{p} \triangleq [l \ L \ h]^T$, comprising the latitude (l), longitude (L), and altitude (h). Additionally, the velocity vector of the platform, when resolved in the NED frame, is represented as $\mathbf{v}^N \triangleq [v_n \ v_e \ v_d]^T$, where v_n , v_e , and v_d correspond to the velocity components in the northward, eastward, and downward directions, respectively. The orientation of the body in relation to the NED frame is represented by the Euler angle vector $\boldsymbol{\psi} \triangleq [\epsilon_r \ \epsilon_p \ \epsilon_y]^T$. This vector consists of the roll (ϵ_r), pitch (ϵ_p), and yaw (or heading) (ϵ_y) angles, all measured in radians. Furthermore, the outputs from the gyroscope and accelerometer of the IMU at a specific time t are signified by $\boldsymbol{\omega}_{\text{IB}}^B(t)$ and $\mathbf{f}_{\text{IB}}^B(t)$, respectively. In INSs, the error dynamics equation for PVA is derived by perturbing the mechanization equations around the nominal PVA solution. This process results in an error dynamics equation that is linear in form. The linearity of this equation makes it suitable for use as the process model in a KF. Consequently, in the context of INS-related fusion problems, the focus shifts to dynamically estimating the errors in PVA, rather than the actual PVA values themselves [2, 3, 64]. In this research, we adopt an augmented error state model as detailed in [14], which includes IMU-related error parameters in the state vector. Consequently, the augmented state vector set to be estimated encompasses various error components:

- Position error: $\delta \mathbf{p} \triangleq [\delta l \ \delta L \ \delta h]^T$.
- Velocity error: $\delta \mathbf{v}^N \triangleq [\delta v_n \ \delta v_e \ \delta v_d]^T$.
- Attitude error: $\delta \boldsymbol{\psi} \triangleq [\delta \epsilon_r \ \delta \epsilon_p \ \delta \epsilon_y]^T$.

Furthermore, the model accounts for various gyroscope error factors, including:

- Scale factor error: $\delta \boldsymbol{\omega}_{\text{SF}}$.
- Misalignment error: $\delta \boldsymbol{\omega}_{\text{MA}}$.
- Bias repeatability error: $\delta \boldsymbol{\omega}_{\text{BR}}$.
- Bias instability error: $\delta \boldsymbol{\omega}_{\text{BI}}$.

Additionally, it includes the accelerometer counterparts of these errors:

- Scale factor error: $\delta \mathbf{f}_{\text{SF}}$.
- Misalignment error: $\delta \mathbf{f}_{\text{MA}}$.
- Bias repeatability error: $\delta \mathbf{f}_{\text{BR}}$.
- Bias instability error: $\delta \mathbf{f}_{\text{BI}}$.

In the mentioned augmented error state model, each of the IMU errors (bias repeatability, bias instability and scale factor) is represented as a 3×1 vector, which encapsulates the corresponding error along the body's X, Y, and Z axes. An exception to this is the misalignment error, which is characterized by a 6×1 vector. Manufacturers typically provide the standard deviation values for these IMU error terms. Alternatively, such statistics can also be deduced through various analytical techniques. One notable method is the Allan variance analysis, which is a widely recognized technique for characterizing the stability and noise properties of IMU sensors. References such as [25], [65], and [66] delve into methodologies for estimating these statistics, offering insights into the precise characterization and quantification of IMU errors.

The (augmented) error state is defined as specified in (5.15).

$$\delta \mathbf{X} \triangleq \begin{bmatrix} \delta \mathbf{X}_{\text{PVA}} \\ \delta \mathbf{X}_{\text{IMU}} \end{bmatrix}_{39 \times 1} \quad (5.15)$$

where;

$$\delta \mathbf{X}_{\text{PVA}} \triangleq \begin{bmatrix} \delta \boldsymbol{\psi}^T & \delta \mathbf{v}^T & \delta \mathbf{p}^T \end{bmatrix}_{9 \times 1}^T \quad (5.16)$$

$$\delta \mathbf{X}_{\text{IMU}} \triangleq \begin{bmatrix} \delta \mathbf{X}_{\text{G}}^T & \delta \mathbf{X}_{\text{A}}^T \end{bmatrix}_{30 \times 1}^T \quad (5.17)$$

$\delta \mathbf{X}_{\text{G}}$ and $\delta \mathbf{X}_{\text{A}}$ are;

$$\delta \mathbf{X}_{\text{G}} \triangleq \begin{bmatrix} \delta \boldsymbol{\omega}_{\text{SF}}^T & \delta \boldsymbol{\omega}_{\text{MA}}^T & \delta \boldsymbol{\omega}_{\text{BR}}^T & \delta \boldsymbol{\omega}_{\text{BI}}^T \end{bmatrix}_{15 \times 1}^T \quad (5.18)$$

$$\delta \mathbf{X}_{\text{A}} \triangleq \begin{bmatrix} \delta \mathbf{f}_{\text{SF}}^T & \delta \mathbf{f}_{\text{MA}}^T & \delta \mathbf{f}_{\text{BR}}^T & \delta \mathbf{f}_{\text{BI}}^T \end{bmatrix}_{15 \times 1}^T \quad (5.19)$$

The conditional error state vector and its corresponding covariance matrix are defined as detailed in (5.20) and (5.21), respectively.

$$\widehat{\delta \mathbf{X}}(t_n | t_m) \triangleq E[\delta \mathbf{X}(t_n) | \delta \mathbf{z}^{t_0:t_m}] \quad (5.20)$$

$$P(t_n | t_m) \triangleq E \left([\widehat{\delta \mathbf{X}}(t_n | t_m) - \delta \mathbf{X}(t_n)] [\widehat{\delta \mathbf{X}}(t_n | t_m) - \delta \mathbf{X}(t_n)]^T | \delta \mathbf{z}^{t_0:t_m} \right) \quad (5.21)$$

In the presented equations, $\delta \mathbf{z}^{t_0:t_m}$ denotes the collection of observed positional errors from the initial time t_0 to time t_m . When t_n equals t_m , the left-hand side of the equations above corresponds to the *estimated* error state and the *estimated* error state covariance. Conversely, when t_n is greater than t_m , it corresponds to the *predicted* error states and the *predicted* error state covariance. To enhance conciseness throughout this section, the superscripts $+$ and $-$ are employed to denote the estimated and predicted states or covariance matrices, respectively. Furthermore, the dynamics of the actual error state can be effectively delineated using a linear state-space model, as stated in (5.22).

$$\delta \dot{\mathbf{X}}(t) = F(t)\delta \mathbf{X} + G(t)\mathbf{w}(t) \quad (5.22)$$

Here, $F(t)_{39 \times 39}$ denotes the state transition matrix. The vector $\mathbf{w}(t)_{12 \times 1}$ represents process noise, where each element is a standard normal random variable. These variables represent the white process noise components, such as VRW and ARW characteristics, inherent to IMUs. Additionally, $G(t)_{39 \times 12}$ is defined as the noise gain matrix. Correspondingly, the measurement equation is given in the following manner:

$$\delta \mathbf{z}(t) = H\delta \mathbf{X}(t) + \mathbf{v}(t) \quad (5.23)$$

where $\delta \mathbf{z}(t)$ symbolizes the measured error, H denotes the measurement matrix, $\delta \mathbf{X}(t)$ represents the state error, and $\mathbf{v}(t)$ encapsulates the measurement noise. Equations (5.22) and (5.23) are linear, and the components of process noise $\mathbf{w}(t)$ and the measurement noise $\mathbf{v}(t)$ are modeled as zero-mean Gaussian variables, independent of each other. This independence is expressed mathematically as $(E[\mathbf{w}(t_n)\mathbf{v}(t_m)]) = \mathbf{0} \quad \forall (t_m, t_n)$. Furthermore, these variables possess covariance matrices $Q(t)$ and $R(t)$ respectively. Consequently, the application of the KF for the recursive estimation of $\widehat{\delta \mathbf{X}}$ and its covariance matrix P is a conventional procedure in addressing fusion problems related to inertial navigation. In an illustrative case such as loosely-coupled INS-GNSS integration, the term $\delta \mathbf{z}(t)$ is evaluated by subtracting the lever-arm compensated geodetic position, as outputted by the GNSS receiver, from the geodetic position determined by the INS [5]. This approach to observing the position error enables the visibility of all components of the $\delta \mathbf{X}_{PVA}$ state, in addition to certain elements of the $\delta \mathbf{X}_G$, and $\delta \mathbf{X}_A$, especially as the platform undergoes various maneuvers [67]. However, in TAN, the measurement of terrain height beneath the platform does not directly inform about the platform's horizontal position (latitude, longitude). To resolve this horizontal location ambiguity,

correlation-based methods such as TERCOM compare the gathered terrain height profile with DTED. Nonetheless, a fundamental assumption in correlation-based approaches is the constancy of position errors during the acquisition of terrain height. The use of tactical or sub-tactical grade IMU, or the presence of large initial velocity error at the start of the acquisition mode, can result in the accumulation of cross-track or along-track errors by the end of the mode. This, in turn, can significantly degrade the accuracy of position error estimation in correlation-based methods [61]. Linearized KF based methods, such as SITAN, address the horizontal position ambiguity by incorporating terrain slope information around the estimated position into the measurement model. However, the linearization process inherent in these methods exhibits a high sensitivity to the terrain's topography, often leading to inaccurate estimates, particularly when the initial position uncertainty is substantial [43]. The main goal of our research is to devise a TAN algorithm that demonstrates robustness against significant initial position, velocity, and attitude errors and maintains adequate performance even with the utilization of lower-grade IMUs.

5.2.2. Multi-Hypothesis Structure In TAN

Hypotheses are the core components in the proposed method. The framework of the proposed method is depicted in Fig. 5.5, which illustrates a loop that begins and concludes with the propagation of the hypothesis set, representing a single iteration of the algorithm. Between terrain aiding cycles, hypotheses are propagated via INS mechanization. During the terrain aiding cycles, multiple horizontal measurements and their covariance matrices are generated based on the statistics of clustered potential locations derived from the hypotheses. These potential locations, distinct for each hypothesis, are identified within a certain percentage of the hypothesis' horizontal uncertainty region. Once these horizontal measurements are formulated, they are associated with and integrated into the suitable hypotheses. This process involves merging similar hypotheses and pruning less likely ones to manage computational demands effectively. The proposed Multi-Hypothesis method includes specialized covariance adjustments to prevent divergence, particularly during instances when the platform is hovering over terrain that is either non-informative or has a multi-modal shape.

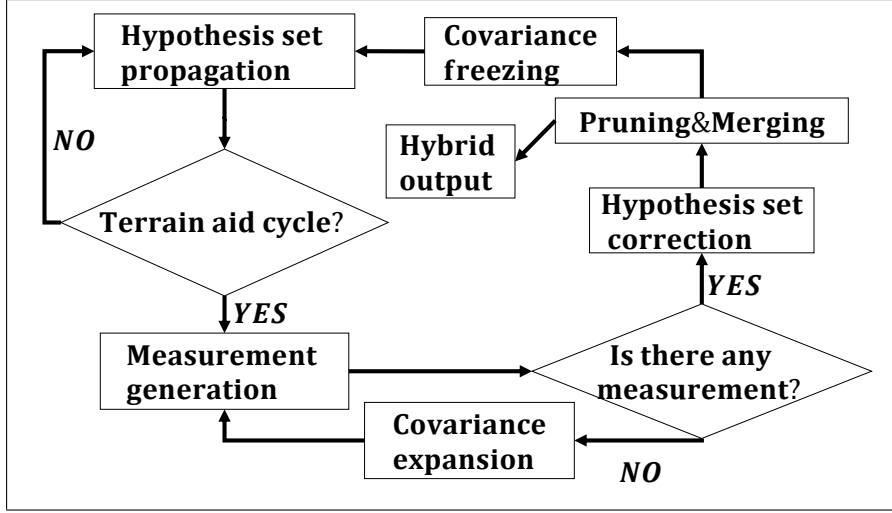


Figure 5.5: Flowchart of the proposed Multi-Hypothesis TAN algorithm.

5.2.2.1. The Definition of Hypothesis & multi-hypothesis

In the proposed Multi-Hypothesis method, each hypothesis is conceived as a separate imaginary INS, characterized by unique navigation attributes, an error state, and an error state covariance matrix. These elements are processed independently and in parallel, utilizing terrain height information. The method implements individual KFs for each hypothesis, which operate concurrently. These filters estimate the error state of their respective hypotheses during each terrain aid cycle, assuming they receive the relevant measurements. The symbol ξ is used to denote a hypothesis, and $\beta^{\xi_i}(t)$ represents the specific feature β of the hypothesis ξ_i at time t . The feature β can represent any PVA property, error state, or error state covariance as detailed in Section-5.2.1. Therefore, β could be any symbol from the set $\{\hat{\mathbf{p}}, \hat{\mathbf{v}}, \hat{\boldsymbol{\psi}}, \delta\hat{\mathbf{X}}, P\}$. If β has a subscript, it signifies that only the elements indicated by the subscript are included. For instance, $\hat{\mathbf{p}}^{\xi_i}(t)$ denotes the 3×1 geodetic position vector $[l \ L \ h]^T$ of hypothesis ξ_i at time t , while $\hat{\mathbf{p}}_{l,L}^{\xi_i}(t)$ refers to the 2×1 vector containing only the horizontal components $[l \ L]^T$ of the geodetic position vector. Similarly, $\delta\hat{\mathbf{X}}^{\xi_i}(t)$ indicates the entire 39×1 error state of hypothesis ξ_i at time t , whereas $\delta\hat{\mathbf{X}}_{\text{N,VE}}^{\xi_i}(t)$ specifically represents the 2×1 error state vector pertaining to the north and east velocity errors of hypothesis ξ_i at time t . Moreover, a multi-hypothesis set, represented as $\Xi(t)$, is essentially an ensemble of individual hypotheses at time t , such as $\{\xi_1, \xi_2, \dots, \xi_n\}$, where ξ_i (for $i = 1, 2, \dots, n$) symbolizes distinct hypotheses at that specific time.

5.2.2.2. Hypothesis Set Propagation

We assume that measurements of the ground clearance distance, acquired via a distance-measuring device like a radar or laser altimeter, are fused with the INS at specific instances denoted as $t_{Ra(k)}$ for k in the set $\{1, 2, \dots, K\}$, where $t_{Ra(k)}$ signifies the k^{th} terrain aid cycle time.

The terms Δt_{IMU} and T_{IMU} are used to represent the time interval between two consecutive IMU measurements and the collection of times at which IMU measurements are obtained, respectively. The hypothesis propagation step is executed during the time frames $T_{Pr}(k)$ for k in $\{1, 2, \dots, K\}$. Here, $T_{Pr}(k)$, as defined in (5.24), corresponds to the set of IMU times that fall between two terrain aid cycles $t_{Ra(k-1)}$ and $t_{Ra(k)}$.

$$T_{Pr}(k) \triangleq \{t | t_{Ra(k-1)} < t \leq t_{Ra(k)}, t \in T_{IMU}\} \quad (5.24)$$

The term "hypothesis set propagation" refers to the process of updating all attributes of each hypothesis in the set between two terrain aid cycles. To execute this propagation, the PVA error state, and error covariance of each hypothesis in the set $\Xi^+(t_{Ra(k-1)})$ are propagated using the set of IMU and barometer measurements from time $t_{Ra(k-1)}$ to $t_{Ra(k)}$. This set is denoted as $\{\omega_{IB}^B(t), \mathbf{f}^B(t), h_B(t) | t \in T_{Pr}(k)\}$. The resulting hypotheses at the end of the cycle constitute the predicted hypothesis set $\Xi^-(t_{Ra(k)})$.

This propagation process is detailed in Algorithm-3. Here, $\hat{\mathbf{X}}_{PVA}^\xi$ represents the PVA vector, which includes the attitude ($\hat{\psi}^\xi$), velocity ($\hat{v}^\xi$), and position ($\hat{p}^\xi$) vectors of the hypothesis ξ . $\delta \hat{\mathbf{X}}^\xi$ and P^ξ represent the error state and the error state covariance matrix of ξ , respectively.

In line 4 of the algorithm, the raw IMU measurements are corrected using the estimated IMU error states from the previous terrain aid cycle for each hypothesis. Lines 5 to 7 update the attitude, velocity, and position of the individual hypotheses using discrete INS mechanization equations as detailed in the Chapter-2. Line 8 performs baro-inertial vertical channel filter [3]. The mechanization equations are not elaborated here to maintain focus on the main subject.

Finally, in lines 9 and 10, the KF error state and error state covariance matrix propagation are performed. Here, $\Phi(t)$ and $Q_d(t)$ denote the discrete state transition and process noise covariance matrices, respectively. These matrices are derived by discretizing their

continuous counterparts, $F(t)$ and $Q(t)$, using Van Loan's method [68]. The elements of $Q(t)$ are derived from the second-order statistics of the IMU's stochastic errors such as ARW, VRW or the driving noise power of bias instability.

Algorithm 3 Hypothesis Set Propagation

Input: $\Xi^+(t_{Ra(k-1)})$, $\{\omega_{IB}^B(t), \mathbf{f}^B(t), h_B(t) \mid t \in T_{Pr}(k)\}$

Output: $\Xi^-(t_{Ra(k)})$

```

1:  $\Xi^-(t_{Ra(k)}) = \emptyset$ 
2: for each  $\xi \in \Xi^+(t_{Ra(k-1)})$  do
3:   for each  $t \in T_{Pr}(k)$  do
4:      $[\bar{\omega}_{IB}^B, \bar{\mathbf{f}}^B] = \text{correctIMU}(\omega_{IB}^B(t), \mathbf{f}^B(t), \delta \hat{\mathbf{X}}_{IMU}^\xi)$ 
5:      $\hat{\mathbf{X}}_{PVA}^\xi = \text{updateAttitude}(\hat{\mathbf{X}}_{PVA}^\xi, \bar{\omega}_{IB}^B)$ 
6:      $\hat{\mathbf{X}}_{PVA}^\xi = \text{updateVelocity}(\hat{\mathbf{X}}_{PVA}^\xi, \bar{\mathbf{f}}^B)$ 
7:      $\hat{\mathbf{X}}_{PVA}^\xi = \text{updatePosition}(\hat{\mathbf{X}}_{PVA}^\xi)$ 
8:      $\hat{\mathbf{X}}_{PVA}^\xi = \text{stabilizeVerticalChannel}(\hat{\mathbf{X}}_{PVA}^\xi, h_B(t))$ 
9:      $\delta \hat{\mathbf{X}}^\xi = \Phi(t) \delta \hat{\mathbf{X}}^\xi$ 
10:     $P^\xi = \Phi(t) P^\xi \Phi(t)^T + Q_d(t)$ 
11:   end for
12:   append  $\xi$  into  $\Xi^-(t_{Ra(k)})$ 
13: end for
```

5.2.2.3. Measurement Generation

The measurement generation step occurs at the start of every terrain aid cycle, Its primary objective is to create horizontal measurements and their corresponding covariance matrices within the uncertainty regions of the available hypotheses. This is accomplished by utilizing the collected altimeter data and the DTED map. To facilitate this process, several key variables are defined based on the geodetic position $\mathbf{p} = [l \ L \ h]^T$. These include:

- $h_{Ter}(\mathbf{p}_{l,L})$: Represents the true terrain height at any given 2D horizontal location $\mathbf{p}_{l,L}$.
- $\tilde{h}_{Bar}(\mathbf{p})$: Denotes the measured barometric altitude at the geodetic position $\mathbf{p}$.

- $\tilde{h}_{Ra}(\mathbf{p})$: Indicates the measured ground clearance distance at the geodetic position $\mathbf{p}$.

Using these variables, the true terrain height at any arbitrary 2D horizontal location $\mathbf{p}_{l,L}$ can be mathematically expressed as shown in (5.25).

$$h_{Ter}(\mathbf{p}_{l,L}) = \left(\tilde{h}_{Bar}(\mathbf{p}) - \Delta_{Bar} \right) - \left(\tilde{h}_{Ra}(\mathbf{p}) - \Delta_{Ra} \right) \quad (5.25)$$

Δ_{Bar} and Δ_{Ra} signify the errors associated with the pressure altimeter and the radar (or laser) altimeter respectively. Likewise, the relationship between the actual terrain height and the terrain height computed through the DTED map at $\mathbf{p}_{l,L}$ is formulated as presented in (5.26).

$$h_{Ter}(\mathbf{p}_{l,L}) = \mathcal{H}(\mathbf{p}_{l,L}) - \Delta_{Dted} \quad (5.26)$$

In this case, $\mathcal{H}(\mathbf{p}_{l,L})$ represents the terrain height referenced to DTED, which is computed at the horizontal position $\mathbf{p}_{l,L}$ through bilinear interpolation utilizing the heights of four surrounding grid points. Δ_{Dted} is the error associated with the DTED at $\mathbf{p}_{l,L}$. Assuming the error terms in (5.25) and (5.26) are independent, have zero mean, and follow Gaussian distributions with variances σ_{Bar}^2 , σ_{Ra}^2 , and σ_{Dted}^2 respectively, the error e can be characterized. This error e represents the difference between the estimated terrain height derived from altimeter readings and the terrain height determined through DTED at the point $\mathbf{p}_{l,L}$.

$$\begin{aligned} e &\triangleq \hat{h}_{Ter}(\mathbf{p}_{l,L}) - \mathcal{H}(\mathbf{p}_{l,L}) \\ &= \{h_{Bar}(\mathbf{p}) - h_{Ra}(\mathbf{p})\} - \mathcal{H}(\mathbf{p}_{l,L}) \\ &= \Delta_{Bar} - \Delta_{Ra} - \Delta_{Dted} \end{aligned} \quad (5.27)$$

Given the aforementioned statistical characteristics of the errors, the variable e would also have a zero-mean Gaussian distribution. The variance of e , denoted as σ_{vert}^2 , is determined by the sum of the variances of the individual error components. This relationship is mathematically expressed as:

$$\sigma_{vert}^2 = \sigma_{Bar}^2 + \sigma_{Ra}^2 + \sigma_{Dted}^2 \quad (5.28)$$

Utilizing the distribution of e , we allocate scores to points within a specified region, correlating with a predefined percentage of the horizontal position uncertainty of a hypothesis. The scoring function $f_{sc}(\bar{\mathbf{p}}_{l,L}; \sigma_{vert}, \hat{h}_{Ter}(\mathbf{p}(t)))$ is established to assign a

score to any arbitrary position $\bar{\mathbf{p}}_{l,L}$. The score of this position is the scaled likelihood of observing the terrain height $\hat{h}_{Ter}(\mathbf{p}(t))$, given that $\mathcal{H}(\bar{\mathbf{p}}_{l,L})$ is true terrain height. This likelihood can be computed as presented in (5.29), where $\hat{h}_{Ter}(\mathbf{p}(t))$ is denoted as $\hat{h}_{Ter}(t)$ for the sake of simplicity, representing the estimated terrain height at time t when the platform is at the position $\mathbf{p}$.

$$\begin{aligned} f_{sc}(\bar{\mathbf{p}}_{l,L}; \sigma_{vert}, \hat{h}_{Ter}(t)) &\triangleq \sqrt{2\pi\sigma_{vert}^2} \mathcal{L}(\hat{h}_{Ter}(t) | \mathcal{H}(\bar{\mathbf{p}}_{l,L})) \\ &= \sqrt{2\pi\sigma_{vert}^2} N(\hat{h}_{Ter}(t) - \mathcal{H}(\bar{\mathbf{p}}_{l,L}); 0, \sigma_{vert}^2) \\ &= \exp\left(-0.5 \frac{(\hat{h}_{Ter}(t) - \mathcal{H}(\bar{\mathbf{p}}_{l,L}))^2}{\sigma_{vert}^2}\right) \end{aligned} \quad (5.29)$$

5.2.2.4. Gridding

In this section of the algorithm, our objective is to create a set of approximately equidistant two-dimensional points situated within the p_c percent probability region of a given hypothesis ξ . As defined in (5.30), $O^\xi(d_g)$ denotes the set of points where the angular separation distance is equivalent to d_g in both the north and east directions.

$$O^\xi(d_g) \triangleq \left\{ \hat{\mathbf{p}}_{l,L}^\xi - \begin{bmatrix} i \\ j \end{bmatrix} d_g \mid i, j \in \mathbb{Z} \right\} \quad (5.30)$$

In this equation, $\mathbb{Z}$ represents the set of all integers, and $\hat{\mathbf{p}}_{l,L}^\xi$ is the estimated horizontal coordinate of the hypothesis ξ . A *gridding* function G is defined for a set of hypotheses as shown in (5.31).

$$G(\xi; p_c, d_g) \triangleq \left\{ \mathbf{x} | (\mathbf{x} - \hat{\mathbf{p}}_{l,L}^\xi)^T (P_{l,L}^\xi)^{-1} (\mathbf{x} - \hat{\mathbf{p}}_{l,L}^\xi) < \chi_{p_c}^2, \mathbf{x} \in O^\xi(d_g) \right\} \quad (5.31)$$

In this expression, p_c represents the coverage probability and $\chi_{p_c}^2$ is the chi-square inverse cumulative distribution function with two degrees of freedom, evaluated at p_c . Using the horizontal position covariance matrix $P_{l,L}^\xi$, the function $G(\xi; p_c, d_g)$, abbreviated as $G(\xi)$, generates a grid of equidistant two-dimensional points within the area that covers p_c percent of the horizontal uncertainty region for the hypothesis ξ . We apply the function described in (5.29) to assign scores to the points in $G(\xi; p_c, d_g)$. Subsequently, a new subset of points, $\Upsilon_1^\xi(\tau_s) \subseteq G(\xi)$, is formed, containing only those

points whose scores exceed a specified threshold τ_s . Therefore, any point $\mathbf{x}$ in $\Upsilon_1^\xi(\tau_s)$ at time t satisfies to the following criterion.

$$f_{sc}(\mathbf{x}; \sigma_{vert}, \hat{h}_{Ter}(t)) > \tau_s \quad (5.32)$$

Additionally, we define $\Upsilon_0^\xi(\tau_s) = G(\xi) \setminus \Upsilon_1^\xi(\tau_s)$ to represent the set of points in $G(\xi)$ that score below the threshold τ_s . To clarify, we depict $\Upsilon_1^\xi(\tau_s)$ and $\Upsilon_0^\xi(\tau_s)$, abbreviated as Υ_1^ξ and Υ_0^ξ respectively, on the same plot in Fig. 5.6. In this example, the threshold τ_s is set to 0.01. Consequently, Υ_1^ξ comprises points from $G(\xi)$ where the terrain height calculated using DTED falls within the range of $\hat{h}_{Ter} - 3\sigma_{vert}$ to $\hat{h}_{Ter} + 3\sigma_{vert}$. In Fig. 5.6, the blue marker at the circle's center signifies $\hat{\mathbf{p}}_{l,L}^\xi$, which are 2D coordinates(latitude, longitude) of the hypothesis ξ . The area within the blue dashed circle corresponds to the 99% uncertainty region for the horizontal position of ξ , where latitude and longitude errors are uncorrelated and possess equal variances, resulting in nearly circular shape. The red points, forming Υ_1^ξ , are labeled as *probable platform positions* since the disparity between their calculated terrain heights and the observed terrain height $\hat{h}_{Ter}$ lies within the $3\sigma_{vert}$ interval of e defined in (5.27). Thus, it is logical to assume that the true position is within the area where red dots are located. Conversely, the elements of Υ_0^ξ , marked in black, are considered as *improbable positions* because the deviation between their calculated terrain heights and $\hat{h}_{Ter}$ exceeds the $3\sigma_{vert}$ range of e .

5.2.3. Clustering

As the position uncertainty of the hypothesis ξ grows, the size of the set Υ_1^ξ increases. Moreover, due to resemblances in terrain height, there might be numerous hypotheses. Consequently, developing a fusion method that processes each element of Υ_1^ξ individually could become computationally burdensome. To address this challenge, we model Υ_1^ξ as a Gaussian mixture (GM) by segmenting it into smaller subsets and extracting the first and second moments of each subset. The k-means clustering algorithm [69] is employed for this purpose. In (5.33), A^ξ represents the approximate area of the 99% horizontal position error uncertainty region for ξ in horizontal geodetic coordinates.

$$A^\xi \triangleq \pi \left(1.5(\sigma_n^\xi + \sigma_e^\xi) \right)^2 \quad (5.33)$$

where σ_n^ξ and σ_e^ξ refer to the square roots of the diagonal elements of $P_{l,L}^\xi$, corresponding to the standard deviations of latitude and longitude errors for the hypothesis ξ . The

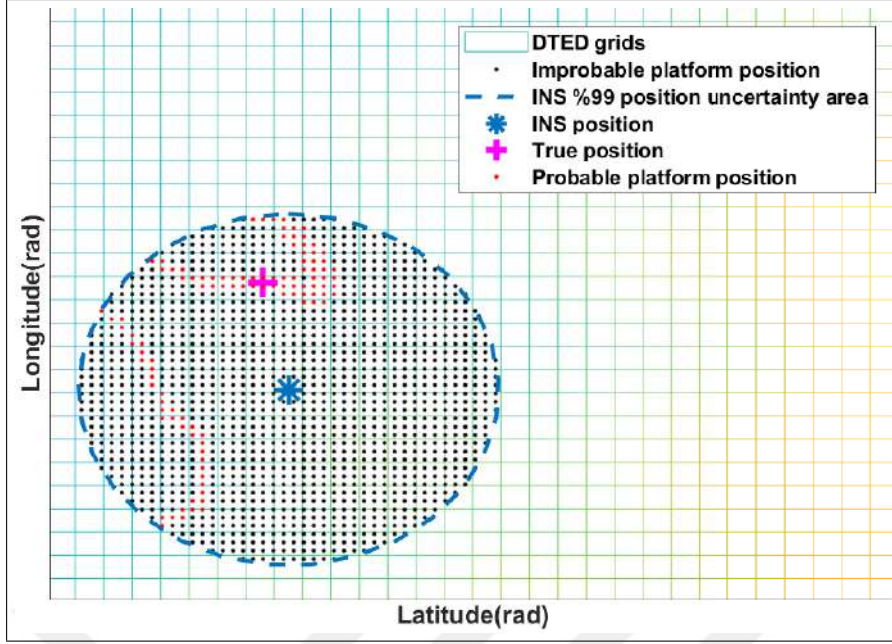


Figure 5.6: Demonstration of Υ_1^ξ and Υ_0^ξ .

variables r_c and n_c in (5.34) signify, respectively, the minimum radius of a cluster in meters and the minimum number of members required for a cluster. The number of clusters for the hypothesis ξ is determined as depicted in (5.34).

$$K^\xi = \max \left(\left\lfloor \min \left(\frac{|\Upsilon_1^\xi| A^\xi}{|G(\xi)|} / \frac{\pi (3r_c)^2}{r_e^2}, \frac{|\Upsilon_1^\xi|}{n_c} \right) \right\rfloor, 1 \right) \quad (5.34)$$

where $|\cdot|$ is used to denote the cardinality of a set and $r_e \approx 6.4 \times 10^6$ meters is the average radius of Earth's curvature at a latitude of $\pi/4$. The rationale behind the formula in (5.34) is as follows: Firstly, the $\max(\cdot)$ function on the right-hand side ensures that K^ξ is at least one. The first term within the $\min(\cdot)$ function signifies the ratio of the area covered by the elements of Υ_1^ξ to the area of a circle with a radius of $3r_c$ meters, scaled by r_e^2 . The second term inside the $\min(\cdot)$ function corresponds to the number of clusters assuming each cluster contains n_c members. The operator $\lfloor \cdot \rfloor$ yields the largest integer less than or equal to the result of the $\min(\cdot)$ function. The equation in (5.34) is designed to define a relationship between the number of clusters, the spatial size of the uncertainty region of ξ , and the cardinality of the set Υ_1^ξ .

This clustering process results in K^ξ subsets $\theta_i \subseteq \Upsilon_1^\xi$, where i ranges from 1 to K^ξ . We generate K^ξ measurements, with values and covariance matrices derived from (5.35)

and (5.36);

$$z_i^\xi = \frac{\sum_{k=1}^{|\theta_i|} \theta_i(k) s_{\theta_i(k)}}{\sum_{k=1}^{|\theta_i|} s_{\theta_i(k)}} \quad (5.35)$$

$$R_i^\xi = \frac{1}{|\theta_i| - 1} \sum_{k=1}^{|\theta_i|} [\theta_i(k) - z_i^\xi][\theta_i(k) - z_i^\xi]^T \quad (5.36)$$

where, $\theta_i(k)$ and $s_{\theta_i(k)}$ denote the 2D position and the score of the k^{th} member of the measurement set θ_i , respectively. Continuing with the example illustrated in Fig. 5.6, six measurements are generated from the set Υ_1^ξ . In Fig. 5.7, the centers of the clusters are represented by the measurements z_i^ξ s, and the ellipses surrounding them depict the 99% uncertainty regions, which are calculated from the covariance matrices R_i^ξ s.

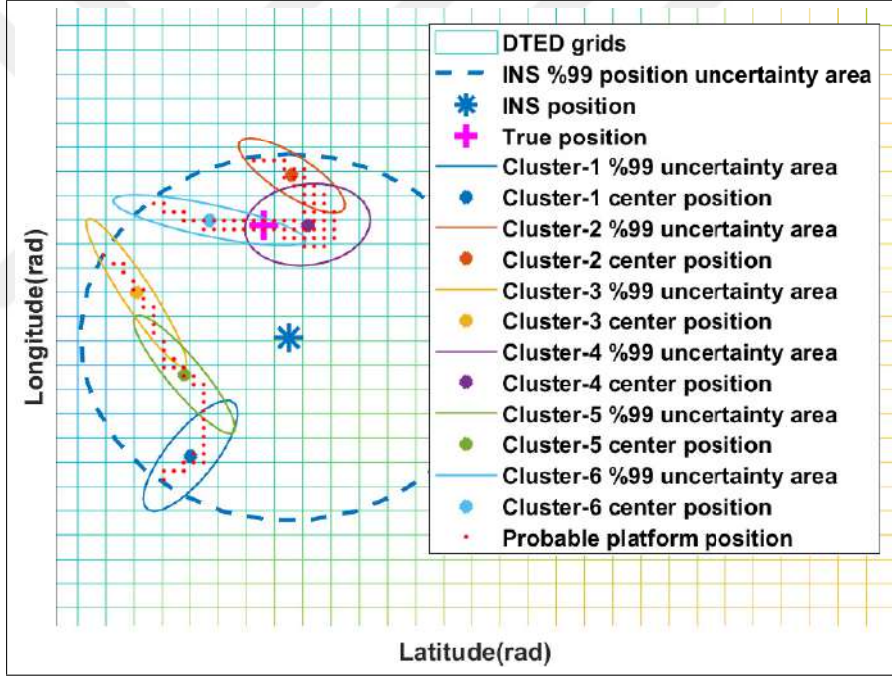


Figure 5.7: Display of measurements and their uncertainty ellipses.

5.2.3.1. Hypothesis Set Correction

After the generation of measurements, the next step involves correcting the hypotheses with appropriate measurements. Let $\Gamma(t_{Ra(k)})$ represent the collection of tuples, each comprising a measurement and its corresponding covariance. These tuples are derived

from all hypotheses at the time $t_{Ra(k)}$, as described below:

$$\Gamma(t_{Ra(k)}) = \left\{ (z_i^\xi, R_i^\xi) | i \in \{1, \dots, K^\xi\}, \text{ for all } \xi \in \Xi^-(t_{Ra(k)}) \right\} \quad (5.37)$$

After the creation of the set $\Gamma(t_{Ra(k)})$, a corrected hypothesis set, denoted as $\Xi^+(t_{Ra(k)})$, is formed. This is achieved by fusing the hypotheses in $\Xi^-(t_{Ra(k)})$ with suitable elements from $\Gamma(t_{Ra(k)})$. It is important to note that this fusion process may lead to a change in the number of hypotheses in the set, indicated by $|\Xi^-(t_{Ra(k)})| \neq |\Xi^+(t_{Ra(k)})|$. The specific rule used for associating measurements to hypotheses is outlined in (5.38). The purpose of employing this particular association rule is to inhibit exponential growth in the number of newly generated hypotheses.

$$\ell(\xi, \gamma) = \begin{cases} 1 & (z^\gamma - \hat{p}_{l,L}^\xi)^T (P_{l,L}^\xi)^{-1} (z^\gamma - \hat{p}_{l,L}^\xi) < \chi_{0.99}^2, \\ & \xi \in \Xi^-, \gamma \in \Gamma \\ 0 & \text{otherwise} \end{cases} \quad (5.38)$$

Let $\Omega(t_{Ra(k)})$ be defined as the set of tuples, where each tuple consists of a hypothesis ξ from Ξ^- and its corresponding associated measurement γ from Γ , as outlined in

$$\Omega(t_{Ra(k)}) = \{(\xi, \gamma) | \xi \in \Xi^-(t_{Ra(k)}), \gamma \in \Gamma(t_{Ra(k)}), \ell(\xi, \gamma) = 1\}$$

After the formation of the association set $\Omega(t_{Ra(k)})$, a closed-loop integration process is performed for each pair of hypothesis ξ and measurement γ within the set $\Omega(t_{Ra(k)})$. The details of this integration procedure are outlined in Algorithm 4. In Algorithm 4, ξ^+ denotes the *corrected* hypothesis. In line 9 of the algorithm, $\hat{C}_{NB}^\xi$ represents the DCM, which portrays the attitude of the hypothesis ξ . The function $DS(\cdot)$ constructs a matrix where estimates of scale factor errors are positioned on the diagonal entries, and estimates of misalignment errors are placed on the off-diagonal entries, as demonstrated in (5.39). Furthermore, $(\hat{M}^{-1})^\xi$ is the inverse of the matrix defined in (5.39) for the specific hypothesis ξ .

$$M_a = \begin{bmatrix} \delta f_{SF_x} & \delta f_{MA_{xy}} & \delta f_{MA_{xz}} \\ \delta f_{MA_{yx}} & \delta f_{SF_y} & \delta f_{MA_{yz}} \\ \delta f_{MA_{zx}} & \delta f_{MA_{zy}} & \delta f_{SF_z} \end{bmatrix} \quad (5.39)$$

Algorithm 4 Hypothesis Set Correction

Require: $\Omega(t_{Ra(k)})$ **Ensure:** $\Xi^+(t_{Ra(k)})$

- 1: $\Xi^+(t_{Ra(k)}) = \emptyset$
 - 2: **for** for each $(\xi, \gamma) \in \Omega(t_{Ra(k)})$ **do**
 - 3: $\delta \mathbf{z} = \hat{\mathbf{p}}_{l,L}^\xi - \mathbf{z}^\gamma$
 - 4: $K_{\text{gain}} = P^\xi H^T (H P^\xi H^T + R^\gamma)^{-1}$
 - 5: $\delta \hat{\mathbf{X}}^{\xi+} = \delta \hat{\mathbf{X}}^\xi + K_{\text{gain}}(\delta \mathbf{z} - H \delta \hat{\mathbf{X}}^\xi)$
 - 6: $P^{\xi+} = (I - K_{\text{gain}} H) P^\xi$
 - 7: $\hat{\mathbf{p}}_{l,L}^{\xi+} = \hat{\mathbf{p}}_{l,L}^\xi - \delta \hat{\mathbf{X}}_{l,L}^{\xi+}$
 - 8: $\hat{\mathbf{v}}_{N,E}^{\xi+} = \hat{\mathbf{v}}_{N,E}^\xi - \delta \hat{\mathbf{X}}_{v_N, v_E}^{\xi+}$
 - 9: $\hat{\mathbf{C}}_{\text{NB}}^{\xi+} = \left(I_{3 \times 3} - S(\delta \hat{\mathbf{X}}_{r,p,y}^{\xi+}) \right)^{-1} \hat{\mathbf{C}}_{\text{NB}}^\xi$
 - 10: $\hat{\mathbf{f}}_{\text{bias}}^{\xi+} = \hat{\mathbf{f}}_{\text{bias}}^\xi + \delta \hat{\mathbf{X}}_{f_{\text{BR}}}^{\xi+} + \delta \hat{\mathbf{X}}_{f_{\text{BI}}}^{\xi+}$
 - 11: $\hat{\omega}_{\text{bias}}^{\xi+} = \hat{\omega}_{\text{bias}}^\xi + \delta \hat{\mathbf{X}}_{\omega_{\text{BR}}}^{\xi+} + \delta \hat{\mathbf{X}}_{\omega_{\text{BI}}}^{\xi+}$
 - 12: $\hat{M}_{\text{acc}}^{-1 \xi+} = \left(I_{3 \times 3} - DS(\delta \hat{\mathbf{X}}_{a_{\text{SF}}}^{\xi+}, \delta \hat{\mathbf{X}}_{a_{\text{MA}}}^{\xi+}) \right) \hat{M}_{\text{acc}}^{-1 \xi}$
 - 13: $\hat{M}_{\text{gyr}}^{-1 \xi+} = \left(I_{3 \times 3} - DS(\delta \hat{\mathbf{X}}_{g_{\text{SF}}}^{\xi+}, \delta \hat{\mathbf{X}}_{g_{\text{MA}}}^{\xi+}) \right) \hat{M}_{\text{gyr}}^{-1 \xi}$
 - 14: set $\delta \hat{\mathbf{X}}^{\xi+} = \mathbf{0}$
 - 15: append ξ^+ into $\Xi^+(t_{Ra(k)})$
 - 16: **end for**
-

5.2.4. Hypothesis Pruning and Merging

Managing the increasing number of hypotheses is a significant computational challenge in our method. Therefore, it is crucial to discard improbable hypotheses and unite the similar ones in each terrain aid cycle. This step is vital to maintain practicality of the algorithm for real-time applications. Following the hypothesis set correction step, we engage in a merging process. In this process, we employ the Cauchy–Schwarz divergence (CSD), as referenced in [70], to compute dissimilarity scores between various hypotheses. Let $D[i, j]$ represent the CSD between hypotheses i and j , and let t_m be the threshold for merging. We opt for CSD owing to its symmetry and non-negativity characteristics, where $D[i, j] = D[j, i]$ and $D[i, j] \geq 0$ for any pair of hypotheses i, j . Hypotheses i and j are merged when $D[i, j] < t_m$. The calculation of $D[i, j]$ is as

follows:

$$D[i, j] = \frac{1}{4} \log \left| \frac{\Sigma_i}{2} \right| + \frac{1}{2} \mu_i^T \Sigma_i^{-1} \mu_i + \frac{1}{4} \log \left| \frac{\Sigma_j}{2} \right| + \frac{1}{2} \mu_j^T \Sigma_j^{-1} \mu_j + \frac{1}{2} \log |\Sigma_j^{-1} + \Sigma_i^{-1}| - \frac{1}{2} (\Sigma_i^{-1} \mu_i + \Sigma_j^{-1} \mu_j)^T \cdot (\Sigma_i^{-1} + \Sigma_j^{-1})^{-1} \cdot (\Sigma_i^{-1} \mu_i + \Sigma_j^{-1} \mu_j) \quad (5.40)$$

In (5.40), $|\cdot|$ represents the determinant operator, μ_i refers to the 9×1 vector composed of PVA values of hypothesis i , while Σ_i is the corresponding 9×9 PVA error covariance matrix of hypothesis i . When two or more hypotheses are regarded as suitable for merging, the mean and covariance of the resultant merged hypothesis are calculated using Salmond's mixture merging framework, as detailed in [71]. The mean and covariance of the merged hypothesis, as outlined in (5.41) and (5.42), are computed for all navigation properties. It is assumed that the weights of the hypotheses, denoted as w_k , are equal to $\frac{1}{N}$ for all k .

$$\mu_{merged} = \sum_{k=1}^N w_k \mu_k \quad (5.41)$$

$$P_{merged} = \sum_{k=1}^N w_k [P_k + (\mu_k - \mu_{merged})(\mu_k - \mu_{merged})^T] \quad (5.42)$$

Figure 5.8 presents a comparison between the original ensemble of six distinct hypotheses and the consolidated set resulting from the merging process. This process is based on the criterion of hypothesis pairs having a CSD of less than 2. The illustration shows the original set of hypotheses on the left and the merged set on the right. During this process, Hypotheses 4 and 5 on the left are combined to form Hypothesis 2 on the right, while Hypotheses 1, 2, 3, and 6 on the left merge to create the new Hypothesis 1 on the right.

During the pruning phase, which takes place at the beginning of the hypothesis correction stage, we allocate a binary variable $\alpha^\xi(t_{Ra(k)})$ belonging to $\{0(\text{discarded}), 1(\text{retained})\}$. This variable signifies the current status of the hypothesis ξ . The assignment of $\alpha^\xi(t_{Ra(k)})$ is determined as outlined in (5.43).

$$\alpha^\xi(t_{Ra(k)}) = \begin{cases} 0 & \text{if } |\Upsilon_1^\xi| = 0 \text{ and } |\Xi^-(t_{Ra(k)})| > 1 \\ 1 & \text{otherwise} \end{cases} \quad (5.43)$$

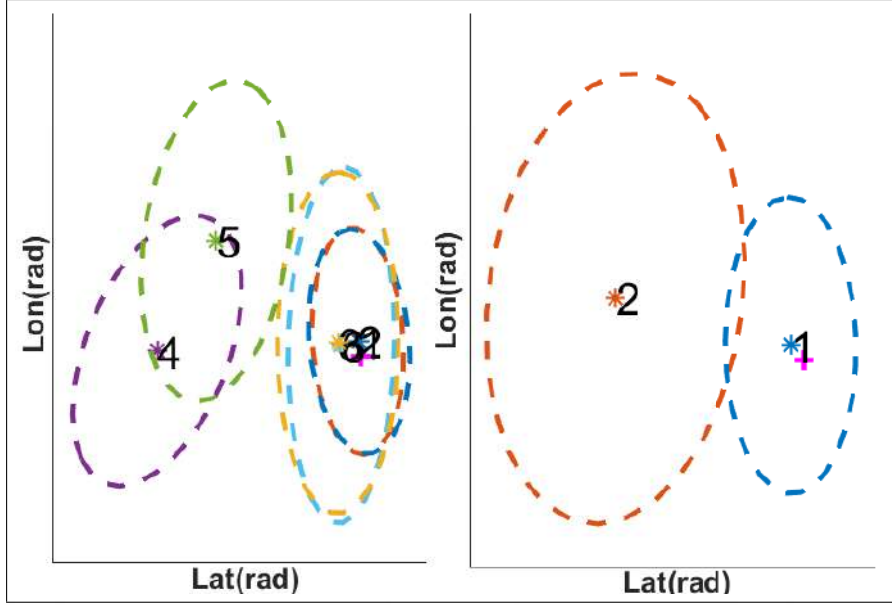


Figure 5.8: 2D illustration of merging: before(left), after(right)

The initial condition in (5.43) suggests that the hypothesis ξ is eliminated if every member location derived from its gridding is deemed improbable and there exists at least one other hypothesis in $\Xi^-(t_{R(k)})$ apart from ξ . Fig. 5.9 illustrates an example of hypothesis pruning and merging across three successive terrain aid cycles. Black lines separate hypothesis sets for different time intervals. For instance, at the initial hypothesis set correction moment $t_{Ra(1)}$, seven emerging hypotheses are positioned variably. These hypotheses are then carried forward until the next hypothesis set correction time $t_{Ra(2)}$. After $t_{Ra(2)}$, the hypotheses begin to cluster around two specific areas. The subsequent hypothesis set correction at $t_{Ra(3)}$ clarifies the ambiguity between these two positions, leading to a concentration of hypotheses around the actual platform location.

5.2.4.1. Ad-Hoc Measures to Prevent Divergence

Unlike GNSS measurements, which are independent of the INS solution, in our approach, the generated position measurements are intrinsically linked to the hypotheses. If the actual platform position falls outside the gridding area of all hypotheses, these hypotheses tend to converge wrong positions and ultimately become invalid. This issue typically arises due to the following reasons:

Firstly, the horizontal position error covariances of the hypotheses reach steady state

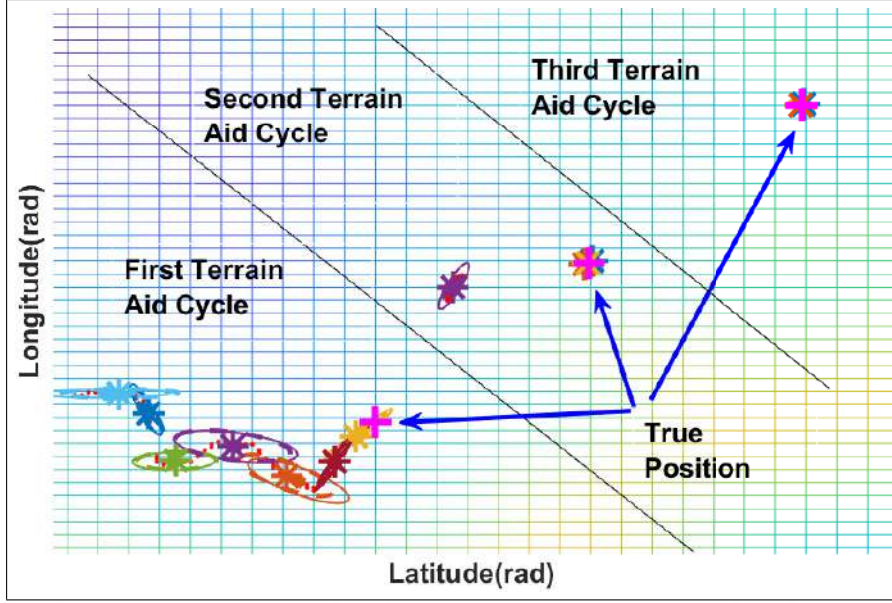


Figure 5.9: The demonstration of hypothesis merging/pruning.

values after several correction steps. Consequently, the gridding regions of these hypotheses may slowly shift away from the true position, especially in areas with gradual terrain height changes. Secondly, the error covariance propagation step, as outlined in Algorithm-3 line 10, assumes realistic error accumulation under small error assumptions, as discussed in [2]. Therefore, if the initial velocity and attitude errors are large, their impact on the growth of position error through propagation may not accurately reflect the actual error increase, leading to the true position being outside the gridding boundaries of all hypotheses.

To mitigate the divergence issue caused by these factors, we propose two ad-hoc methods: *covariance freezing* and *covariance expansion*. In covariance freezing, we adjust the horizontal position and velocity error covariances under specific conditions. The position error covariance is manipulated to ensure that the area of each hypothesis's gridding region on the local level frame remains larger than that of a circle with radius r_{low} , while maintaining the direction of the principal axes. Similarly, if the square root of the sum of the standard deviations of north and east velocity errors is less than the v_{low} parameter, the horizontal velocity error variances are set to v_{low}^2 . In the case of covariance expansion, we increase the position error covariances of all viable hypotheses by scaling their eigenvalues with a factor s_{cof} in scenarios where no valid measurement is found within the gridding regions of all active hypotheses.

The details of these divergence prevention measures are elaborated in Algorithms 5 and 6. In these algorithms, σ_n^ξ and σ_e^ξ are the square roots of the diagonal elements of the horizontal position error covariance matrix $P_{l,L}^\xi$. e_1^ξ and e_2^ξ are the eigenvalues of $P_{l,L}^\xi$, while U^ξ and V^ξ are its left and right singular matrices, respectively. The term $e_{\max}^\xi$ equals the maximum of e_1^ξ and e_2^ξ . The parameters s_{cof} and r_{low} refer to the uncertainty scale coefficient and the approximate minimum radius for position coverage, respectively.

Algorithm 5 Covariance Freezing

Require: $\Xi^+(t_{Ra(k)}), r_{\text{low}}, v_{\text{low}}$

Ensure: $\Xi^+(t_{Ra(k)})$

```

1: for all  $\xi \in \Xi^+(t_{Ra(k)})$  do
2:   if  $(1.5(\sigma_n^\xi + \sigma_e^\xi)r_e)^2 < r_{\text{low}}^2$  then
3:      $\kappa = \frac{r_{\text{low}}^2}{(1.5(\sigma_n^\xi + \sigma_e^\xi)r_e)^2}$ 
4:      $P_{l,L}^\xi \leftarrow U^\xi \begin{bmatrix} \kappa e_1^\xi & 0 \\ 0 & \kappa e_2^\xi \end{bmatrix} V^{\xi T}$ 
5:   end if
6:   if  $\left(\sqrt{P_{V_N}^\xi + P_{V_E}^\xi}\right) < v_{\text{low}}$  then
7:      $P_{V_N, V_E}^\xi \leftarrow \begin{bmatrix} v_{\text{low}}^2 & 0 \\ 0 & v_{\text{low}}^2 \end{bmatrix}$ 
8:   end if
9: end for
10: return  $\Xi^+(t_{Ra(k)})$ 

```

Algorithm 6 Covariance Expansion

Require: $\Xi^-(t_{Ra(k)}), s_{\text{cof}}$

Ensure: $\Xi^-(t_{Ra(k)})$

```

   while  $\alpha^\xi(t_{Ra(k)}) = 0$  for all  $\xi \in \Xi^-(t_{Ra(k)})$  do
2:    $P_{l,L}^\xi \leftarrow s_{\text{cof}} \begin{bmatrix} e_{\max}^\xi & 0 \\ 0 & e_{\max}^\xi \end{bmatrix}$ 
   end while
4: return  $\Xi^-(t_{Ra(k)})$ 

```

Figure 5.10 illustrates a scenario where the platform's true position is lost during straight flight over a terrain with minimal elevation change, but is subsequently reacquired after carrying out covariance expansion technique. In this specific case, the platform navigates from the upper right to the lower left quadrant. The platform's actual location history are denoted by a purple '+' symbols, whereas the red dots and ellipses symbolize the probable measurements and hypothesis regions derived from the gridding process, respectively. As the platform approaches the area marked by the green circle, it starts to acquire measurements corresponding to the blue hypothesis areas, despite drifting from its true location. This divergence is attributed to the negligible variation in terrain elevation beneath the platform compared to that within the gridding region of the hypothesis. However, when the true position intersects the green circle, a significant difference is observed between the altimeter-derived terrain height and the calculated heights within the hypothesis's gridding region. This leads to the absence of horizontal measurements, potentially causing track loss unless further actions are taken. Executing covariance expansion at this point enlarges the position error covariances for viable hypotheses. This expansion facilitates the generation of horizontal measurements, which in turn results in the true position lying within the extended gridding region of the hypothesis. Notably, the covariance expansion process may also lead to the emergence of false hypotheses, as demonstrated in Fig. 5.10. Nevertheless, depending on the topographical context, these false hypotheses typically dissipate after several correction cycles.

5.2.5. Hybrid Output

Each living hypothesis at any given moment represents a potential solution in our approach. From the perspective of a platform user, the presence of multiple hypotheses related to PVA information may not be practical. Instead, users often prefer a singular, consolidated PVA output. For this reason, we designate an output hypothesis, which is obtained by merging all living hypotheses at that time instance using (5.41) and (5.42).

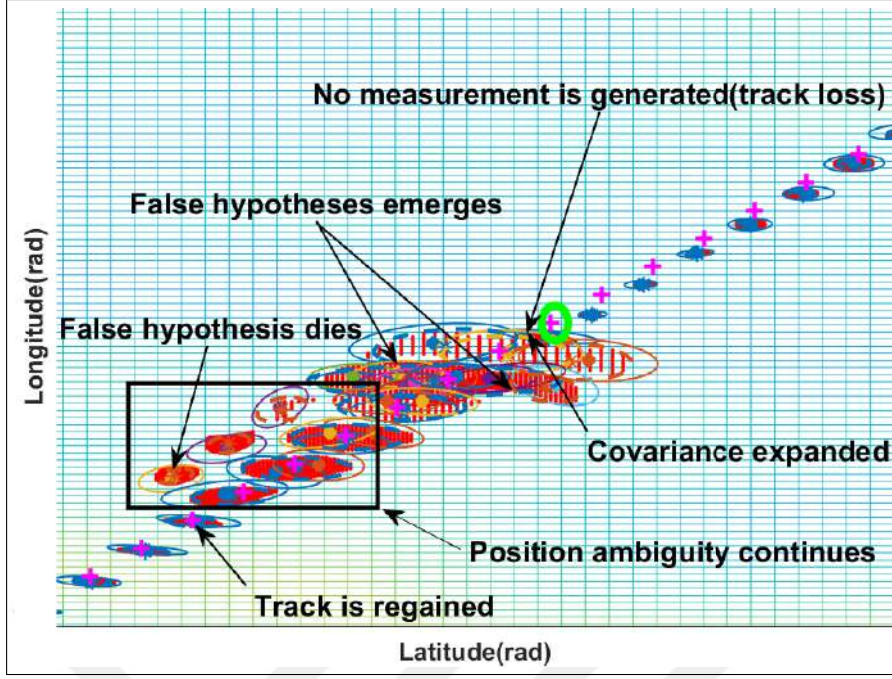


Figure 5.10: Visualization of covariance expansion effect.

5.2.6. Simulation Results

To evaluate our TAN approach, we initially design a flight path as depicted in Fig.5.11 and generate its corresponding IMU data following the steps in Section. Subsequently, distorted IMU measurements are derived using equations given in the subsection 3.3. The flight spans 36 minutes, featuring sequences of straight-level flights followed by coordinated turns exerting 1.5 G. The aircraft maintains a constant velocity of 250 m/s, and the generation of data excludes considerations of side-slip and angle of attack. For terrain data, we use real level-1 DTED covering a rectangular geographic region from $17^{\circ}11'19''$ to $18^{\circ}58'27''$ in latitude, and $98^{\circ}32'55''$ to $99^{\circ}41'40''$ in longitude. In our simulation, Micro-Electro-Mechanical Systems (MEMS) grade IMU error specifications are applied to distort the authentic IMU data. To assess the consistency of the proposed filter, an offline Monte Carlo test [13] is used, wherein the normalized estimation error squared (NEES) for only the horizontal position states is evaluated at each terrain aid cycle. The Monte Carlo test is performed with 100 iterations. The gyroscope error specifications have the following standard deviation values: bias repeatability error of $7^{\circ}/hour$, scale factor error of 400 ppm, misalignment error of 0.5 mrad, bias instability error of $0.3^{\circ}/hr$, and angle random walk error of $0.15^{\circ}/\sqrt{hour}$. Additionally,

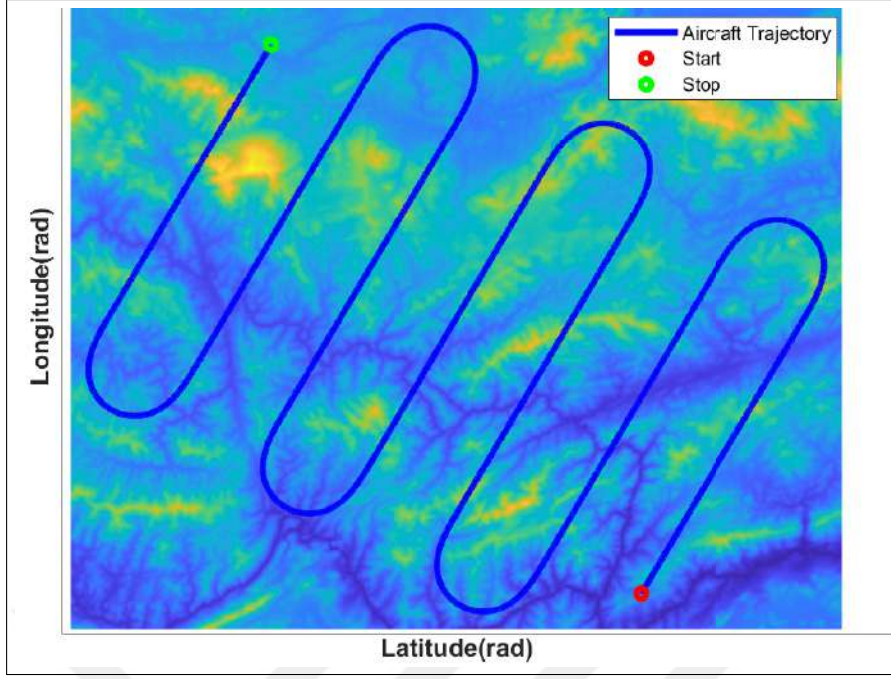


Figure 5.11: The bird's eye view of flight path.

the accelerometer error specifications include a bias repeatability error of 2 mg, scale factor error of 200 ppm, misalignment error of 0.5 mrad, bias instability error of 0.04 mg, and velocity random walk error of $0.03 \text{ m/s}/\sqrt{\text{hour}}$. The measured terrain height error is presumed to be a zero-mean Gaussian with a standard deviation of 15 meters. Initial position, velocity, and heading error standard deviations are set at 2km for both north and east directions, 10 m/s for both north and east directions, and 200 mrad for the heading angle. The terrain aid cycle frequency is set to 0.5 Hz.

Algorithm parameters, determined through empirical results, are set as follows: $\tau_s = 0.01$, $r_c = r_{low} = 100$ meters, $n_c = 40$, $s_{cof} = 3$, and a hypothesis merging distance threshold of 1.8. The parameter d_g , unique to each hypothesis, is set as $d_g^\xi = 0.06(\sigma_n^\xi + \sigma_e^\xi)r_e^{-1}$. Fig. 5.12 shows the result of the NEES test statistic, with the dashed line representing the upper bound of the confidence limit 95%. Fig. 5.12 illustrates that, except for a few instances, the filter is consistent as the computed NEES remains below the 95% confidence threshold. This indicates that the horizontal position errors are consistent with the filter's position covariance outputs. Across all Monte Carlo simulations, the average absolute errors in northing and easting are approximately 100 meters, which is also equal to the r_c parameter. Figures 5.13 and 5.14 illustrate the estimated horizontal velocity and heading angle against their actual values. Although

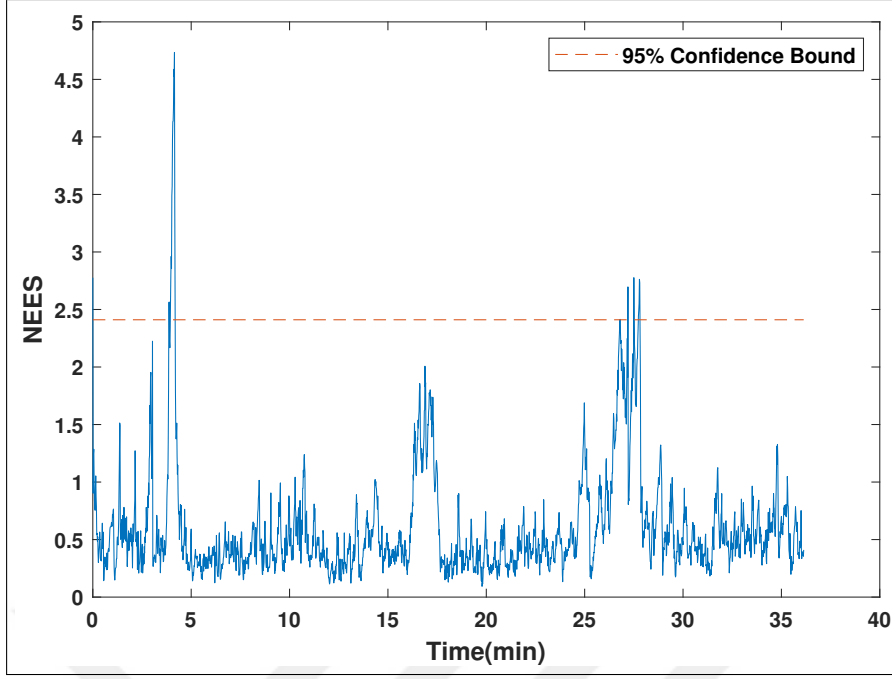


Figure 5.12: NEES plot.

the error in east velocity estimation decreases in the initial phase of the flight, the heading error consistently remains around 120 mrad during straight flights, reducing to about 15 mrad during banking maneuvers. Figure 5.15 displays the average number of hypotheses (ANH) in all Monte Carlo simulations in two separate plots. The upper subplot reveals a rapid increase in ANH to approximately 20 within the initial terrain aid cycles in the first 3 seconds, followed by a decrease due to the pruning and merging of hypotheses. The lower subplot, representing the period after the first minute, shows the ANH is around around one, with minor increases during certain flight segments due to measurement uncertainties caused by terrain symmetry or flatness. This pattern suggests that the proposed method effectively adjusts the hypothesis count and maintains accurate tracking despite encountering topographical similarities or flatness.

5.2.7. Discussion

In this study, we present a novel TAN method that employs multiple hypotheses. This multi-hypothesis approach demonstrates robustness against large initial errors in position, velocity, and heading, unlike traditional TAN methods. Our proposed framework adjusts the number of hypotheses based on terrain topography, increasing them over non-

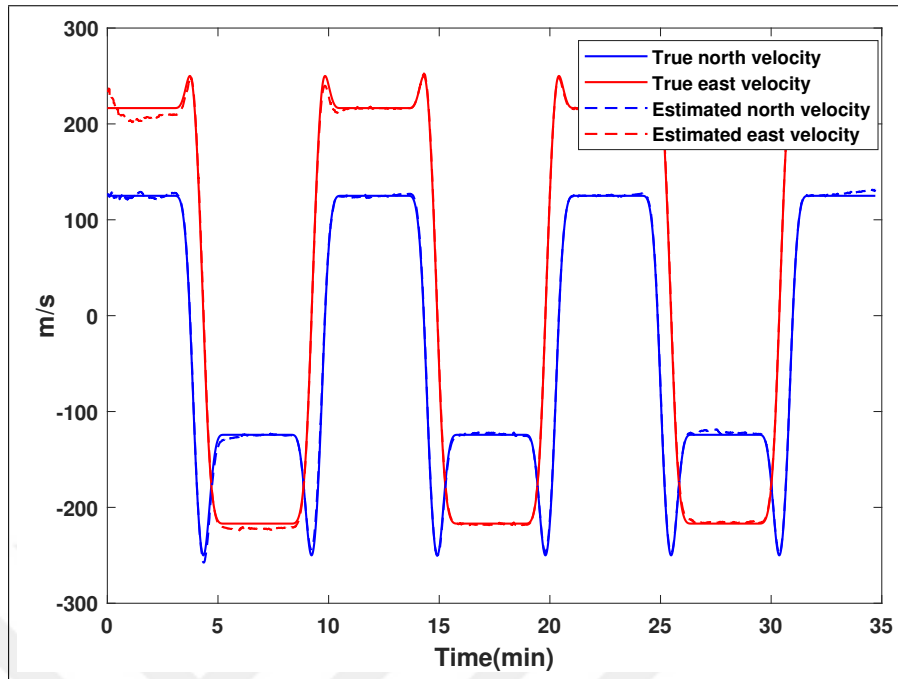


Figure 5.13: Single-run true velocity versus estimated velocity.

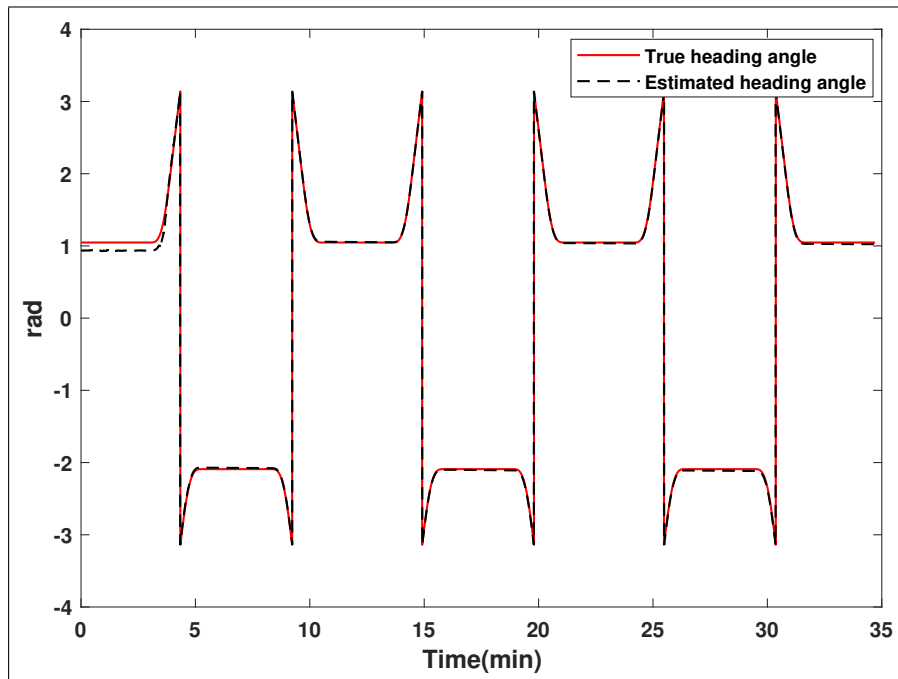


Figure 5.14: Single-run true heading versus estimated heading.

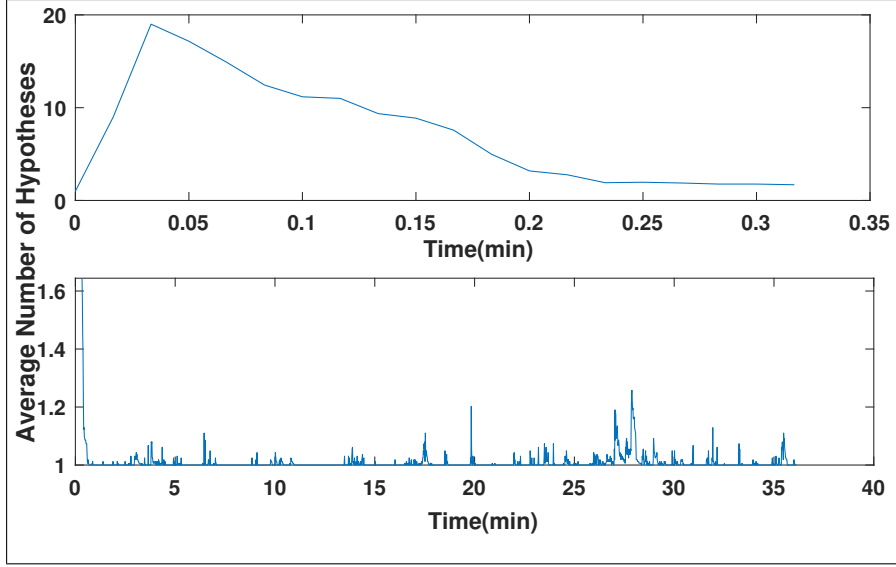


Figure 5.15: Average number of hypotheses.

informative or repetitive terrains and decreasing them once the ambiguity is resolved. However, a limitation of our method is that, positioning error reduces to r_{low} value and fluctuates around this value as we do not want position error covariance to be too small to divert divergence. We observed that reducing this r_{low} limit enhances accuracy for high-quality radar/laser altimeters and distinct terrains. However, decreasing r_{low} also increases the likelihood of losing the true position, in other words, it may lead to divergence. The root cause of this divergence versus low accuracy tradeoff stems from the fact that measurement covariances are not evaluated from an independent source, but are directly derived from the hypotheses themselves. One possible solution could be to make the r_{low} parameter adaptive to the terrain, where it becomes smaller for regions with large height changes and larger for areas that are relatively flat with minimal height variations.

6. VISUAL AIDED NAVIGATION

In image-based navigation systems, the objective is to determine the absolute location or self-motion of the vehicle by utilizing the visual data captured by a camera mounted on the platform. Recently, the accessibility of small-sized, cost-effective cameras has led to the prominence of vision-based and vision-aided localization techniques. These techniques are increasingly being adopted either as alternatives or as complementary solutions to the traditional GNSS aided inertial navigation. Examining the literature reveals that visual navigation methodologies primarily branch into two categories: Relative Visual Positioning (RVP) and Absolute Visual Positioning (AVP). In the subsequent part of this section, methods related to RVP are briefly discussed, although not extensively, as RVP falls outside the scope of this dissertation. Instead, the primary focus is on the AVP problem within the context of Visual Aided Navigation (VAN). As mentioned in Subsection 6.2, visual feature extraction and matching play a critical role in addressing the AVP problem. Accordingly, various visual feature descriptors have been evaluated, and their positioning performances have been compared. In the simulations, real orthophoto data is utilized, and images captured from a simulated trajectory are synthetically generated using the pinhole camera model.

6.1. Relative Visual Positioning

RVP includes methodologies such as Visual Odometry (VO) and Simultaneous Localization and Mapping (SLAM). In VO, successive images captured from a moving vehicle are analyzed to estimate the platform's self-motion and facilitate the 3D reconstruction of surrounding structures [72]. The initial studies in VO were driven by NASA's Mars exploration program, focusing on enabling planetary rovers to determine their six-degree-of-freedom (DoF) movement, particularly in challenging terrains characterized by uneven surfaces and wheel slippage [73, 74]. Most studies on VO have predominantly employed stereo cameras since using a monocular camera limits motion recovery to a scale factor. L. Matthies and S. Shafer emphasize a crucial problem in stereo navigation systems that depend exclusively on camera ego-motion estimates, revealing how these systems accumulate errors at a rate that grows super-linearly with

the distance traveled, primarily due to growing orientation errors. To counteract this, the authors introduce a robust estimation technique specifically designed for long-distance rover navigation [75]. Lacroix *et al.* [76] also adopted a stereo visual odometry approach for planetary rovers. The distinct aspect of their implementation was in the selection of key points. Contrary to employing the Forstner detector, they utilized dense stereo and then chose potential key points by examining the peaks of the correlation function. Milella [77] proposed an algorithm that enhances the precision and efficiency of real-time ego-motion estimation by integrating image intensity information with 3D stereo data into the Iterative Closest Point (ICP) framework.

There are also a large number of studies that try to address VO problem using monocular cameras. The key distinction in monocular VO compared to the stereo approach is that it necessitates computing both the relative motion and the 3D structure exclusively from 2D image data. Nister *et al.* presents a method for estimating the motion of a stereo head or a single moving camera on ground vehicles. It employs a feature tracker as the front end and a geometric hypothesize-and-test architecture to produce accurate motion estimates from visual input alone, without the need for prior knowledge of the scene or the motion [78]. Two novel methods of real-time VO specifically designed for planetary rovers. The first method utilizes a robust approach to estimate optical flow, followed by its integration, while the second method involves a comprehensive structure-from-motion solution. Both methods employ omnidirectional imagery to accurately estimate the motion of the rover [79]. Tardif *et al.* concentrate on urban environments, implementing a monocular VO system that emphasizes incremental trajectory estimation without a motion model, suitable for cluttered urban scenes [80]. The main drawback of monocular VO is the scale ambiguity. This issue arises because, with a single camera, it is impossible to determine the absolute scale of the scene or the distance the camera has moved based solely on the images. This is because the same image could be produced by a small object close to the camera or a large object far away. Consequently, monocular VO can only provide motion estimates up to an unknown scale factor. There are several approaches commonly used to address this issue. In [81], authors proposed a method that analyzes human motion using inertial data, such as step length estimation, to dynamically resolve scale ambiguity. In [82], a framework is proposed that leverages accurate ground plane estimation by assuming constant camera

height above the ground and scale recovery is achieved through a robust least-squares solver based on RANSAC.

6.2. Absolute Visual Positioning

The primary benefit of AVP lies in its built-in resistance to gradual position shifts over time. This technique relies on pre-existing data (referred to as reference data) for determining the UAV's location. This data must be accurately georeferenced before it is employed for localization purposes. The reference data might consist of aerial imagery, either as individual images or as a merged mosaic.



Figure 6.1: A satellite image illustration.

The goal of AVP is to determine the UAV's position by comparing the images captured by the UAV's ground-facing camera with the georeferenced images in the database. The key challenges associated with this approach include the substantial data requirements and the complexity of matching or registering images obtained from diverse sources. These images are often acquired using various sensors, at different time intervals, under varying lighting conditions, and with different resolutions. A single reference map or reference image set may encompass images originating from multiple sources [83].

In literature, methods based on template matching represent one of the most fundamental techniques for identifying the position of an image patch within a larger reference image.

This approach employs an image patch comparison operator, like the sum of squared differences (SSD), to evaluate and ascertain the degree of similarity between two image patches. But, template matching is a process that requires significant computational resources. Therefore, without modifications to reduce the search space, its application in real-time systems is not feasible [84, 85].

Similar to the methods used in RVP, the literature on AVP also frequently employs feature point matching approaches. Feature point matching consists of two primary phases: Firstly, the extraction of feature points along with their descriptors from both the reference image and the image to be registered. Secondly, establishing the connections between the extracted feature points of the two images. In the field of computer vision, a diverse range of feature detector algorithms exists. These include corner detectors such as Forstner, Harris, Moravec, Shi-Tomasi, and Features from Accelerated Segment Test (FAST), as well as blob detectors like Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and Center Surround Extremas (CENSURE). Each of these algorithms has its own set of advantages and disadvantages when compared to the others. Fraundorfer presented the Table 6.1 to compare these algorithms across various aspects [1].

Table 6.1: Comparison of feature point detectors. C.D.: *Corner Detector*, B.D.: *Blob Detector*, R.I.: *Rotation Invariant*, S.I.: *Scale Invariant*, A.I.: *Affine Invariant*, Rep: *Reproducibility*, Loc. Accu.: *Localization Accuracy*, Rob.: *Robustness*, Eff.: *Efficiency*, courtesy of Fraundorfer & Scaramuzza [1].

Detector	C.D.	B.D.	R.I.	S.I.	A.I.	Rep.	Loc. Accu.	Rob.	Eff.
Harris	x					+++	+++	++	++
Shi-Tomasi	x					+++	+++	++	++
FAST	x		x			++	++	++	++++
SIFT	x	x	x	x		+++	++	+++	+
SURF	x	x	x	x	x	+++	++	++	++
CENSURE	x	x	x	x	x	+++	++	+++	+++

In Fig. 6.2, key points identified by four distinct feature extraction algorithms are illustrated on an aerial image of an urban area.

Features, once extracted, are paired utilizing matching algorithms such as Brute force



(a) Fast features.



(b) Harris features.



(c) Shi-Tomasi features.



(d) SIFT features.

Figure 6.2: Examples of key feature detectors in aerial images.

[86], FLANN [87], and best-bin-first search [88]. Provided that a minimum of six image points are accurately registered in the reference image, with known world coordinates for their pixel locations, it becomes feasible to estimate the camera's position and orientation with respect to the world coordinate frame. The subsequent part of this section primarily concentrates on this estimation technique.

6.3. Pinhole Camera Model & Projection Matrix

The pinhole camera model represents the most fundamental geometric abstraction, where light rays traversing through a singular point, known as the "pinhole", form an image on a plane located behind this point. The image is inversely and proportionally projected relative to the distance from the pinhole. This model is crucial for comprehending the correlation between three-dimensional points in the real world and their two-dimensional depictions in an image.

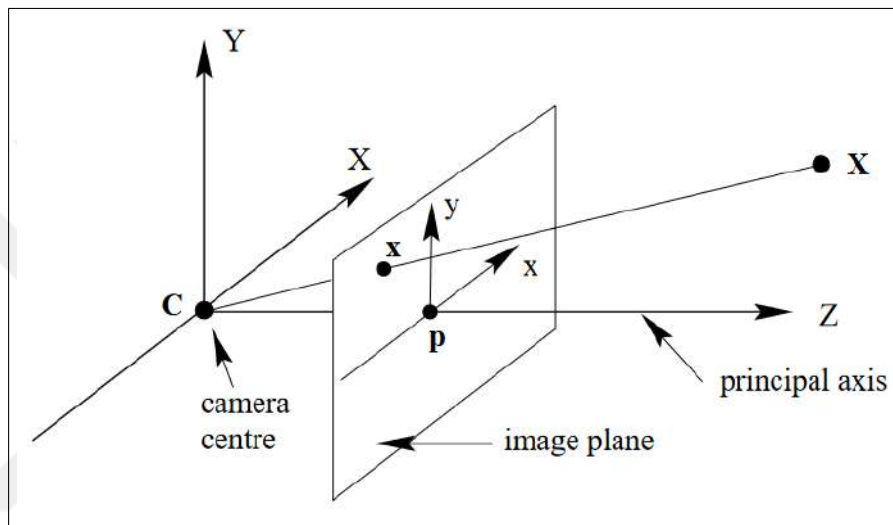


Figure 6.3: Pinhole camera model (3D).

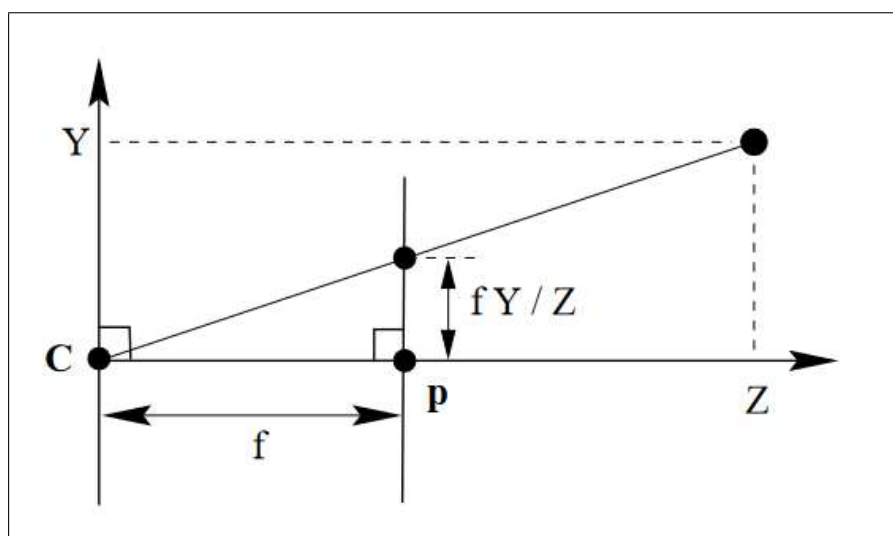


Figure 6.4: Pinhole camera model (side view).

In Fig. 6.3 and Fig.6.4 pinhole camera model is illustrated. The image plane is positioned behind the camera center, contrary to the real-world configuration where it is typically in front of the pinhole. In this model, the image plane is where the 2D projection of the 3D world is formed. The 2D point x' , which is the projection of the 3D point X onto the image plane, is inversely proportional to the distance of the point from the pinhole, causing the image to be upside down and reversed left to right on the image plane. The point $(X, Y, Z)^T$ is mapped onto the point $(f\frac{X}{Z}, f\frac{Y}{Z}, f)^T$ on the image plane. This represents a transformation from the 3D Euclidean space $\mathbb{R}^3$ to the 2D Euclidean space $\mathbb{R}^2$, where f is the focal length of the pinhole camera, which scales the coordinates relative to the depth Z . To establish a linear mapping between world coordinates and image plane coordinates in the pinhole camera model, points are typically expressed in homogeneous coordinates. This mathematical representation extends the conventional Cartesian coordinates by adding an extra dimension [89].

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6.1)$$

In (6.1), the 3×4 matrix, commonly referred to as the *camera projection matrix*, is adapted to include pixel mapping and a world coordinate frame differing from the one centered at the camera. Fig. 6.5 illustrates a more practical setup, where the camera frame is a translated and rotated version of the world coordinate frame. In this scenario, the camera projection matrix is defined as shown in (6.2).

$$P = K[R|\mathbf{t}] \quad (6.2)$$

where K denotes the camera calibration matrix containing the intrinsic camera parameters, while R is the DCM that defines the attitude between the camera frame and the world coordinate frame. $\mathbf{t}$ is the translation vector calculated by (6.4) where $\tilde{C}$ is the camera position vector in the world coordinate frame.

$$K = \begin{bmatrix} fm_x & 0 & m_x p_x \\ 0 & fm_y & m_y p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

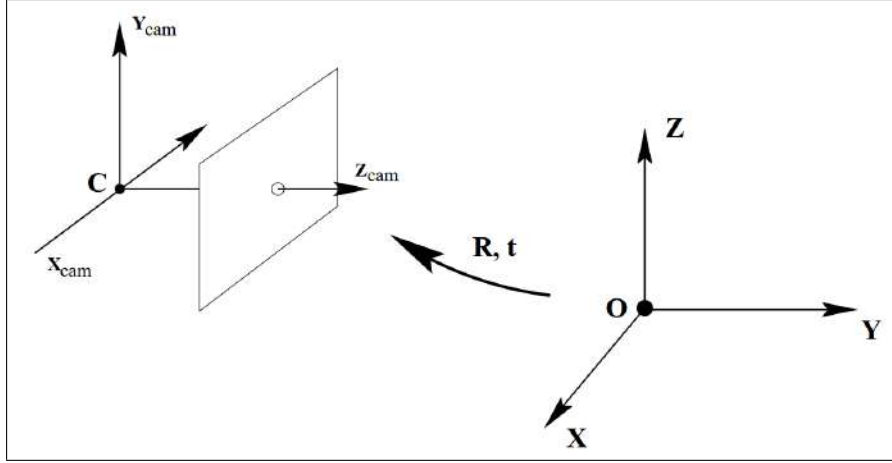


Figure 6.5: Generalized camera configuration.

$$t = -R\tilde{C} \quad (6.4)$$

In (6.3), fm_x and fm_y represent the camera's focal length expressed in pixel units along the x and y axes, respectively. $m_x p_x, m_y p_y$ principal point coordinates in terms of pixel units. The pixel location of any 3D point is determined by the linear mapping described by the subsequent expression.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6.5)$$

In (6.5), u and v represent the pixel coordinates of the 3D point, calculated by normalizing x and y with respect to w , such that $u = \frac{x}{w}$ and $v = \frac{y}{w}$.

6.4. Estimation of Camera Projection Matrix

Intrinsic (K) and extrinsic (R, t) parameters of the camera can be estimated if world coordinates of at least 6 image points are known. In this subsection we discuss the details of Direct Linear Transformation (DLT) approach [90]. A single 3D point is mapped to image coordinate as follows,

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6.6)$$

and u and v coordinates are;

$$u = \frac{x}{w} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \quad (6.7)$$

$$v = \frac{y}{w} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \quad (6.8)$$

Multiplying the denominators in equations (6.7) and (6.8) and then subtracting the left-hand sides from each side of these equations results in the matrix equation presented in (6.9).

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -uX & -uY & -uZ & -u \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -vX & -vY & -vZ & -v \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \quad (6.9)$$

The terms $\mathbf{p}_1^T$, $\mathbf{p}_2^T$, and $\mathbf{p}_3^T$ represent the transposes of each row of the camera matrix as detailed in (6.6). The 2×12 matrix described in (6.9) is formed using the information from a single correspondence between a 3D world point and its image point. By creating a 2×12 matrix for each point correspondence and stacking them vertically for n points, we construct a $2n \times 12$ matrix denoted as A . Our objective then becomes solving for p such that (6.10) is satisfied. Given that the matrix P comprises 12 elements and, when disregarding scale, has 11 degrees of freedom, it is essential to establish 11 equations to determine P . As each point correspondence contributes two equations, a minimum of 5 and a half such correspondences are necessary to solve for P . The "1/2" implies that only one equation from the sixth point is utilized, meaning that it's sufficient to know either the x-coordinate or the y-coordinate of the sixth image point.

$$Ap = \mathbf{0} \quad (6.10)$$

In real-world scenarios, precise 3D to 2D point correspondences may not always be available, and having multiple point correspondences results in an overdetermined system. To address this, a standard method involves selecting the eigenvector that

corresponds to the smallest eigenvalue of the matrix A . This eigenvector is regarded as the optimal approximation of the vector $\mathbf{p}$.

6.4.1. Decomposition of $\mathbf{P}$

Once the camera projection matrix P is estimated, it can be decomposed into the intrinsic matrix K , the rotation matrix R , and the translation vector $\mathbf{t}$ using QR factorization [91]. Retrieving intrinsic and extrinsic parameters can be done by following the steps given in (6.11), (6.12) and (6.13)

$$H = \hat{P}_{3 \times 3}^{-1} \quad (6.11)$$

$$\tilde{C} = -H\hat{P}_{:4} \quad (6.12)$$

$$\hat{R}^T, \hat{K}^{-1} = \text{QR Decomposition}(H) \quad (6.13)$$

The matrix $\hat{P}_{3 \times 3}$ denotes the 3x3 matrix constructed from the initial three rows and columns of $\hat{P}$, while $\hat{P}_{:4}$ represents the fourth column of $\hat{P}$.

6.5. Simulations

In this section, the localization performance of DLT method across different scenarios is presented. For this purpose, we use a georeferenced aerial imagery (orthophotos) dataset obtained from the US Geological Survey's Earth Explorer website, accessible at <https://earthexplorer.usgs.gov/>. In Fig. 6.6, an orthophoto with a resolution of 100 megapixels, spanning an area of 2.56 square kilometers, is displayed.

6.5.1. Synthetic Camera Image Generation

To evaluate various feature extraction and matching algorithms, it was decided that generating images using a camera with a well-defined camera projection matrix P would be the most effective method. In the following algorithm, the procedure that followed to generate artificial camera images are outlined.

Algorithm 7 Image Generation

Require: $I_{orth}, 2Dlats_{orth}, 2Dlons_{orth}, f, m_x, m_y, p_x, p_y, lLh_{cam}, C_{BODY}^{CAM}, C_{NED}^{BODY}$

Ensure: I_{cam}

- 1: Flatten M by N $2Dlats_{orth}, 2Dlons_{orth}$ and stack vertically.
 - 2: Create the matrix Pos_{lLh} as a $3 \times (MN)$ matrix, L , where each column corresponds to the latitude, longitude, and altitude coordinates of the locations of the orthophoto pixels, and all altitude values are set to zero.
 - 3: Convert each column of Pos_{lLh} into ecef position vector and form Pos_{Ecef}
 - 4: Calculate intrinsic camera calibration matrix K using (6.3)
 - 5: Calculate $R = C_{CAM}^{ECEF} = C_{BODY}^{CAM} C_{NED}^{BODY} C_{ECEF}^{NED}$
 - 6: Calculate $\mathbf{t} : -R \text{ILh2Ecef}(lLh_{cam})$
 - 7: Construct P as given in (6.2)
 - 8: Evaluate pixel coordinates p_{uv} of each vector in $\hat{P}os_{Ecef}$ using $p_{uv} = P \hat{P}os_{Ecef}$ where
 - 9: Flatten each channel of I_{orth} to get $I_R^{flatten}, I_G^{flatten}, I_B^{flatten}$
 - 10: Discretize p_{uv} Discard pixel locations outside the camera resolution limits, apply the same indexing operation to $I_R^{flatten}, I_G^{flatten}, I_B^{flatten}$
 - 11: Determine the average intensity values for each distinct pixel location by averaging out the intensity values at repeated pixel locations.
 - 12: Reshape $I_R^{flatten}, I_G^{flatten}, I_B^{flatten}$ to match the camera's resolution and stack these arrays along the third dimension.
-

In Fig. 6.7, we present a synthetic image captured by a ground-looking camera. The intrinsic and extrinsic parameters of the camera are detailed in Table 6.2 and 6.3.

Cam Latitude	0.76073611
Cam Longitude	-1.68791624
Cam Altitude	1000
R,P,Y(Cam 2 Body)	[0 0 0]
R,P,Y(Body 2 NED)	[0 0 $\pi/2$]

Table 6.2: Camera Extrinsics.

f	4 mm
m_x	400
m_y	400
p_x	1.2 mm
p_y	1.2 mm

Table 6.3: Camera Intrinsics.

In Fig. 6.8, the positions of the camera centers, where patches are created, are indicated



Figure 6.6: 100 MP orthophoto downloaded from USGS.

by red dots. We extract four different feature descriptors for each patch and attempt to match these generated features with the corresponding feature descriptors obtained from the reference orthophoto. Fig. 6.9 to Fig.6.12 present the error graphs for DLT when various feature matching algorithms are applied. These graphs depict the horizontal axis representing the camera location indices, as shown in 6.8, and the vertical axis indicating the associated positioning error, measured in meters. In the presented error diagrams, bars colored in red mean the absence of valid correspondences between the image features and the reference orthophoto features. SIFT descriptor demonstrates superior accuracy in feature matching for the specified orthophoto, outperforming other feature descriptors. The AKAZE and ORB descriptor, while slightly less robust than SIFT, are useful in producing reliable feature matching. In contrast, FAST descriptor seems to be unreliable for this type of matching problem, as the matching algorithm often struggles to match these features.

Figure 6.13 depicts a valid feature association example, where the top right portion of



Figure 6.7: Synthetically produced camera image.

the image displays the synthetic view captured from the camera location number 4 as showed in Fig. 6.8 and the remaining part of the image shows the orthophoto. The lines connect the feature points of both images that are associated with each other.

6.6. Discussion

In this section of the thesis, we concentrate on the AVP problem, employing the DLT algorithm. This approach involves extracting various types of visual features from both reference orthophotos and synthetically generated images that capture a small portion of the reference orthophoto. The simulation outcomes reveal that accurate positioning is achieved when feature points from the camera image and the reference orthophoto are matched, provided that SIFT or AKAZE is employed for feature extraction. The precision in matching feature points, extracted from the camera image, to those from the reference orthophoto directly influences the accuracy of the DLT. However, a substantial drawback of the approach employed in these simulations is that the camera images



Figure 6.8: Camera locations.

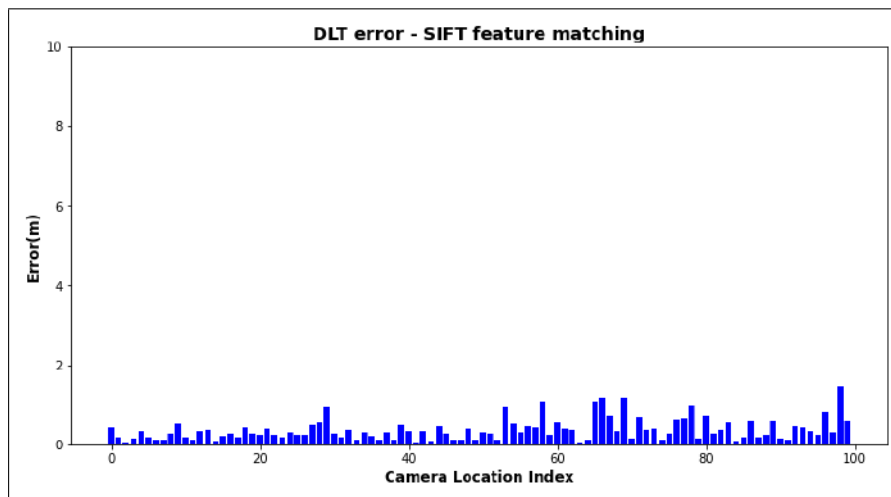


Figure 6.9: DLT positioning error - (SIFT).

originate from the orthophoto itself. As a result, foreground elements such as vehicles or shadows of large objects are identical in both the generated and reference images.

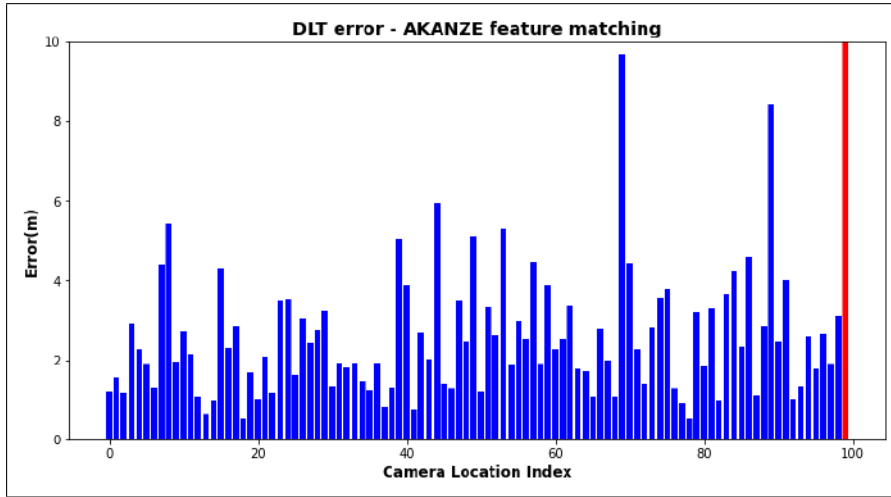


Figure 6.10: DLT positioning error - (AKAZE).

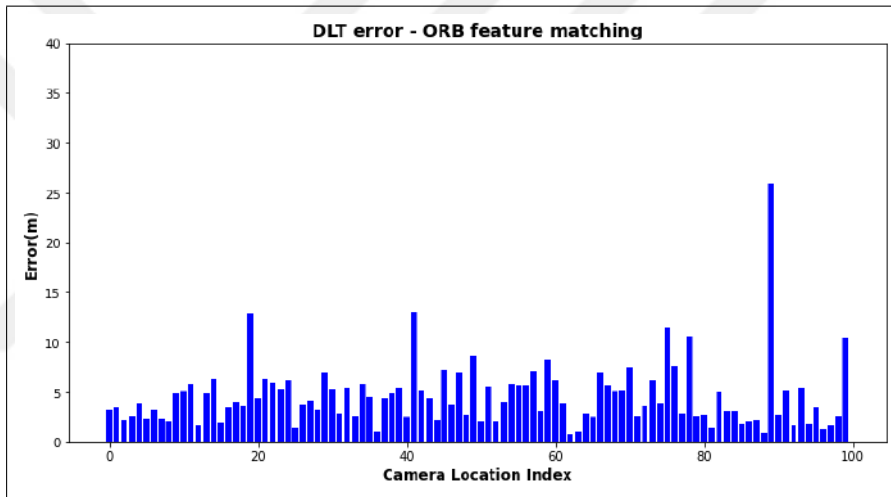


Figure 6.11: DLT positioning error - (ORB).

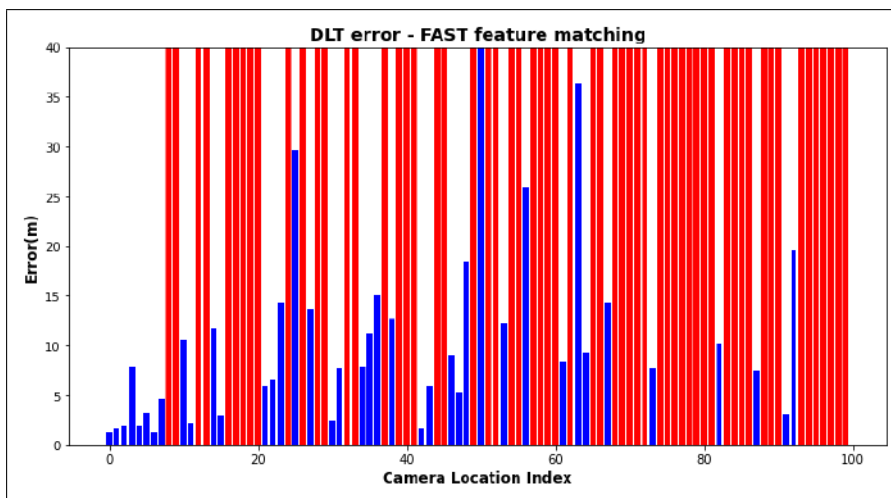


Figure 6.12: DLT positioning error - (FAST).



Figure 6.13: A valid feature association example.

7. CONCLUSIONS AND FUTURE WORK

7.1. Conclusions

This dissertation primarily addresses two subjects. First, it explores the integration of information from external sensors, including cameras and distance measuring devices, such as altimeters, into INS to mitigate their intrinsic unbounded position errors. A novel multi-hypothesis based TAN algorithm is proposed and its effectiveness is tested on simulation environment with a real terrain dataset. Simulation outcomes reveal that, despite significant initial errors in position, velocity, and heading, the proposed approach yields accurate position estimates. Furthermore, the adaptability of the hypothesis count to the terrain, which increases over nearly flat or multimodal regions and decreases over rough terrain, strengthens the proposed framework's resilience to divergence issues. Furthermore, this thesis further engages with the VAN problem by examining the utility of different visual descriptors in feature matching-based AVP. The approach encompasses the synthesis of camera images, achieved by leveraging an actual orthophoto and a known camera calibration matrix. This step is followed by the feature matching process, where features from synthetic camera images are matched with the same types of features extracted from the orthophoto.

This thesis also focuses on developing a technique for generating realistic IMU data for standalone INS or master-slave configurations. This method takes into account various internal error sources and external factors, such as vibrations, enabling the testing and validation of an integrated navigation algorithm within a simulation environment. This approach facilitates bypassing extensive data collection in real systems, allowing for a rapid evaluation of the overall performance of an integrated navigation algorithm.

7.2. Future Works

In this section, we present potential research directions for both the TAN and VAN problems, based on the insights obtained from the simulation results of each problem.

7.2.1. Terrain Aided Navigation

- When deriving the MHF for the TAN problem, we assumed that the radar altimeter consistently measures distance in the direction of gravity, with its error characterized as white noise with constant variance. However, the error characteristics of radar altimeters may vary with the aircraft's altitude and may not exhibit purely white noise behavior. Therefore, to implement the proposed MHF in a real-time system, it is crucial to model the error characteristics of the radar (or laser) altimeter by analyzing data collected from actual flights. If a statistical error model for a radar altimeter is known or modeled, the likelihood function in the measurement generation step of the proposed MHF can be adjusted to prevent performance degradation.
- Similar to the assumption regarding radar altimeter error, pressure altimeter error is also presumed to be white with constant variance. However, in real systems, pressure altimeter errors are affected by various factors such as temperature changes, humidity, and aircraft speed. Thus, in the real-time implementation of the MHF, the likelihood function in the measurement generation step could be modified in accordance with the actual error characteristics of the pressure altimeter, determined by extensive analysis of the collected pressure altimeter data.
- In this study, data delays, which are often present in real-time systems, are ignored. The proposed MHF could be further enhanced for effective operation in real-time systems by incorporating time of reception and validity mechanisms to account for the deterministic part of the delays. Additionally, further analysis and fine tunings in parameters could help mitigate performance degradation caused by unmodeled stochastic data delays in real-time systems.

7.2.2. Vision Aided Navigation

- Weather conditions like cloudiness in camera images can hide many ground features. Moreover, in a realistic setting, the orthophoto and camera image are taken at different times, leading to minor discrepancies between the camera image

and the same scene in the orthophoto. These factors increase the likelihood of incorrect feature matching, resulting in decreased positioning accuracy in AVP. With the advent of generative AI, it has become more feasible to make such modifications or enhancements to images. For instance, a recent study [92] demonstrates the use of Generative Adversarial Networks (GANs) to remove clouds from satellite images. Such frameworks could potentially be used to create various modifications or enhancements on images captured by ground-looking cameras.



REFERENCES

- [1] Fraundorfer F., Scaramuzza D., (2012), “Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications”, IEEE Robotics & Automation Magazine, 19(2), 78–90.
- [2] Titterton D., Weston J., (2004), “Strapdown inertial navigation technology”, 2nd Edition, Institution of Engineering and Technology.
- [3] Rogers R.M., (2003), “Applied Mathematics in Integrated Navigation Systems”, 2nd Edition, American Institute of Aeronautics and Astronautics.
- [4] Kaplan E.D., Hegarty C., (2017), “Understanding GPS/GNSS: principles and applications”, 3rd Edition, Artech House.
- [5] Groves P.D., (2013), “Principles of gnss, inertial, and multi-sensor integrated navigation systems”, 2nd Edition, Artech House.
- [6] Hibbeler R.C., Yap K.B., (2017), “Engineering mechanics: Dynamics”, Fourteenth in SI Units, Global Edition, Pearson Education Harlow.
- [7] Anon., (1997), “World geodetic system 1984”, Technical Report No:TR8350.2, National Imagery and Mapping Agency (now NGA) Technical Reports, 1(3).department of Defense.
- [8] Altamimi Z., Sillard P., Boucher C., (2002), “Itrf2000: A new release of the international terrestrial reference frame for earth science applications”, Journal of Geophysical Research: Solid Earth, 107(B10), ETG 2–1–ETG 2–19.
- [9] Malys S., et al., (1997), “Refinements to the world geodetic system, 1984”, Proc. ION GPS97, 915–920, Kansas, MO, September.
- [10] Savage P.G., (1998), “Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms”, Journal of Guidance, Control, and Dynamics, 21(2), 208–221.
- [11] Ignagni M.B., (1996), “Efficient class of optimized coning compensation algorithms”, Journal of Guidance, Control, and Dynamics, 19(2), 424–429.
- [12] Ignagni M., (2020), “Enhanced Strapdown Attitude Computation”, Journal of Guidance, Control, and Dynamics, 43(6), 1220–1224.
- [13] Bar.Shalom Y., Li X.R., Kirubarajan T., (2001), “Estimation with applications to tracking and navigation: Theory, algorithms and software”, 1st Edition, John Wiley & Sons, Inc..
- [14] Farrell J.A., Barth M., (1999), “The global positioning system and inertial navigation: Theory and practice”, 1st Edition, McGraw-Hill.
- [15] Costley A., Christensen R., Leishman R.C., Droge G., (2021), “Analytical Aircraft State and IMU Signal Generator from Smoothed Reference Trajectory”, arXiv:2105.12001 [cs, eess], .arXiv: 2105.12001.

- [16] Giroux R., Landry R.J., Leach B., Gourdeau R., (2003), "Validation and Performance Evaluation of a Simulink Inertial Navigation System Simulator", Canadian Aeronautics and Space Journal, 49(4), 14.
- [17] Gonzalez R., Giribet J.I., Patiño H.D., (2015), "Navego: A simulation framework for low-cost integrated navigation systems", Journal of Control Engineering and Applied Informatics, 17(2), 110–120.
- [18] Tan C., Zhu X., Wang Y., Su Y., (2016), "A new IMU data generator with suppression of discretization error for the SINS simulation", 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), 177–181, Nanjing, China, Aug..
- [19] Nassar S., Schwarz K.P., El.Sheimy N., Noureldin A., (2004), "Modeling Inertial Sensor Errors Using Autoregressive (AR) Models", Navigation, 51(4), 259–268.
- [20] Yuksel Y., Kaygisiz H.B., (2006), "Analysis of Vision Aided Inertial Navigation Systems", 2006 IEEE 14th Signal Processing and Communications Applications, 1–4, Antalya, Turkey, June.
- [21] Groves P.D., (2015), "Navigation using inertial sensors [Tutorial]", IEEE Aerosp. Electron. Syst. Mag., 30(2), 42–69.
- [22] Dierendonck A.J., McGraw J.B., Brown R.G., (1984), "Relationship between allan variances and kalman filter parameters", Proc. 16th Annu. Precise Time Time Interval (PTTI) Appl. Plan. Meet., 273–293, Greenbelt, MD, USA, Jan..
- [23] Ing Z., Gebre.Egziabher D., (2008), "Modeling and bounding low cost inertial sensor errors", 2008 IEEE/ION Position, Location and Navigation Symposium, 1122–1132, Monterey, CA, May.
- [24] Stebler Y., Guerrier S., Skaloud J., Victoria.Feser M.P., (2012), "A framework for inertial sensor calibration using complex stochastic error models", Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium, 849–861, Myrtle Beach, South Carolina, Aug.
- [25] Farrell J.A., Silva F.O., Rahman F., Wendel J., (2022), "Inertial Measurement Unit Error Modeling Tutorial: Inertial Navigation System State Estimation with Real-Time Sensor Calibration", IEEE Control Systems Magazine, 42(6), 40–66.
- [26] Hemerly E.M., (2017), "Mems imu stochastic error modelling", Systems Science & Control Engineering, 5(1), 1–8.
- [27] Wall J.H., Bevely D.M.e.a., (2005), "Characterization of various imu error sources and the effect on navigation performance", Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2005), 967–978, Long Beach, California, June.
- [28] Oppenheim A.V., Schafer R.W., (2009), "Discrete-time signal processing", 3, Pearson.
- [29] Irvine T., (2002), "An introduction to the shock response spectrum", Rev P, Vibrationdata, .

- [30] Command N.M., (1979), "Navy manufacturing screening program: Decreased corporate costs increase fleet readiness", NAVMAT P-9492, 11–12, Washington, D.C., May.
- [31] Gebre.Egziabher D., Gleason S., (2009), "Gnss applications and methods", 1st Edition, Artech House.
- [32] Kalman R.E., (1960), "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME–Journal of Basic Engineering, 82(Series D), 35–45.
- [33] Kalman R.E., (1960), "A new approach to linear filtering and prediction problems", Transactions of the ASME–Journal of Basic Engineering, 82(Series D), 35–45.
- [34] St..Pierre M., Gingras D., (2004), "Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system", IEEE Intelligent Vehicles Symposium, 2004, 831–835, Parma, Italy, June.
- [35] Bar.Itzhack I., (1989), "Observability studies of inertial navigation systems", Guidance, Navigation and Control Conference, 1283–1289, Boston, MA, U.S.A., Aug..
- [36] Golden J.P., (1980), "Terrain Contour Matching (TERCOM): A Cruise Missile Guidance Aid", Image Processing for Missile Guidance, 10–18, Washington, D.C., Dec..
- [37] Baker W., Clem R., (1977), "Terrain contour matching [TERCOM] primer", Technical Report ASP-TR-77-61. Aeronautical Systems Division, Wright-Patterson AFB, .
- [38] Hollowell J., (1990), "Helisitan: a terrain referenced navigation algorithm for helicopters", IEEE Symposium on Position Location and Navigation: A Decade of Excellence in the Navigation Sciences, 616–625, Las Vegas, Nevada, Mar..
- [39] Zhao L., Gao N., Huang B., Wang Q., Zhou J., (2015), "A novel terrain-aided navigation algorithm combined with the tercom algorithm and particle filter", IEEE Sensors Journal, 15(2), 1124–1131.
- [40] Sonmez T., Bingol H.E., (2008), "Modeling and simulation of a terrain aided inertial navigation algorithm for land vehicles", 2008 IEEE/ION Position, Location and Navigation Symposium, 1046–1052, Monterey, CA, USA, May.
- [41] Qingtang F., Lincheng S., Wenseng C., (2003), "Terrain aided navigation using PDAF", IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003, 1063–1068 vol.2, Changsha, China, Oct..
- [42] Ding P., Cheng X., (2022), "A new contour-based combined matching algorithm for underwater terrain-aided strapdown inertial navigation system", Measurement, 202, 111870.
- [43] Wei E., Dong C., Yang Y., Liu J., Gong G., Tang S., Deng Z., (2017), "A robust solution of integrated sitan with tercom algorithm weight-reducing iteration technique

for underwater vehicles' gravity-aided inertial navigation system", *Navigation*, 64(1), 12.

- [44] ZePeng H., Yun Y., WenWei W., Zhi M.B., (2018), "An algorithm of terrain matching aided navigation for uavs", 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), 1–6, Xiamen, China, Aug..
- [45] Convie M., Wilkinson N., Powlesland R., (2008), "Latest development of the terprom® digital terrain system (dts)", 2008 IEEE/ION Position, Location and Navigation Symposium (PLANS), 1219–1229, Monterey, California, May.
- [46] Yuan G., Zhang H., Yuan K., Zhu L., (2012), "Improved sitan algorithm in the application of aided inertial navigation", *Proceedings of 2012 International Conference on Measurement, Information and Control*, 922–926, Harbin, China, May.
- [47] Enns R., Morrell D., (1995), "Terrain-aided navigation using the Viterbi algorithm", *Journal of Guidance, Control, and Dynamics*, 18(6), 1444–1449.
- [48] Bergman N., Ljung L., Gustafsson F., (1999), "Terrain navigation using Bayesian statistics", *IEEE Control Systems Magazine*, 19(3), 33–40.
- [49] Bergman N., (1997), "A Bayesian Approach to Terrain-Aided Navigation", *IFAC Proceedings Volumes*, 30(11), 1457–1462.
- [50] Gustafsson F., Gunnarsson F., Bergman N., Forssell U., Jansson J., Karlsson R., Nordlund P.J., (2002), "Particle filters for positioning, navigation, and tracking", *IEEE Trans. Signal Process.*, 50(2), 425–437.
- [51] Turan B., Kutay A.T., (2016), "Particle filter studies on terrain referenced navigation", 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS), 949–954, Savannah, GA, USA, August.
- [52] Palmier C., Dahia K., Merlinge N., Del.Moral P., Laneville D., Musso C., (2019), "Adaptive approximate bayesian computational particle filters for underwater terrain aided navigation", 2019 22nd International Conference on Information Fusion (FUSION), 1–8, Ottawa, Ontario, Canada, Jul..
- [53] Salavasidis G., Munafò A., McPhail S.D., Harris C.A., Fenucci D., Pebody M., Rogers E., Phillips A.B., (2021), "Terrain-aided navigation with coarse maps—toward an arctic crossing with an auv", *IEEE Journal of Oceanic Engineering*, 46(4), 1192–1212.
- [54] Teixeira F.C., Pascoal A., Maurya P., (2012), "A Novel Particle Filter Formulation with Application to Terrain-Aided Navigation", *IFAC Proceedings Volumes*, 45(5), 132–139.
- [55] Rupeng W., Ye L., Teng M., Zheng C., Yusen G., Pengfei X., (2020), "Improvements to terrain aided navigation accuracy in deep-sea space by high precision particle filter initialization", *IEEE Access*, 8, 13029–13042.
- [56] Turan B., (2020), "Comparison of nonlinear filtering methods for terrain referenced aircraft navigation", 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), 144–149, Portland, Oregon, Apr..

- [57] Nordlund P., Gustafsson F., (2009), “Marginalized Particle Filter for Accurate and Reliable Terrain-Aided Navigation”, *IEEE Transactions on Aerospace and Electronic Systems*, 45(4), 1385–1399.
- [58] Ristic B., Arulampalam S., Gordon N., (2003), “Beyond the kalman filter: Particle filters for tracking applications”, 1, Artech House.
- [59] Palmier C., Dahia K., Merlinge N., Laneuville D., del Moral P., (2021), “Interacting weighted ensemble kalman filter applied to underwater terrain aided navigation”, 2021 American Control Conference (ACC), 1541-1546, Virtual Conference, originally scheduled for New Orleans, LA, USA, May.
- [60] Teixeira F.C., Quintas J., Maurya P., Pascoal A., (2017), “Robust particle filter formulations with application to terrain-aided navigation”, *International Journal of Adaptive Control and Signal Processing*, 31(4), 608–651.
- [61] Siouris G.M., (2004), “Missile guidance and control systems”, 1st, Springer.
- [62] Mok S.H., Bang H., (2013), “Terrain Slope Estimation Methods Using the Least Squares Approach for Terrain Referenced Navigation”, *International Journal of Aeronautical and Space Sciences*, 14(1), 85–90.
- [63] Tokta A., Hocaoglu A.K., (2023), “A novel multi-hypothesis filter for terrain aided navigation”, *IEEE Access*, 11, 62758–62770.
- [64] Noureldin A., Karamat T.B., Georgy J., (2013), “Fundamentals of inertial navigation, satellite-based positioning and their integration”, 1, Springer.
- [65] Yuksel Y., ElSheimy N., Noureldin A., (2010), “Error modeling and characterization of environmental effects for low cost inertial MEMS units”, *IEEE/ION Position, Location and Navigation Symposium*, 598–612, Renaissance Esmeralda Resort and Spa, Indian Wells, California, USA, May.
- [66] ElSheimy N., Hou H., Niu X., (2008), “Analysis and modeling of inertial sensors using allan variance”, *IEEE Transactions on Instrumentation and Measurement*, 57(1), 140–149.
- [67] Goshen.Meskin D., Bar.Itzhack I.Y., (1992), “Observability analysis of piece-wise constant systems. i. theory”, *IEEE Transactions on Aerospace and Electronic Systems*, 28(4), 1056–1067.
- [68] Van.Loan C., (1978), “Computing integrals involving the matrix exponential”, *IEEE Transactions on Automatic Control*, 23(3), 395–404.
- [69] Hartigan J.A., Wong M.A., (1979), “Algorithm AS 136: A K-Means Clustering Algorithm”, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108.publisher: [Wiley, Royal Statistical Society].
- [70] Nielsen F., Sun K., Marchand.Maillet S., (2017), “On Hölder Projective Divergences”, *Entropy*, 19(3), 122.
- [71] Salmond D.J., Atherton D.P., Bather J.A., (1988), “Mixture Reduction Algorithms for Uncertain Tracking”, *IFAC Proceedings Volumes*, 21(9), 775–780.
- [72] Nister D., Naroditsky O., Bergen J., (2004), “Visual odometry”, *Proceedings of*

the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 652–659, USA, 27 June–2 July.

- [73] Olson C.F., Matthies L.H., Schoppers M., Maimone M.W., (2000), “Robust stereo ego-motion for long distance navigation”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 453–458, Hilton Head, South Carolina, Jun..
- [74] Moravec H.P., (1980), “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover” [Ph.D. Dissertation]. Stanford University. Stanford, CA
- [75] Matthies L., Shafer S., (1987), “Error modeling in stereo navigation”, IEEE Journal of Robotics and Automation, 3(3), 239–248.
- [76] Lacroix S., Mallet A., Chatila R., Gallo L., (1999), “Rover self localization in planetary-like environments”, Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), 433–440, Noordwijk, The Netherlands, Jun..
- [77] Milella A., Siegwart R., (2006), “Stereo-based ego-motion estimation using pixel tracking and iterative closest point”, Proceedings of the IEEE International Conference on Vision Systems (ICVS), 21–24, New York, NY, USA, Jan..
- [78] Nistér D., Naroditsky O., Bergen J., (2006), “Visual odometry for ground vehicle applications”, Journal of Field Robotics, 23(1), 3–20.
- [79] Corke P.I., Strelow D., Singh S., (2004), “Omnidirectional visual odometry for a planetary rover”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4007–4012, Sendai, Oct..
- [80] Tardif J.P., Pavlidis Y., Daniilidis K., (2008), “Monocular visual odometry in urban environments using an omnidirectional camera”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2531–2538, Nice, France, Sep..
- [81] Chiodini S., Giubilato R., Pertile M., Debei S., (2020), “Retrieving scale on monocular visual odometry using low-resolution range sensors”, IEEE Transactions on Instrumentation and Measurement, 69(8), 5875–5889.
- [82] Antigny N., Uchiyama H., Servieres M., Renaudin V., Thomas D., Taniguchi R.I., (2019), “Solving monocular visual odometry scale factor with adaptive step length estimates for pedestrians using handheld devices”, Sensors, 19(4), 953.
- [83] Xu Y., Pan L., Du C., Li J., Jing N., Wu J., (2018), “Vision-based UAVs Aerial Image Localization: A Survey”, Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI’18), 9–18, Seattle, Washington, USA, nov.
- [84] Theodoridis S., Koutroumbas K., (2008), “Pattern recognition”, 4, Academic Press.
- [85] Brunelli R., (2009), “Template matching techniques in computer vision: Theory and practice”, 1, John Wiley Sons.

- [86] Jakubović A., Velagić J., (2018), “Image feature matching and object detection using brute-force matchers”, 2018 International Symposium ELMAR, 83-86, Zadar, Croatia, Sep..
- [87] Muja M., Lowe D.G., (2009), “Fast approximate nearest neighbors with automatic algorithm configuration”, Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (VISIGRAPP 2009) - Volume 1: VISAPP, 331-340, Lisboa, Portugal, Feb..
- [88] Beis J.S., Lowe D.G., (1997), “Shape indexing using approximate nearest-neighbour search in high-dimensional spaces”, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1000-1006, Puerto Rico, Jun..
- [89] Szeliski R., (2022), “Computer vision: Algorithms and applications”, 2nd Edition, Springer International Publishing.
- [90] Hartley R., Zisserman A., (2003), “Multiple view geometry in computer vision”, 2nd Edition, Cambridge University Press.
- [91] Strang G., (2016), “Linear algebra and its applications”, 5th Edition, Cengage Learning.
- [92] Chen.Song D., Khalaji E., Rani V., (2022), “Mm811 project report: Cloud detection and removal in satellite images”, arXiv, .

BIOGRAPHY

Aybars TOKTA received his B.S. and M.S. degrees from Gebze Technical University, where he is also pursuing his Ph.D. degree. From 2015 to 2021, he served as a research assistant at Gebze Technical University. Currently, he works as a navigation system design engineer at Aselsan Inc. His research interests include signal processing, target tracking, and navigation.



PUBLICATIONS WITHIN the SCOPE of THESIS STUDY

- [1] Tokta A., Hocaoglu A. K. (2023), “A Novel Multi-Hypothesis Filter for Terrain Aided Navigation”, IEEE Access, 11, 62758–62770.
- [2] Tokta A., Hocaoglu A. K. ”A Tutorial on Synthetic IMU Data Generation” (to be submitted to IEEE Access)
- [3] Tokta A., Hocaoglu A. K. (2019), “Sensor Bias Estimation for Track-to-Track Association”, IEEE Signal Processing Letters, 26(10), 1426–1430.
- [4] Tokta A., Hocaoglu A. K. (2018), “A Fast Track to Track Association Algorithm by Sequence Processing of Target States”, Proceedings of the 21st International Conference on Information Fusion (FUSION), 1-5, Cambridge, UK, July.
- [5] Tokta A., Hocaoglu A. K. (2018), “A Track to Track Association Algorithm Based on Weighted State Correlation Similarity”, Proceedings of the International Conference on Artificial Intelligence and Data Processing (IDAP), 1-4, Malatya, Turkey, September.