A COMPARATIVE STUDY ON TIGHTLY COUPLED VISUAL AIDED INERTIAL NAVIGATION SYSTEMS FOR UNMANNED AERIAL VEHICLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TALHA İNCE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2018

Approval of the thesis:

A COMPARATIVE STUDY ON TIGHTLY COUPLED VISUAL AIDED INERTIAL NAVIGATION SYSTEMS FOR UNMANNED AERIAL VEHICLES

submitted by **TALHA İNCE** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tolga Çiloğlu ______________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Afşar Saranlı ______________
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Mübeccel Demirekler
Electrical and Electronics Eng. Dept., METU ______________

Assoc. Prof. Dr. Afşar Saranlı
Electrical and Electronics Eng. Dept., METU ______________

Prof. Dr. Abdullah Aydın Alatan
Electrical and Electronics Eng. Dept., METU ______________

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Eng. Dept., METU ______________

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Eng. Dept., Hacettepe University ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: TALHA İNCE

Signature :

ABSTRACT

A COMPARATIVE STUDY ON TIGHTLY COUPLED VISUAL AIDED INERTIAL NAVIGATION SYSTEMS FOR UNMANNED AERIAL VEHICLES

İNCE, TALHA
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Afşar Saranlı

JANUARY 2018, 123 pages

An Inertial Navigation System (INS) is a combination of hardware (accelerometers and gyroscopes) and algorithms to calculate the position, orientation and velocity of a mobile platform. Because of the need to integrate the measurements over time, INS is subjected to cumulative error characteristics, hence cannot provide an accurate navigation solution over long durations. Global Positioning System (GPS) is often used for long time-long distance problems aiding INS. GPS relies on external signals received from satellite networks which gives bounded error. Unfortunately, these signals can be jammed, spoofed or may not be available at indoor applications. Hence, alternatives to the GPS sensor are required. Such a recent alternative is the camera sensor, resulting in Visual Aided Inertial Navigation System (VINS). VINS aims to provide accurate navigation solution using a fusion of imaging and inertial sensor data. The main goal of this thesis is to analyze and increase the performance of VINS sensor fusion algorithms on an unmanned aerial vehicle (UAV). For this purpose, a realistic simulation environment is implemented and different VINS methods are comparatively studied. Extensive simulation studies are conducted to characterize the performance of map-based and mapless VINS methods and to study the effects of performance critical parameters. Finally, a modified Gaussian Mixture Filter and feature region selection method are proposed for increasing the performance of VINS.

Keywords: Visual Aided Inertial Navigation Systems, Sensor Fusion Filters, Indirect Extended Kalman Filter, Modeling and Simulation of Navigation Algorithm

ÖZ

İNSANSIZ HAVA ARAÇLARI İÇİN SIKICA BAĞLI GÖRSEL YARDIMLI ATALETSEL SEYRÜSEFER SİSTEMLERİ ÜZERİNE KARŞILAŞTIRMALI BİR ÇALIŞMA

İNCE, TALHA
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Afşar Saranlı

Ocak 2018 , 123 sayfa

Ataletsel Seyrüsefer Sistemi (ASS) hareketli bir platformun pozisyon, yönelim ve hızını hesaplamak için donanım (ivmeölçer ve dönüölçer) ve algoritmalardan oluşan bir seyrüsefer tekniğidir. ASS, ölçümlerin zaman üzerinden integralini alması sebebiyle biriken bir hata karakteristiğine sahiptir. Bu sebeple uzun süreli seyirlerde hassas bir seyrüsefer çözümü sağlayamaz. ASS'yi desteklemek için uzun süre ve uzun mesafeli problemlerde Küresel Konumlandırma Sistemi (KKS) kullanılır. KKS uydu ağlarından sınırlı hata ile aldığı küresel bilgiye bağımlıdır. Bu sinyaller karıştırılabilir, aldatılabilir veya kapalı mekan uygulamalarında alınamayabilir. Bu sebeple KKS algılayıcısına bir alternatif gerekmektedir. Böyle güncel bir alternatif Görsel Yardımlı Ataletsel Seyrüsefer Sistemi'ni (GYAS) meydana getiren kamera algılayıcısıdır. GYAS görüntüleme ve ataletsel algılayıcıları kullanarak hassas seyrüsefer çözümü sağlamayı amaçlar. Bu tezin amacı bir insansız hava aracı üzerinde GYAS algılayıcı birleştirme yöntemlerini analiz etmek ve performanslarını artırmaktır. Bu amaçla, gerçekçi bir benzetim ortamı gerçeklenmiş ve farklı GYAS yöntemleri karşılaştırmalı olarak çalışılmıştır. Harita tabanlı ve haritasız GYAS yöntemlerinin performanslarını ortaya koymak ve performans açısından önemli değişkenleri çalışmak için geniş çaplı analizler yürütülmüştür. Son olarak GYAS'ın performansını artırmak için değiştirilmiş bir Gauss Karışım Filtresi ve bir öznitelik bölgesi seçim yöntemi sunulmuştur.

Anahtar Kelimeler: Görsel Yardımlı Ataletsel Seyrüsefer Sistemleri, Algılayıcı Birleştirme Filtreleri, Dolaylı Genişletilmiş Kalman Süzgeci, Seyrüsefer Algoritması Modelleme ve Benzetimi

To My Family

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor Assoc. Prof. Dr. Afşar SARANLI for his guidance and encouragement throughout this study. He teach me showing patience to solve academic problems and also broaden my academic horizon to a much higher level with his great personality and profound knowledge. It has been a great honour and pleasure working with him on such a challenging topic. I am grateful for his valuable time that he allocated for this thesis.

This work has been supported by TÜBİTAK-SAGE (*The Scientific and Technological Research Council of Turkey – Defense Industries Research and Development Institute*). I would like to express my appreciation to head of the Guidance and Control Division; Dr. Kadriye KUTLUAY and head of the Navigation Division; Haluk Erdem BİNGÖL for giving me opportunity to conduct this study. I would like to show my gratitude to all my colleagues working in these divisions, especially Cansu Irmak CÖMERT, Ahmet ÇAKIROĞLU, Ali Can ARIK, Kerem EYİCE and Onur ÇULHA for their continuous support and consultancy. I am proud of working with such humble and leading experts in their research area.

Finally, I would like to thank my family for their life long support.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xix

LIST OF SYMBOLS xxi

CHAPTERS

1 INTRODUCTION 1

1.1 Introduction 1

1.2 Scope, Contributions and Assumptions 2

1.3 Outline of Thesis 4

2 LITERATURE REVIEW 5

3 REVIEW OF INERTIAL NAVIGATION SYSTEM MODELING . . 15

3.1 Background Information 16

3.1.1 Reference Frames 16

3.1.2 Direction Cosine Matrices 19

3.2 Strapdown System Modeling 21

3.2.1 Attitude Dynamics 22

3.2.2 Velocity Dynamics 23

3.2.3 Position Dynamics 24

3.3 Perturbation Analysis 24

3.3.1 Perturbation Analysis of Navigation Variables . . . 25

3.3.2 Error State Matrices 27

3.4 Discretization of Error State Matrices 29

3.5 Modeling Inertial Sensor Measurements 31

4 FUSION OF IMAGING AND INERTIAL MEASUREMENTS 35

4.1 Observation Model 36

4.1.1 Perturbation Analysis on Perspective Camera Model 38

4.1.2 Derivation of Landmark Position Covariance . . . 41

4.2 Extended Kalman Filter 46

4.3 Unscented Kalman Filter 52

4.4 Gaussian Mixture Filter 58

4.4.1 Resampling 64

4.5 An Efficient Feature Region Selection Method for Simultaneously Localization and Mapping 66

5 COMPARATIVE EVALUATION OF VINS ALGORITHMS IN A SIMULATION ENVIRONMENT 71

5.1 Methodology 71

5.1.1 Evaluation of EKF based GPS-INS Integration . . 76

5.2 Simulations on Map Based VINS 79

5.2.1 Effect of Measurement Noise on Map-Based VINS Performance 80

5.2.2 Joint Analysis of Landmarks Position Uncertainty and Number of Tracked Landmarks on Map-Based VINS Performance 83

5.2.3 Comparative Evaluation of Sensor Fusion Filters on Map-Based VINS 89

5.3 Simulations on Mapless VINS 93

5.3.1 Effect of Number of Landmarks on Mapless VINS Performance 94

5.3.2 Effect of Number of Tracked Frames on Mapless VINS Performance 97

5.3.3 Comparative Evaluation of Sensor Fusion Filters on Mapless VINS 100

6 CONCLUSION AND FUTURE WORKS 107

REFERENCES 111

APPENDICES

A CHI-SQUARE GOODNESS OF FIT TEST 117

B COMPUTING CLOSEST SYMMETRIC AND POSITIVE SEMIDEFINITE MATRIX 121

LIST OF TABLES

TABLES

Table 3.1 WGS-84 model [51] 19

Table 3.2 Typical Accelerometer and Gyro Biases for Different Grades of IMU [18] 31

Table 5.1 Tested UAV Configuration 74

Table 5.2 IMU Specifications 74

Table 5.3 Simulated Camera Configurations 74

Table 5.4 Initial Navigation Solution Error Standard Deviations 75

Table 5.5 Mixture Flight Profile: RMS and Std of Navigation Solution Errors . 83

Table 5.6 Ellipsis Flight Profile: RMS and Std of Navigation Solution Errors . 83

Table 5.7 Mixture Flight Profile: RMS and Std. Dev. of Position Error 88

Table 5.8 Mixture Flight Profile: RMS and Std. Dev. of Velocity Error 88

Table 5.9 Mixture Flight Profile: RMS and Std. Dev. of Attitude Error 88

Table 5.10 Ellipsis Flight Profile: RMS and Std. Dev. of Position Error 88

Table 5.11 Ellipsis Flight Profile: RMS and Std. Dev. of Velocity Error 89

Table 5.12 Ellipsis Flight Profile: RMS and Std. Dev. of Attitude Error 89

Table 5.13 Straight Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS 92

Table 5.14 Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS 92

Table 5.15 Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS 92

Table 5.16 The Average Time Intervals for Each INS Step of VINS 93

Table 5.17 Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Landmarks 97

Table 5.18 Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Landmarks 97

Table 5.19 Convergence Percentages under Different Number of Landmarks for All Flight Profiles 97

Table 5.20 Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Tracked Frames 100

Table 5.21 Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Tracked Frames 100

Table 5.22 Convergence Percentages under Different Number of Frames for All Flight Profiles 100

Table 5.23 Straight Flight Profile: RMS and Std of Navigation Solution Error . 104

Table 5.24 Mixture Flight Profile: RMS and Std of Navigation Solution Error . 104

Table 5.25 Ellipsis Flight Profile: RMS and Std of Navigation Solution Error . . 104

Table 5.26 Stability Percentages of Different Sensor Fusion Filters on Mapless VINS 104

Table 5.27 The Average Time Intervals for Each Step of VINS 105

LIST OF FIGURES

FIGURES

Figure 1.1 INS Error D4ivergence 2

Figure 1.2 Sub-systems of Visual Aided Inertial Navigation. The Thesis Focus is indicated in the center as the sensor fusion block. 3

Figure 2.1 Subproblems of VINS 5

Figure 2.2 Scale Ambiguity Problem 8

Figure 3.1 Inertial, Earth and Navigation Frames [51] 17

Figure 3.2 Body Reference Frame 18

Figure 3.3 Camera Frame 18

Figure 3.4 Geographic Coordinate System 19

Figure 4.1 Perspective Projection Model 36

Figure 4.2 Image Plane 37

Figure 4.3 VINS Operation 48

Figure 4.4 Input Parameters to Sensor Fusion 50

Figure 4.5 I-EKF Based VINS Main Flowchart 51

Figure 4.6 I-EKF Based VINS Prediction Flowchart 52

Figure 4.7 I-EKF Based VINS Update Flowchart 52

Figure 4.8 Propagation GRV with EKF 53

Figure 4.9 Propagation of Sigma Points through Nonlinear Function 53

Figure 4.10 I-UKF Based VINS Prediction Flowchart 56

Figure 4.11 I-UKF Based VINS Main Flowchart 57

Figure 4.12 I-UKF Based VINS Update Flowchart 58

Figure 4.13 Resampling Main Flowchart 62

Figure 4.14 I-GMF Based VINS Main Flowchart 63

Figure 4.15 Systematic Resampling Flowchart [3] 65

Figure 4.16 Camera Gimbal Angle Control and Imaging Geometry in terms of Vertical Field of View 66

Figure 4.17 Images are projection of trapezoid areas in real world to 2D image plane. 67

Figure 4.18 Imaging Geometry in terms of Horizontal Field of View 69

Figure 5.1 Tested Flight Profiles 73

Figure 5.2 Straight Flight Profile: GPS-INS Integration RMS Navigation Solution Errors 76

Figure 5.3 Straight Flight Profile: GPS-INS Integration Std. Dev. of Navigation Solution Errors 77

Figure 5.4 Mixture Flight Profile: GPS-INS Integration RMS Navigation Solution Errors 78

Figure 5.5 Mixture Flight Profile: GPS-INS Integration Std. Dev. of Navigation Solution Errors 79

Figure 5.6 Straight Flight Profile: RMS Navigation Solution Errors versus Pixel Errors(color traces) 81

Figure 5.7 Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms versus Pixel Errors(color traces) 82

Figure 5.8 Advantage of Using Multiple Landmarks 84

Figure 5.9 Straight Flight Profile: RMS Navigation Solution Errors versus Number of Landmarks (x-axis) and Std. Dev. of Landmark Position Errors (color traces) 85

Figure 5.10 Straight Flight Profile: Std. Dev. of Navigation Solution Errors versus Num. of Landmarks (x-axis) and Std. Dev. of Landmark Pos. Errors (color traces) 86

Figure 5.11 Straight Flight Profile: RMS Navigation Solution Errors versus Std. Dev. of Landmark Pos. Errors(x-axis) and Number of Landmarks(color traces) 87

Figure 5.12 Straight Flight Profile: Std. Dev. of Navigation Solution Errors versus Std. Dev. of Landmark Pos. Errors(x-axis) and Num. of Landmarks(color traces) 87

Figure 5.13 Straight Flight Profile: RMS Navigation Solution Errors of Different Sensor Fusion Filters and only INS 90

Figure 5.14 Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms of Different Sensor Fusion Filters and only INS 91

Figure 5.15 Straight Flight Profile: RMS Navigation Solution Error versus Number of Landmarks 95

Figure 5.16 Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms versus Number of Landmarks 96

Figure 5.17 Straight Flight Profile: RMS Navigation Solution Error versus Number of Tracked Frames 98

Figure 5.18 Straight Flight Profile: Std. Dev. of Navigation Solution Error versus Number of Tracked Frames 99

Figure 5.19 Straight Flight Profile: RMS Navigation Solution Error under Different Sensor Fusion Filters and Only INS 102

Figure 5.20 Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms under Different Sensor Fusion Filters and Only INS 103

Figure A.1 PDF of Chi Square Distribution 118

LIST OF ABBREVIATIONS

INS	Inertial Navigation System
GPS	Global Positioning System
VINS	Visual Aided Inertial Navigation System
UAV	Unmanned Aerial Vehicle
SLAM	Simultaneously Localization and Mapping
ASS	Ataletsel Seyrüsefer Sistemi
KKS	Küresel Konumlandırma Sistemi
GYAS	Görsel Yardımlı Ataletsel Seyrüsefer Sistemi
EKVH	Eşzamanlı Konumlandırma ve Haritalama
IMU	Inertial Measurement Unit
GMF	Gaussian Mixture Filter
DSMAC	Digital Scene Matching Area Correlator
SfM	Structure from Motion
PnP	Perspective N Point Problem
PnL	Perspective N Line Problem
DEM	Perspective N Line Problem
LiDAR	Light Detection and Ranging
i-frame	Inertial Frame
e-frame	Earth Centered Earth Fixed frame
ECEF	Earth Centered Earth Fixed frame
n-frame	Navigation Frame
b-frame	Body Frame
c-frame	Camera Frame
NED	North, East and Down
LLH	Latitude, Longitude and Height
WGS-84	World Geodetic System of 1984
DCM	Direction Cosine Matrix
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
GRV	Gaussian Random Variable

UT	Unscented Transform
I-UKF	Indirect Unscented Kalman Filter
RMS	Root Mean Square
Std. Dev.	Standard Deviation
Pos.	Position
Vel.	Velocity
Att.	Attitude

LIST OF SYMBOLS

$\boldsymbol{x}(t)$	State vector at time t
$\boldsymbol{u}(t)$	Input vector at time t
$\boldsymbol{\delta x}(t)$	State vector at time t
$\dot{\boldsymbol{x}}(t)$	Derivative of state vector at time t
$\dot{\boldsymbol{\delta x}}(t)$	Derivative of error state vector at time t
$\bar{\boldsymbol{x}}(t)$	True value of state vector at time t
$\bar{\boldsymbol{u}}(t)$	True value of input vector at time t
$\mathbf{x}_k$	State vector at time step k
$\mathbf{u}_k$	Input vector at time step k
$\mathbf{w}_k$	Process noise vector at time step k
$\mathbf{v}_k$	Process noise vector at time step k
$\mathbf{z}_k$	Measurement vector at time step k
$f_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1} + \mathbf{w}_k)$	Nonlinear differential equation of state space representation of system at time step k
$h_k(\mathbf{x}_k)$	Nonlinear observation model at time step k
$\boldsymbol{F}(t)$	Continuous time linear state transition matrix
$\boldsymbol{B}(t)$	Continuous time linear input matrix
$\boldsymbol{F}_k$	Discrete time linear state transition matrix
$\boldsymbol{G}_k$	Discrete time linear noise influence matrix
$\boldsymbol{H}_k$	Discrete time linear measurement matrix
$\boldsymbol{Q}_k$	Discrete time input noise covariance matrix
$\boldsymbol{R}$	Measurement noise covariance matrix
R	Length of the semi-major axis
r	Length of the semi-minor axis
f	Flattening of the ellipsoid
e	Major eccentricity of the ellipsoid
Ω	Earth's rate
R_N	Meridian radius of curvature
R_E	Transverse radius of curvature
$\boldsymbol{p}^n$	Position of the mobile platform in n-frame

$\boldsymbol{p}^{llh}$	Position of the mobile platform expressed in geodesic coordinate system
$\boldsymbol{v}^{n}$	Velocity of the mobile platform in n-frame
$\mathbf{C}_b^n$	DCM transforming a vector from b-frame to n-frame
$\mathbf{C}_n^b$	DCM transforming a vector from n-frame to n-frame
$\boldsymbol{f}^b$	Specific force vector in b-frame
$\boldsymbol{f}^n$	Specific force vector in n-frame
ϕ	Roll
θ	Pitch
ψ	Yaw
v_N	North velocity
v_E	East velocity
v_D	Down velocity
L	Latitude
l	Longitude
h	Height above sea level
$\boldsymbol{\Omega}_{nb}^b$	Skew-symmetric form of angular rate vector.
$\times\boldsymbol{\omega}_{nb}^b$	Skew-symmetric form of $\boldsymbol{\omega}_{nb}^b$ vector
$\boldsymbol{\omega}_{nb}^b$	Turn rate of b-frame w.r.t. n-frame expressed in b-frame, angular rate vector
$\boldsymbol{\omega}_{ib}^b$	Turn rate of b-frame w.r.t. i-frame expressed in b-frame
$\boldsymbol{\omega}_{ie}^n$	Turn rate of e-frame w.r.t. i-frame expressed in n-frame
$\boldsymbol{\omega}_{en}^n$	Turn rate of n-frame w.r.t. e-frame expressed in n-frame, transport rate
$\boldsymbol{g}_l^n$	Local gravity vector in n-frame
$\boldsymbol{\psi}$	Attitude error vector
$\boldsymbol{\omega}_{ib}^n$	Turn rate of b-frame w.r.t. i-frame expressed in n-frame
$\boldsymbol{\delta\omega}_{ib}^n$	Gyroscope's random walk vector
$\boldsymbol{\delta f}^b$	Accelerometer's random walk vector
$\boldsymbol{B_G}$	Fixed bias error of gyroscope
$\boldsymbol{B_g}$	g-dependent bias coefficients
$\boldsymbol{S_G}$	Constant scale factor matrix of gyroscope
$\boldsymbol{M_G}$	Mounting and cross-coupling error between axes of gyroscope
$\boldsymbol{w}_G$	Gyroscope's random walk

$\boldsymbol{B_A}$	Fixed bias error of accelerometer
$\boldsymbol{S_A}$	Constant scale factor matrix of accelerometer
$\boldsymbol{M_A}$	Mounting and cross-coupling error between axes of accelerometer
$\boldsymbol{w}_A$	Accelerometer's random walk
$\boldsymbol{s}^{pix}$	Unnormalized pixel coordinates of scene point
$\bar{\boldsymbol{s}}^{pix}$	Normalized pixel coordinates of scene point
$\boldsymbol{s}^c$	Coordinates of imaged scene point in c-frame
$\boldsymbol{s}_z^c$	z component of $\boldsymbol{s}^c$
$\boldsymbol{T}_c^{pix}$	Intrinsic camera matrix
s_x	Pixel pitch along x axis of c-frame
s_y	Pixel pitch along y axis of c-frame
res_x	Camera resolution along x axis of c-frame
res_y	Camera resolution along y axis of c-frame
f	Focal length
$\boldsymbol{T}_c^b$	Translation from b-frame to c-frame
$\boldsymbol{y}^n$	Position of landmark in n-frame
$\boldsymbol{H_{z\psi}}$	Derivative of nonlinear perspective projection equation with respect to attitude error
$\boldsymbol{H_{zv^n}}$	Derivative of nonlinear perspective projection equation with respect to velocity
$\boldsymbol{H_{zp^{llh}}}$	Derivative of nonlinear perspective projection equation with respect to position expressed in geodesic coordinate system
$\boldsymbol{H_{zy^n}}$	Derivative of nonlinear perspective projection equation with respect to position of landmarks expressed in n-frame
$\widetilde{\boldsymbol{s}}^c$	Estimated position of imaged scene point in c-frame
$\widetilde{\boldsymbol{C}}_n^b$	Estimated DCM
$\boldsymbol{I}_{3\times3}$	3×3 identity matrix
$\boldsymbol{0}_{3\times3}$	3×3 zero matrix
$\boldsymbol{P}_{k\|k-1}$	Covariance prediction
$\hat{\boldsymbol{x}}_{k\|k-1}$	State prediction
$\hat{\boldsymbol{z}}_{k\|k-1}$	Measurement estimate
$\hat{\boldsymbol{y}}_k$	Innovation or measurement residual
$\boldsymbol{S}_k$	Innovation [or residual] covariance
$\boldsymbol{K}_k$	Kalman gain

$\hat{\boldsymbol{x}}_{k\|k}$	Updated state prediction
$\boldsymbol{P}_{k\|k}$	Updated covariance prediction
$N(\boldsymbol{a},\boldsymbol{b})$	Normal Distribution having mean $\boldsymbol{a}$ and standart deviation $\boldsymbol{b}$
$\boldsymbol{P}_{0\|0}$	Initial state covariances
d_m	Distance between the mobile platform and visual landmark
$\boldsymbol{P}_{\boldsymbol{zz}}$	Pixel location uncertainty covariance
$\boldsymbol{P}_{\boldsymbol{xx}}$	State covariance
$\boldsymbol{y}_m^n$	Position of visual landmark in n-frame
$g()$	Visual landmark position function
$\delta\boldsymbol{z}$	Pixel location error
δd	Distance error
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{x}}$	Derivative of position of landmark m in n-frame w.r.t. navigation solution
$\boldsymbol{G}_{\boldsymbol{y}^n\boldsymbol{x}}$	Derivative of positions of landmarks in n-frame w.r.t. navigation solution
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{z}_m}$	Derivative of position of landmark m in n-frame w.r.t. pixel measurement
$\boldsymbol{G}_{\boldsymbol{y}^n\boldsymbol{z}}$	Derivative of positions of landmarks in n-frame w.r.t. pixel measurements
$\boldsymbol{G}_{\boldsymbol{y}_m^n d_m}$	Derivative of position of landmark m in n-frame w.r.t. distance between the mobile platform and landmark m
$\boldsymbol{G}_{\boldsymbol{y}^n d}$	Derivative of position of landmarks in n-frame w.r.t. distance between the mobile platform and landmarks
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{\psi}}$	Derivative of position of landmark m in n-frame w.r.t. attitude error
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{v}}$	Derivative of position of landmark m in n-frame w.r.t. velocity
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{p}^n}$	Derivative of position of landmark m in n-frame w.r.t. position in n-frame
$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{p}^{llh}}$	Derivative of position of landmark m in n-frame w.r.t. position expressed in geodesic coordinate system
L	Number of dimensions
$\boldsymbol{\chi}_i$	i-th sigma point
γ_i	Propagated sigma point covariance
$\boldsymbol{\mu}_{\boldsymbol{x}}$	Mean value of the state distribution
$\boldsymbol{\chi}_0$	Central sigma point
$\boldsymbol{\chi}_i$	Off-center sigma points

λ	Scaling parameter
β	Secondary scaling parameter
α	Spreading parameter
$\boldsymbol{\gamma}_i$	Propagated i-th sigma point
$W_i^{(m)}$	Weight of i-th sigma point for mean calculation
$W_i^{(c)}$	Weight of i-th sigma point for covariance calculation

CHAPTER 1

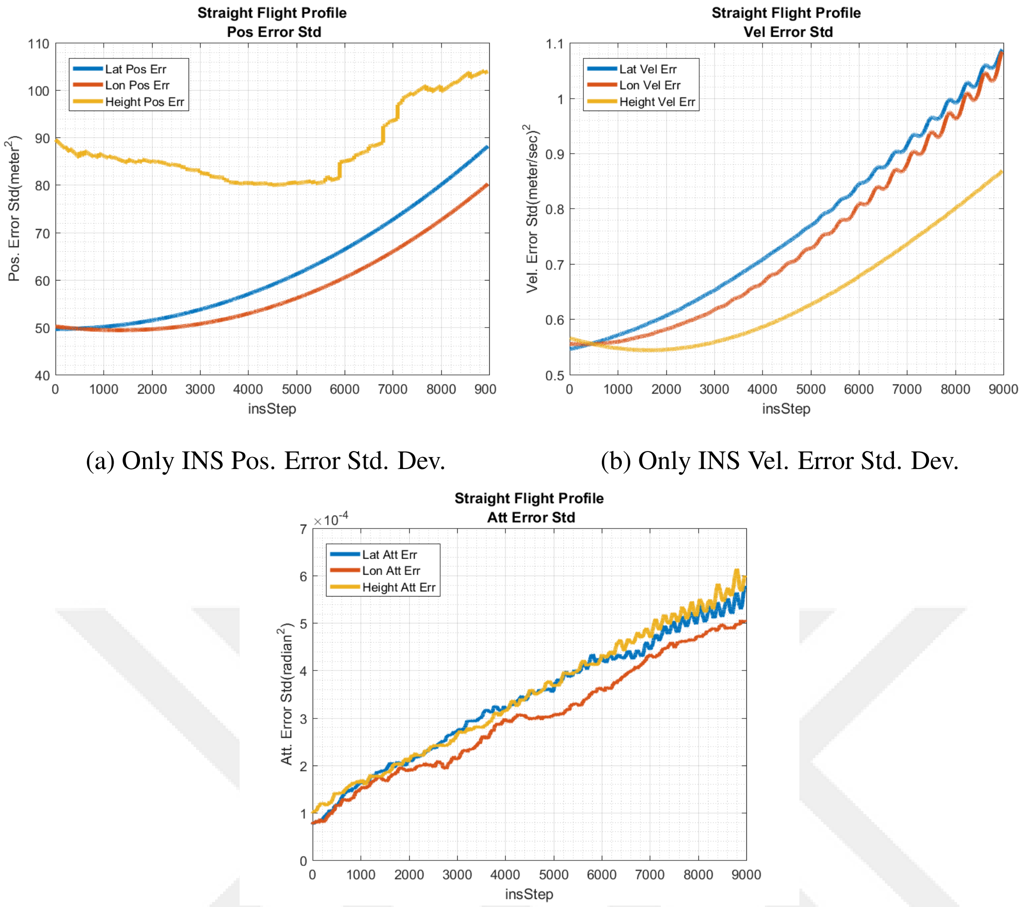
INTRODUCTION

1.1 Introduction

Navigation is a field of study that focuses on the process of monitoring the movement of a mobile platform from one place to another [4]. Navigation solution which includes position, velocity and attitude of the mobile platform is used by a pilot to reach the destination point. In order to generate a navigation solution, a navigation technique must be defined.

Required precision of the navigation solution differs for every application. For example, meter level precision may be redundant for a vessel's navigation while centimeter level precision may be a requirement for a military application. According to the required precision and available sensors on the mobile platform, different navigation techniques can be used.

In this thesis, a navigation technique on an unmanned aerial vehicle (UAV) is studied. UAVs use an inertial measurement unit (IMU) which contains accelerometers and gyroscopes to monitor the rotation and the acceleration of the mobile platform. Navigation technique using IMU to cumulatively calculate the navigation solution is referred as Inertial Navigation System (INS). Because of the need to integrate the IMU measurements which contains bias and noise over time, INS has divergent error characteristics as shown in Figure 1.1. Hence it must be aided with another navigation technique for long time - long distances problems. Traditional approaches use Global Positioning System (GPS) for aiding INS. However, GPS signals can be jammed, spoofed or may not be received at indoor applications. Mobile platform to be controlled may not have a GPS sensor because of size, weight, power consumption or cost constraints. Hence an alternative navigation technique must be used if the mobile platform is desired be to operated under GPS-denied conditions.



(a) Only INS Pos. Error Std. Dev.

(b) Only INS Vel. Error Std. Dev.

(c) Only INS Att. Error Std. Dev.

Figure 1.1: INS Error D4ivergence

A key parameter for the navigation technique selection is the availability of the required sensors on the mobile platform. Most of the UAVs uses cameras for various tasks. Using this available sensor, a method for aiding INS inspired by the nature is proposed in literature. Visual Aided Inertial Navigation Systems (VINS) provide a navigation solution using imaging and inertial sensors. VINS is a navigation aid independent from external sources. This standalone system attracted more researchers as the computer vision, information fusion, data association algorithms, signal processing and sensor technologies are developed.

1.2 Scope, Contributions and Assumptions

The study considers Visual Aided Inertial Navigation (VINS) which is outlined in Figure 1.2. The primary focus of our study is the Sensor Fusion block illustrated in the figure where a Bayesian approach in the form of estimation filters is utilized. We also consider some problems associated with the Measurement and Data Association block.

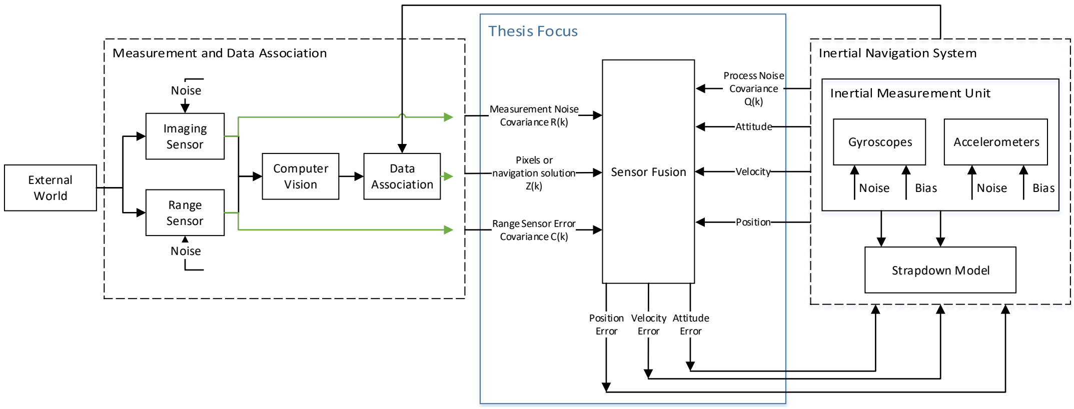


Figure 1.2: Sub-systems of Visual Aided Inertial Navigation. The Thesis Focus is indicated in the center as the sensor fusion block.

Within this context, the thesis contributions can be summarized as follows:

- We develop a carefully constructed scenario based simulation environment where a realistic camera model is combined with IMU data from a flight simulator. The IMU data itself is borrowed from an external simulator that uses precise strapdown equations to transform platform flight data into IMU data.

- We make use of the multi-sensor data supplied by our simulator to comparatively study Bayesian Gaussian filters (namely Extended Kalman and Unscented Kalman Filters) to evaluate data fusion performance,

- We also study these filters for two different VINS structures, namely Map Based and Mapless VINS,

- We consider performance limiting factors, such as number of landmarks, number of tracked frames, camera noise as well as landmark position uncertainty,

- We propose a novel Gaussian-Sum Filter for the considered problem dimension and comparatively study its performance,

- We propose a feature selection algorithm to deal with the fast moving camera problem on straight flight profile.

In our study we assume that there exist no data association errors. This assumption can be justified for the considered problem domain which can make use of available navigation solution and its uncertainties to associate the data from different images. Data association errors can be avoided by applying measurement gating and sample consensus algorithms.

Although real flight tests are not performed due to lack of available facilities, a realistic simulation environment is implemented. Hence we believe that proposed methods are realizable and conducted analyses are generalizable for having a stable and accurate navigation solution on a VINS.

1.3 Outline of Thesis

In the introduction part, significance of the thesis problem was discussed, scope and contributions of this study was given. The rest of the thesis is organized as follows:

In Chapter 2, a detailed review of related work is presented.

In Chapter 3, inertial navigation system equations, perturbation analysis to obtain error states for indirect filters and inertial measurement unit model are given.

In Chapter 4, the sensor fusion methods are explained. Camera model is presented and a feature region selection method is also proposed in this chapter.

In Chapter 5, followed methodology is explained in detail. Specifications of simulated UAV, IMU and camera models are given. Results of comparative evaluation of sensor fusion filters and detailed analysis of selected parameters are presented.

In Chapter 6, results are summarized and studies which can be conducted after this thesis are stated.

CHAPTER 2

LITERATURE REVIEW

VINS is an estimation theory problem which is a joint study of control and statistical signal processing theory. In VINS, navigation solution errors of INS are tried to be estimated and corrected by the information extracted from visual measurements. In order to establish a VINS, 5 main subproblems shown in Figure 2.1 need to be addressed. Studies in literature can be categorized according to solutions they suggest for one or more of these subproblems.

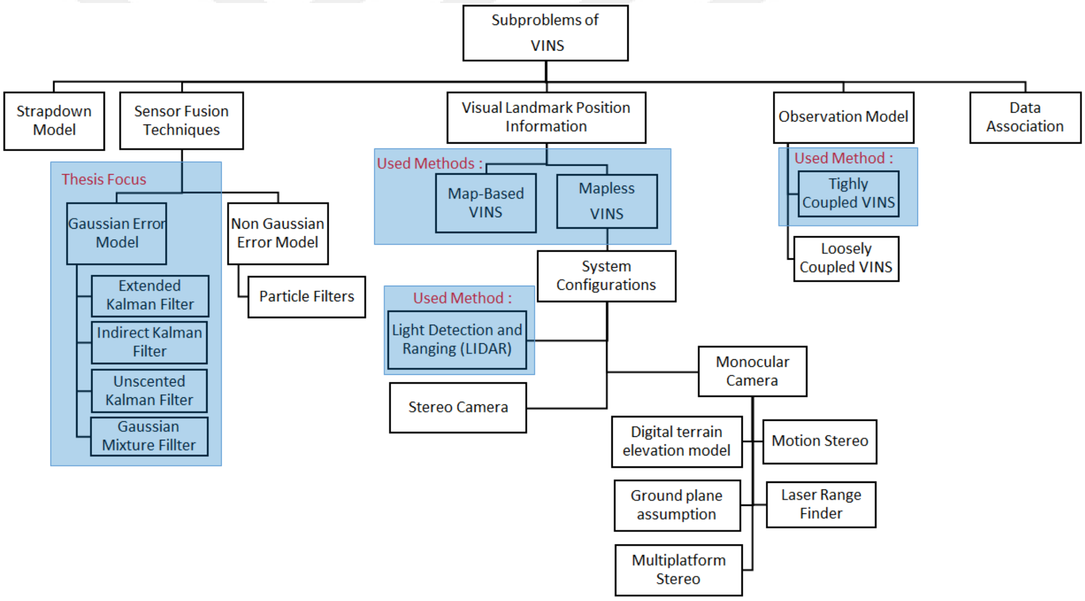


Figure 2.1: Subproblems of VINS

VINS techniques can be classified into two categories according to existence of visual landmark's position information before operation. The problem of recovering the navigation solution of the mobile platform with respect to landmarks with known positions is referred as map-based VINS. In map-based VINS, images of terrain under trajectory of UAV or descriptors of position fixed landmarks are available to the mobile platform before starting navigation operation. This approach requires a pre-planning phase. "Digital Scene Matching Area Correlator (DSMAC)" is a map-based

VINS, proposed in [5]. DSMAC uses position fixed terrain images for correcting platform's navigation solution. Images captured during flight are correlated with a terrain image database, loaded to mobile platform before flight. Map-Based VINS suffers from low precision at data association. DSMAC uses static morphological properties of position fixed scenes and has extensive preplanning phase before flight to solve low data association precision problem. Accuracy of the DSMAC is suggested as few tens of meters [18]. Map-Based VINS using position fixed images suffers from memory and high computational complexity problems. In order to solve these problems, feature point based solutions are proposed in literature [41], [8]. Compared to storing terrain images of all of the trajectory, feature based solutions have more abstract representation of a map. But they may suffer from data association problems.

Having position information of visual landmarks before flight might not be possible for all applications. Moreover, representation of this information like position fixed images or feature descriptors might not be accurate enough to match during operation. Additionally information provided by the map, like position of visual landmarks might not provide information with enough accuracy to fulfill application's navigation solution precision criteria. Therefore, a more standalone approach referred as mapless VINS is proposed as an alternative to map-based VINS. This method shows similarities to Structure from Motion (SfM) techniques [45]. But rather than creating a visual map, mapless VINS focuses on refinement of navigation solution of the mobile platform. Mapless VINS creates temporary feature point maps and applies Simultaneous Localization and Mapping(SLAM) algorithm to refine navigation solution of the mobile platform while refining temporary maps. Numerous studies in can be found in literature about mapless VINS.(see e.g. [52], [55], [6], [14])

VINS can be further categorized into "tightly-coupled" and "loosely-coupled" versions depending on how the navigation solution(state) is transferred into camera sensor measurements (i.e. the type of measurement model).

In loosely coupled VINS, translational and rotational displacements of the camera are calculated from pixel measurements. Loosely coupled integration is studied under visual odometry and egomotion topics [45]. In loosely coupled Map-Based VINS, problem of estimating position and attitude of the mobile platform is studied under

"Perspective N Point Problem(PnP)" or "Perspective N Line Problem (PnL)" topics. PnP uses points while PnL uses lines as reference visual inputs. To solve PnP or PnL problems, iterative or non-iterative approaches are applied for minimizing the defined cost function which generally consist of constrained reprojection error between model and measurement. Non-iterative techniques shows more sensitivity to image noise compared to iterative techniques while iterative techniques have the risk of not converging to global minimum [56]. In loosely coupled mapless VINS, camera translation and rotation is calculated between sequential frames using feature point matches [50], [43], [47].

While loosely coupled VINS calculates the camera translation and rotation explicitly, tightly coupled VINS uses perspective camera model as observation model. Perspective camera model relates the navigation solution and pixel measurements directly in a nonlinear fashion. In tightly coupled map-based VINS, feature points with known positions are used [27]. In tightly coupled mapless VINS, feature points used as landmarks are extracted during flight. These features are matched between consecutive frames. Then updated pixel values are directly used without explicitly calculating camera movement. (see e.g. [52], [22], [23])

Computational complexity of tightly coupled approach is far less than loosely coupled approach. In [7], Chu et. al. compares the performance of tightly and loosely coupled VINS. Results show that tightly coupled approach can provide more accurate navigation solution while loosely coupled approach is more stable.

Scale ambiguity is a problem that needs to be addressed in mapless VINS studies. A representation of this problem can be found in Figure 2.2. Given two images for consecutive measurements from t_i moment and t_{i+1}, any camera movement from line l_1 to l_2 indicated on Figure 2.2 is indistinguishable without depth information.

In order to solve scale ambiguity problem, feature points must be introduced to VINS in 3D. In map-based VINS, this information is already provided to system before flight. Altought, a map is not provided in mapless VINS, temporary feature point maps must be constructed in 3D. Since imaging sensor projects 3D world into a 2D plane, range information must be retrieved for reprojecting pixel coordinates to 3D

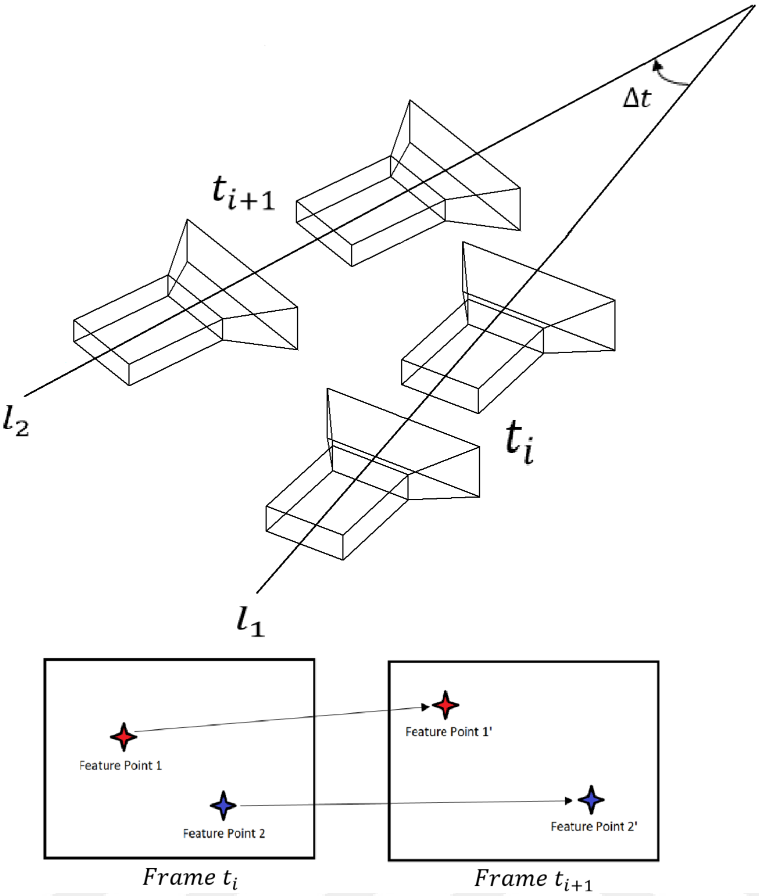


Figure 2.2: Scale Ambiguity Problem

world. In VINS, proposed solutions for retrieving range information can be classified into two categories as estimation and measurement based methods.

Using stereo cameras for retrieving range information in VINS is an option proposed in [52] [50] [31] . Range estimation using stereo camera setup may suffer from stereo baseline limits and feature matching errors. Stereo baseline (distance between camera pairs) limits the maximum range that can be calculated using stereo geometry. This can limit the usage of stereo cameras for the applications where platform and imaged terrain is distant like UAVs. Data association errors can also cause range estimation errors.

Usage of stereo camera setup might not be possible for all UAV applications because of size, cost and power consumption constraints. Different depth estimation solutions are proposed in literature for the mobile platform where only a monocular camera exists.

Range can be calculated under the assumption that ground level is flat. This assumption can be valid under assumption that platform is flying at relatively high altitudes [58], [8], [55]. Barometric altimeter can also be used for reducing the range estimation uncertainties where ground level is assumed to be flat. Therefore, studies making flat ground level assumption, mostly uses a UAV having barometric altimeter. Although using multiple sensors, flat ground level assumption causes high range estimation uncertainties.

Another alternative for depth estimation when only monocular camera exists on platform uses Digital Terrain Model (DEM) [15], [35], [29]. DEM maps contains 3D representation of terrain surface. Using the intersection of line of sight (LoS) of camera and DEM map, range can be calculated. Range estimation uncertainty depends on the fixed uncertainty of the map as well as the position and orientation uncertainties of the mobile platform.

Motion stereo is a range estimation method when multiple cameras are not available on the mobile platform. Stereo geometry is constructed with the help of estimated position and attitude of the mobile platform. Madison et. al. derives motion stereo equations and their uncertainties for a VINS system in [14]. In this study, failure modes of motion stereo are also discussed. [30] derives the relation between three camera views, called Trifocal Tensor, and motion stereo to solve scale ambiguity problem. In motion stereo, depth estimation uncertainty depends on position and orientation uncertainties of the mobile platform while uncertainty of traditional stereo vision with two cameras depends only on uncertainty of placement of cameras.

Another interesting method is using multiple platforms to construct a stereo geometry for range estimation. A *federated filter* framework with master and local filters by cooperating landmark information from multiple platforms is proposed in [2]. Stereo geometry between two platforms are also used for estimating the range information. In [48], a mapless VINS without using SLAM approach is proposed using range estimation from multiple vehicles.

Range uncertainty is directly proportional to the mobile platform's navigation solution's uncertainty in motion stereo, DEM based range estimation and multi-platform stereo. Using a range sensor enables us to eliminate the effects of the mobile platform's navigation solution's uncertainty. Two different sensor types are proposed in VINS studies focused on UAVs. Laser Range Finder is a sensor using laser pulses to measure distance between sensor and targeted point. It provides a single measurement from a scene. Since generally camera and range finder optics are geometrically aligned, range measurement belongs to a pixel at the line of sight of camera. In [57], a gimbaled laser range finder method is proposed for solving scale ambiguity on VINS. Range information of pixels not receiving range measurement is estimated using imaging sensor's intrinsic parameters and height assumptions on terrain. Therefore range estimation errors cannot be avoided using laser range finder.

LiDAR (Light Detection and Ranging) is an electronic instrument which can provide depth information of all individual pixels and an intensity image for a scene. LiDAR sensors are optimal for estimating the feature point's 3D position in the sense that they provide full range information of the scene without any range estimation errors. Because of this advantage, LiDAR is proposed method in VINS literature [10], [11], [49] and also used in this study for range estimation.

There exist two sources of information in VINS. First, INS is the main source of information to form navigation solution. The second source is the measurements which aids to INS with information obtained from camera and data association. The information from these two sources must be fused by using a mathematical framework which is referred as sensor fusion(or information fusion) filter. Sensor fusion filters try to estimate the unknown probability density function of the state recursively over time using incoming measurements and mathematical process model by applying probabilistic approaches [9]. Methods proposed in literature differs in terms of stability, state error, computational complexity and assumptions where filters are designed to operate under.

Kalman filter(KF) is one of the most common sensor fusion methods used in many applications. KF assumes that the process noise and measurement noise are zero mean Gaussian white noise sequence. Initial state, process noise and measurement noise are assumed to be jointly independent. State equations and measurement model is assumed to be linear. Although linearity assumption doesn't hold for most of the real world problems, KF is convergent and optimal in the sense of state error if assumptions are satisfied.

In order to extend the KF to nonlinear real world problems, a solution is taking the linear part of the system by linearizing it around estimated state. This approach is referred as Extended Kalman Filtering (EKF). Linearization errors causes suboptimal performance in terms of state errors and stability in EKF. For highly nonlinear systems, the effect of linearization errors are greater compared to simpler nonlinear systems, since fidelity of the system representation is decreased more by linearization. Visual aided Inertial Navigation Systems are nonlinear systems in terms of both state equations and measurement model. Using Extended Kalman Filter is proposed in literature by many studies. Studies using EKF as sensor fusion filter in VINS propose different solutions for increasing the stability, decreasing the navigation solution errors and computational cost. As an example; linearization errors in EKF based VINS causes unobservability along certain directions. These directions differs for assumptions and measurement model of the used EKF framework. In [32], Roumeliotis et. al. studies the observability properties of standard EKF based Tightly coupled VINS and proposes a observability constrained VINS(OC-VINS) to avoid gaining spurious information which can lead inconsistency in the filter. Linearization errors may reduce the observable directions. Number of unobservable directions are forced to stay same by applying an observability constrain which modifies measurement and state transition matrices in OC-VINS.

In [52], Veth proposed using Jacobian matrices, uncertainties of navigation solution and uncertainties of landmark positions to define an ellipsis around the feature to be matched which limits the search region and decrease the number of wrong matches. This increase on performance of data association also refines the performance of VINS. The purposed method also decreases the computational cost of feature matching, since feature to be matched in previous image is searched in just a small part of the image. If this method is not applied, ideally distance between descriptors of all features in previous and current images should be calculated. This yields poor data association performance and high computational cost.

In order to cope with linearization errors introduced to system with EKF, an alternative approach based on Unscented Transform (UT) is proposed by [54]. This approach is referred as Unscented Kalman Filter (UKF). UKF still assumes that process noise and measurement noise are zero mean Gaussian white noise sequence and initial state, process noise and measurement noise are assumed to be jointly independent but eliminates assumption of having linear state equations and measurement model. In every prediction step, state density is sampled at multiple points with UT. These sample points are referred as sigma points. Sigma points are propagated via nonlinear equations of system and weighted sum of these sigma points are used for calculating the mean and uncertainty of propagated state. In [12], Ebcin compared the EKF with UKF for a tightly coupled mapless VINS. Results shows that UKF over performs EKF in terms of increasing stability and decreasing navigation solution error.

EKF and UKF makes Gaussianity assumption for the state distribution and limits the number of components that can be utilized for modeling the state distribution. These assumptions and limitations can be eliminated using weighted sum of Gaussian components as proposed in [1]. This approach is referred as Gaussian Mixture Filter (GMF) or Gaussian Sum Filter (GSF) in literature. GMF also makes linearity assumption about system. From this point of view, it corresponds to utilizing weighted sum of multiple Extended Kalman Filters.

In [34], Kwok. et al. proposes using GMF for 2D mobile robot localization and mapping. Number of Gaussian components for minimizing the computational cost introduced as a result of using multiple components is tried to be held minimum with a method called sequential probability ratio test in this study.

Components of GMF are not dynamically generated as in UKF. Weights of some components are dominated by other components after a few weight updates. In this case propagating components with lower weights is not efficient. As the number of dominant component increases, filter loses advantage of having multiple components. This is referred as sample degeneracy problem. In order to solve this problem in GMF, applying resampling is proposed in [37].

For the systems neither linear nor having zero mean Gaussian errors, particle filters (PF) are proposed. PF samples the probability density of current state to generate particles representing this density. These particles are propagated according to the state dynamics and weighted according to measurements. The number of particles allowed in system determines the accuracy of estimated state density. But, the required number particles increases exponentially with the state dimension. Roughly, each particle corresponds to a single Kalman filter in terms of computational cost added to navigation computer. In a 6 DOF mobile platform using SLAM technique, $(9+3\times k)$ states, where k representing number of landmarks, exists. Therefore, unless a marginalization technique like Rao-Blackwellized Particle Filtering is not used, PF is not suitable for real time implementation of SLAM based VINS. Hence PFs will not be studied under this thesis.

CHAPTER 3

REVIEW OF INERTIAL NAVIGATION SYSTEM MODELING

Inertial navigation system (INS) is a system composed of hardware and algorithms to cumulatively calculate the navigation solution of a platform. INS consists of two main blocks: navigation computer and inertial sensors.

Navigation computer is the electronic system which calculates the navigation solution with the help of a navigation technique. If external information is not provided, navigation techniques use dead reckoning method. Dead reckoning is the process of calculating the mobile platform's position, attitude and velocity of the mobile platform using previous navigation solution and estimated change on this solution over a time period.

Change on the navigation solution over a time period can be obtained either in two ways. First approach uses physical dynamics of the mobile platform which *indirectly* relates physical dynamics and navigation solution. For example, using diameter of wheels and number of rotations of motor shaft, linear displacement and velocity of a wheeled robot can be calculated. For a flying vehicle, this process would be more complex.

The other approach uses inertial measurement units (IMU). IMU is composed of gyroscopes and accelerometers to *directly* measure translational acceleration and angular rate of the mobile platform . This sensor is rigidly attached, in other words "strapped down" to the mobile platform. Measurements from this sensor can be integrated to calculate current navigation solution. Navigation technique using this approach is referred as "strapdown" inertial navigation system in literature [51]. Modeling complexity of strapdown INS is far less than first approach since physical dynamics of the mobile platform are not considered the while deriving navigation equations. This enables us to keep state transition equations more precise and simpler. Because of these advantages, navigation state transitions are modeled using strapdown equations in UAVs.

Strapdown equations consist of nonlinear differential equations of position, velocity and attitude of the platform under accelerometer and gyroscope measurements of IMU. These equations will be referred as state dynamics or state dynamic equations for the rest of this thesis. IMU measurements are transformed to related coordinate frames and cumulatively integrated to calculate the current states. While deriving strapdown equations, Coriolis force and centripetal force of the Earth must be taken into account in order to navigate over long distances and durations at high precision.

An important part of VINS consists of the Inertial Navigation equations, often called "INS mechanization"[51]. INS equations define the dynamics of navigation solution of the mobile platform under inertial measurements. INS equations with their error perturbations constitute the part of VINS shown by Eq. 3.1. A method referred as the Indirect (Error State) Kalman Filter attempts to estimate the errors of INS for each time step and correct the corresponding navigation states accordingly. This further requires modeling the IMU error sources and their interaction with INS equations. In this chapter, we are summarizing the main results related with these methods in order to facilitate the development of the Visual aided approach to this problem (which strongly relates with the measurement equation Eq. 3.2).

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1} + \mathbf{w}_k) \tag{3.1}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.2}$$

3.1 Background Information

3.1.1 Reference Frames

Reference frames describe the coordinate system which navigation information belongs. In a navigation system, rotation and translation must be given with the information of reference axes. For example, IMU provides measurements with respect to the body of the mobile platform but platform navigates with respect to the Earth.

Right-handed reference frames used in this study are explained in this section.

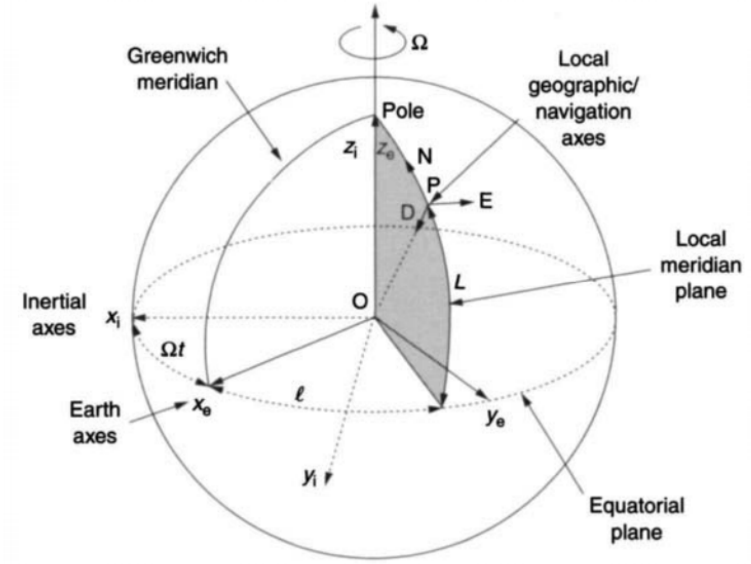


Figure 3.1: Inertial, Earth and Navigation Frames [51]

- ***The inertial frame (i-frame)***: Origin of i-frame is located at the center of the Earth. This reference frame is defined by x_i, y_i and z_i axes. Axes are *non-rotating* with respect to the fixed stars. z_i axis is coincident with the Earth's polar axis. Gyroscopes provide turn rate of the body frame with respect to i-frame.

- ***The Earth centered Earth fixed frame (e-frame, ECEF)***: Origin of the e-frame is located at the center of the Earth. e-frame is defined by x_e, y_e ans z_e axes. z_e axis is coincident with the Earth's polar axis. The axis x_e lies along the intersection of Greenwich meridian with the Earth's equatorial plane. The Earth frame *rotates*, with respect to the inertial frame, at a rate of Ω about the axis z_i [51].

- ***The navigation frame (n-frame, NED)***: Origin of the n-frame is located on the mobile platform for attitude and velocity representations. Axes of n-frame lie along north, east and down (NED) direction with respect to the mobile platform. Hence axes rotate as the mobile platform moves. n-frame is selected as the reference frame for attitude and velocity dynamics of the platform in this study. Position of the mobile platform in n-frame, $\boldsymbol{p}^n$, is calculated with respect to a fixed point in geodesic coordinate system rather than an origin fixed on the mobile platform. Directions of axes still show NED but determined according to the position of the reference point on the Earth. This frame will be used for managing the relations between landmark and platform position in a coordinate system using metric unit.

- ***The body frame (b-frame)***: Origin of the b-frame is located on the mobile platform. b-frame is defined by x_b, y_b and z_b axes. x_b axis lies along nose and y_b lies along right wing of the platform as shown in Figure 3.2. Accelerometers provide the specific force measurements which is indicated by vector quantity $\mathbf{f}^b$ in body frame.

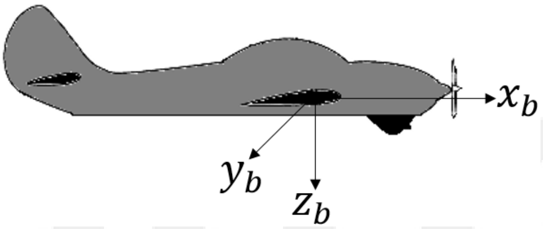


Figure 3.2: Body Reference Frame

- ***The camera frame (c-frame)***: Origin of the c-frame is located at camera lens. Z axis lies along camera LoS and y-axis lies along top of camera as shown in Figure 3.3.

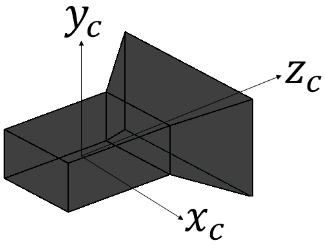
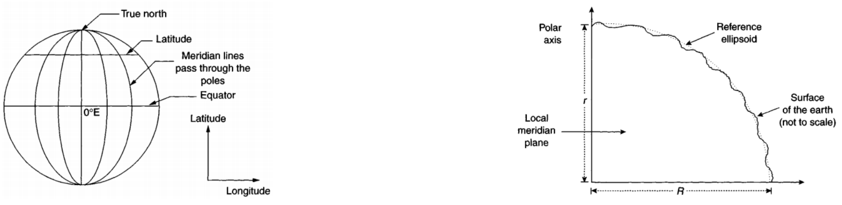


Figure 3.3: Camera Frame

- ***Geographic Coordinate System***: In order to navigate over long distances and durations at high precision on the Earth, position data must be expressed in a coordinate system which models the shape of the Earth. Such an alternative is Geographic Coordinate System. Geographic Coordinate System is a coordinate system where position data is expressed in latitude, longitude and height above sea level. Longitude and latitude are angles measured from the Earth's center to a point on the Earth's surface. Lines of constant latitude and longitude are called parallels and meridians. [51]. World Geodetic System of 1984 (WGS-84) is used as Geographic Coordinate System in this study. Parameters of the WGS-84 Earth model is defined as in Table 3.1.



(a) Latitude and Longitude Representations [51] (b) Semi-major and Semi Minor Axes of Earth [51]

Figure 3.4: Geographic Coordinate System

Table 3.1: WGS-84 model [51]

Length of the semi-major axis,	R	6378137.0m
Length of the semi-minor axis,	$r = R(1-f)$	6356752.3142m
Flattening of the ellipsoid,	$f = (R-r)/R$	1/298.257223563
Major eccentricity ofthe ellipsoid,	$e = [f(2-f)]^{1/2}$	0.0818191908426
Earth's rate	Ω	7.292115×10^{-5} rad/s
Meridian radius of curvature ,	R_N	$R_N = \frac{R(1-e^2))}{(1-e^2 sin^2 L)^{3/2}}$
Transverse radius of curvature,	R_E	$R_E = \frac{R}{(1-e^2 sin^2 L)^{1/2}}$
Latitude of the platform	L	

3.1.2 Direction Cosine Matrices

A Direction Cosine Matrix (DCM) is 3x3 matrix used for transforming a vector from one reference frame to another. Basically, DCM rotates the vector about roll,pitch and yaw directions of operand vector respectively. A DCM transforming a vector from b-frame to n-frame is given in Eq. 3.3.

$$\mathbf{C}_b^n = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{31} & c_{33} \end{bmatrix} \tag{3.3}$$

$$\mathbf{f}^n = \mathbf{C}_b^n \mathbf{f}^b \tag{3.4}$$

Properties of direction cosine matrices are very useful for deriving equations in navigation systems. These properties are given below:

$$\mathbf{C}_b^n = (\mathbf{C}_n^b)^T = (\mathbf{C}_n^b)^{-1} \tag{3.5}$$

$$\mathbf{C}_b^n = \mathbf{C}_e^n \mathbf{C}_b^e \tag{3.6}$$

$$\det(\mathbf{C}_b^n) = 1 \tag{3.7}$$

DCMs for rotating a vector about yaw, pitch and roll axes separately are given below:

$$rotation\ \psi\ about\ z\ axes,\ yaw,\ \mathbf{C}_1 = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

$$rotation\ \theta\ about\ y\ axes,\ pitch, \mathbf{C}_2 = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 0 & 1 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{3.9}$$

$$rotation\ \phi\ about\ x\ axes,\ roll,\ \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{3.10}$$

To obtain a DCM transforming a vector from one reference frame to another, order of rotations applied about independent axes must be followed on multiplications of DCMs. A DCM, transforming a vector in n-frame to b-frame can be composed as in Eq. 3.11 if yaw, pitch and roll convention is used respectively for the rotation sequence.

$$\mathbf{C}_n^b = \mathbf{C}_3 \mathbf{C}_2 \mathbf{C}_1 \tag{3.11}$$

Similarly using first property of DCM stated in Eq. 3.5, $\mathbf{C}_b^n$ can be written as:

$$\mathbf{C}_b^n = \mathbf{C}_1^T \mathbf{C}_2^T \mathbf{C}_3^T \tag{3.12}$$

$$\mathbf{C}_b^n = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 0 & 1 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix} \tag{3.13}$$

Euler angles can also be derived from DCMs. Using numeric representations of elements of DCM given in 3.3, equations of Euler angles are written below:

$$\phi = arctan[\frac{c_{32}}{c_{33}}] \tag{3.14}$$

$$\theta = arctan[-c_{31}] \tag{3.15}$$

$$\psi = arctan[\frac{c_{21}}{c_{11}}] \tag{3.16}$$

3.2 Strapdown System Modeling

Strapdown equations model the dynamics of navigation information under IMU measurements in detail. Strapdown model includes transforming calculated or measured quantities to related frames and models: change of gravitational field over the Earth, centripetal force experienced by the mobile platform because of rotation of the Earth, geometric shape of the Earth, effects of moving w.r.t. a rotating frame. This extensiveness of the strapdown model enables the mobile platform to navigate in high precision over long distances and durations.

In strapdown equations, differential equations of three navigation variables each having 3 dimensions are derived. These variables are:

- attitude: roll ϕ, pitch θ and yaw ψ angles from body to navigation frame
- velocity: north v_N, east v_E and down v_D velocities in navigation frame
- position: L latitude, l longitude and h height above sea level in geographic coordinate system.

Representing position, velocity and attitude as p, v and θ respectively, state vector for a more generic control system would be expected as follows:

$$x = \begin{bmatrix} \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} & \boldsymbol{v} & \dot{\boldsymbol{v}} & \boldsymbol{p} & \dot{\boldsymbol{p}} \end{bmatrix}^T \tag{3.17}$$

Because of $\dot{\boldsymbol{p}} = \boldsymbol{v}$, the term $\dot{\boldsymbol{p}}$ is not included as an extra state element in strapdown INS. Since $\dot{\boldsymbol{v}}$ is the translational acceleration supplied by accelerometers and $\dot{\boldsymbol{\theta}}$ is the turn rate supplied by gyroscopes, these terms can be included in input term $\boldsymbol{u}$ of the control system. Therefore INS state vector is:

$$x = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{v} & \boldsymbol{p} \end{bmatrix}^T \tag{3.18}$$

In the following section, strapdown equations representing state equations of navigation system 3.1 will be summarized. One can refer to [51] for details of derivations of strapdown equations. Since these derivations are out of focus of this thesis, only equations and short explanations of equation elements will be presented.

3.2.1 Attitude Dynamics

Attitude of the mobile platform is tracked in Euler angles from body to navigation frame. Attitude is propagated in DCM form as shown in 3.19:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \tag{3.19}$$

where:

$$\boldsymbol{\Omega}_{nb}^b = [\times \boldsymbol{\omega}_{nb}^b] \tag{3.20}$$

$$\boldsymbol{\Omega}_{nb}^b = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & \omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3.21}$$

Explicit equation of $\boldsymbol{\omega}_{nb}^b$, turn rate of b-frame w.r.t. n-frame, is written as:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b[\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n] \tag{3.22}$$

where $\boldsymbol{\omega}_{ib}^b$ is turn rate of b-frame w.r.t. the i-frame and directly obtained from gyroscope measurements, $\boldsymbol{\omega}_{ie}^n$ is turn rate of e-frame w.r.t i-frame and it is written as:

$$\boldsymbol{\omega}_{ie}^{n} = [\Omega cosL \;\; 0 \;\; -\Omega sinL]^{T} \tag{3.23}$$

where Ω is the turn rate of Earth given in Table 3.1. $\boldsymbol{\omega}_{en}^{n}$ is turn rate of the n-frame w.r.t. e-frame. $\boldsymbol{\omega}_{en}^{n}$ can be written explicitly as shown in 3.24.

$$\boldsymbol{\omega}_{en}^{n} = \begin{bmatrix} \frac{v_E}{R_E+h} & \frac{-v_N}{R_N+h} & \frac{-v_E tanL}{R_N+h} \end{bmatrix}^{T} \tag{3.24}$$

where L is latitude and h is altitude of the platform. Equations of meridian and traverse radius of curvature, which are R_E and R_N respectively, are given in Table 3.1.

The term adding up $\boldsymbol{\omega}_{ie}^{n}$ and $\boldsymbol{\omega}_{en}^{n}$ defines the effect of navigating w.r.t. a rotating frame which is referred as Coriolis acceleration and also defines the effect of centripetal acceleration on attitude dynamics of platform originated from Earth's rotation.

After calculating $\boldsymbol{C}_{b}^{n}$ using equations given above, roll, pitch and yaw angles of platform can be obtained separately from calculated DCM using Eq. 3.14 , 3.15 and 3.16.

3.2.2 Velocity Dynamics

Dynamics of velocity are tracked in n-frame in meter/sec unit along north, east and down directions respectively. Velocity dynamics is written as below:

$$\dot{\mathbf{v}}_{e}^{n} = \mathbf{C}_{b}^{n}\mathbf{f}^{b} - [2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}] \times \mathbf{v}_{e}^{n} + \mathbf{g}_{l}^{n} \tag{3.25}$$

$$\mathbf{v}_{e}^{n} = [v_N \; v_E \; v_D]^{T} \tag{3.26}$$

where the term $[2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}]$ defines the effect of navigating w.r.t. a rotating frame. $\mathbf{f}^{b} = [f_N \; f_E \; f_D]^{T}$ represents specific force(acceleration) measured by accelerometers. . The term $\mathbf{g}_{l}$ represents the local gravity vector including mass attraction force and centripetal force caused by rotation of the Earth is written as:

$$\mathbf{g}_{l} = [0 \; 0 \; \mathbf{g}_{h}]^{T} \tag{3.27}$$

$$\mathbf{g}_{h} = \frac{\mathbf{g(0)}}{(1 + h/R_0)^2} \tag{3.28}$$

$$\mathbf{g(0)} = 9.780318(1 + 5.3024 \times 10^{-3} sin^2 L - 5.9 \times 10^{-6} sin^2 2L) \tag{3.29}$$

where L is latitude of the mobile platform.

3.2.3 Position Dynamics

Position dynamics can be easily expressed after deriving velocity dynamics. Transferring velocity information in n-frame to geographic coordinate system yields:

$$\dot{L} = \frac{v_N}{R_N + h} \tag{3.30}$$

$$\dot{\ell} = \frac{v_E secL}{R_E + h} \tag{3.31}$$

$$\dot{h} = -v_D \tag{3.32}$$

where L is latitude position of the platform, ℓ is longitude position of the platform, h is altitude above sea level.

3.3 Perturbation Analysis

In a control system, small "perturbations" about a state can be approximated by using Taylor Series. Perturbation Analysis is a method based on Taylor Series approximation and used for obtaining dynamics of state errors.

Consider continuous nonlinear differential equation of a state space representation of system:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{3.33}$$

Because of the input noise and its cumulative effect on states, state values are represented as the sum of true value of state $\bar{\boldsymbol{x}}(t)$ and an error term $\delta \boldsymbol{x}$ as shown in Eq. 3.34. Also input is decomposed as true value and error term as shown in Eq. 3.35.

$$\boldsymbol{x}(t) = \bar{\boldsymbol{x}}(t) + \delta \boldsymbol{x} \tag{3.34}$$

$$\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t) + \delta \boldsymbol{u} \tag{3.35}$$

Writing state and input with small error "perturbations" Eq. 3.34, 3.35 to Eq. 3.33 yields:

$$\dot{\boldsymbol{x}} = f(\bar{\boldsymbol{x}}(t) + \delta\boldsymbol{x}, \bar{\boldsymbol{u}}(t) + \delta\boldsymbol{u}, t) \tag{3.36}$$

Applying Taylor Series approximation to Eq. 3.36 gives:

$$\dot{\boldsymbol{x}} = f(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t) + \boldsymbol{F}(t)\delta\boldsymbol{x} + \boldsymbol{B}(t)\delta\boldsymbol{u} + \boldsymbol{\epsilon} \tag{3.37}$$

where,

$$\boldsymbol{F}(t) = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\bar{\boldsymbol{x}}(t),\bar{\boldsymbol{u}}(t),t} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial f_n}{\partial \boldsymbol{x}_n} \end{bmatrix}\right|_{\bar{\boldsymbol{x}}(t),\bar{\boldsymbol{u}}(t),t} \tag{3.38}$$

$$\boldsymbol{B}(t) = \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{\bar{\boldsymbol{x}}(t),\bar{\boldsymbol{u}}(t),t} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{u}_1} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{u}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \boldsymbol{u}_1} & \cdots & \frac{\partial f_n}{\partial \boldsymbol{u}_n} \end{bmatrix}\right|_{\bar{\boldsymbol{x}}(t),\bar{\boldsymbol{u}}(t),t} \tag{3.39}$$

and ϵ term is linearization error, $\boldsymbol{F}(t)$ and $\boldsymbol{B}(t)$ are Jacobian matrices about true values of state and input respectively. But in real case, values $\bar{\boldsymbol{x}}(t)$ and $\bar{\boldsymbol{u}}(t)$ are never known as this was the problem from the beginning. Hence Jacobians are calculated about erroneous state values, since this is the only available information.

Eq. 3.37 can be decomposed as the dynamics of true state and dynamics of error terms as shown below:

$$\dot{\boldsymbol{x}} = \dot{\bar{\boldsymbol{x}}}(t) + \delta\dot{\boldsymbol{x}} \tag{3.40}$$

Since dynamics of only true state is:

$$\dot{\bar{\boldsymbol{x}}} = f(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t) \tag{3.41}$$

Using Eq. 3.37, Eq.3.40 and 3.41, error state dynamics can be written as:

$$\delta\dot{\boldsymbol{x}} \approx \boldsymbol{F}(t)\delta\boldsymbol{x} + \boldsymbol{B}(t)\delta\boldsymbol{u} \tag{3.42}$$

Knowing dynamics of state errors allows us to correct them using sensor fusion filters explained in Chapter 5.

3.3.1 Perturbation Analysis of Navigation Variables

Indirect filters attempt to estimate and correct navigation solution errors. Dynamics of errors must be derived using perturbation analysis for this estimation and correction process. The errors in INS's attitude, velocity and position arises from 3 main sources: bias and random walk of IMU, discretization error owing to derivation of strapdown equations in continuous time and state initialization errors before flight [18].

Effect of the state errors cannot be written as an additive term to state equations since propagation of states are modeled using differential equations. Hence, these errors have their own dynamics.

One can find detailed derivations of perturbation equations for navigation variables in [51]. Here only the equations will be given. In [51], perturbation equations are linearized by omitting second order error terms which is the interested case of this study.

Dynamics of attitude errors, $\boldsymbol{\psi}$, is expressed as:

$$\dot{\boldsymbol{\psi}} \approx -\boldsymbol{\omega}_{in}^{n} \times \boldsymbol{\psi} + \boldsymbol{\delta\omega}_{in}^{n} - \boldsymbol{C}_{b}^{n}\boldsymbol{\delta\omega}_{ib}^{n} \tag{3.43}$$

where $\boldsymbol{\omega}_{in}^{n}$ is the turn rate of n-frame w.r.t. i-frame and obtained by $\boldsymbol{\omega}_{in}^{n} = \boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}$. $\boldsymbol{\delta\omega}_{ib}^{n}$ is gyroscope's random walk vector.

Dynamics of velocity errors, $\boldsymbol{\delta v}$, is expressed as:

$$\boldsymbol{\delta\dot{v}}^{n} = [\times \boldsymbol{f}^{n}]\boldsymbol{\psi} + \boldsymbol{C}_{b}^{n}\boldsymbol{\delta f}^{b} \tag{3.44}$$

where $[\times \boldsymbol{f}^{n}]$ is skew-symmetric form of accelerometer measurements and $\boldsymbol{\delta f}^{b}$ is accelerometer's random walk vector.

After writing dynamics of velocity errors, dynamics of position errors in n-frame can be easily written as:

$$\boldsymbol{\delta\dot{p}}^{n} = \boldsymbol{\delta v}^{n} \tag{3.45}$$

Dynamics of position error in n-frame must be transferred to geodesic coordinate system, since position is tracked in geodesic coordinate system. Position of the mobile platform expressed in geodesic coordinate system will be represented by $\boldsymbol{p}^{llh}$.

$$\delta\dot{L} = \frac{1}{R_N + h}\delta v_N - \frac{v_N}{(R_N + h)^2}\delta h \tag{3.46}$$

$$\delta\dot{\ell} = \frac{1}{(R_E + h)cos(L)}\delta v_E + \frac{v_E \cdot tanL}{(R_E + h)cos(L)}\delta L - \frac{v_E}{(R_E + h)^2 cos(L)}\delta h \tag{3.47}$$

$$\delta\dot{h} = -\delta v_D \tag{3.48}$$

3.3.2 Error State Matrices

Error state equations are written in the matrix form for state space representation and discretized for later using in covariance propagation step of Bayesian filters. Using error state dynamics of strapdown equations Eq. 3.42 is written as:

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}(t)\delta\boldsymbol{x} + \boldsymbol{G}(t)\delta\boldsymbol{u} \tag{3.49}$$

where,

$$\boldsymbol{\delta x} = \begin{bmatrix}\delta\alpha & \delta\beta & \delta\gamma & \delta v_N & \delta v_E & \delta v_D & \delta L & \delta l & \delta h\end{bmatrix}^T \tag{3.50}$$

$$\delta\boldsymbol{u} = \begin{bmatrix}\delta\omega_x & \delta\omega_y & \delta\omega_z & \delta f_x & \delta f_y & \delta f_z\end{bmatrix}^T \tag{3.51}$$

Note that $\boldsymbol{\delta\omega}_{ib}^{n}$ and $\boldsymbol{\delta f}^{b}$ are gyroscope and accelerometer measurement errors respectively. Using Eqs. 3.43 and 3.44, $\boldsymbol{G}(t)$ which represents the input noise influence matrix and can be written using convention defined in Eq. 3.3 as follows:

$$\boldsymbol{G}(t) = \begin{bmatrix} -\boldsymbol{C}_b^n & 0 \\ 0 & \boldsymbol{C}_b^n \end{bmatrix} = \begin{bmatrix} -c_{11} & -c_{12} & -c_{13} & 0 & 0 & 0 \\ -c_{21} & -c_{22} & -c_{23} & 0 & 0 & 0 \\ -c_{31} & -c_{32} & -c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{11} & c_{12} & c_{13} \\ 0 & 0 & 0 & c_{21} & c_{22} & c_{23} \\ 0 & 0 & 0 & c_{31} & c_{32} & c_{33} \end{bmatrix} \tag{3.52}$$

Taking the derivative of error state transition equation's terms according to state variables, using the property of cross product defined in Eq. 3.53, chain rule and equations given in Section 3.3.1 , error state transition matrix $\boldsymbol{F}(t)$ can be calculated as in Eq. 3.54.

$$\boldsymbol{a} \times \boldsymbol{b} = [\times\boldsymbol{a}]\boldsymbol{b} = \begin{bmatrix} 0 & -\boldsymbol{a_3} & \boldsymbol{a_2} \\ \boldsymbol{a_3} & 0 & -\boldsymbol{a_1} \\ -\boldsymbol{a_2} & \boldsymbol{a_1} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix} \tag{3.53}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are operand vectors.

$$
\boldsymbol{F}(t) = \begin{bmatrix}
0 & \begin{matrix} -\Big(\Omega\cdot sin(L) \\ +\frac{v_E}{R_N+h} * tan(L)\Big) \end{matrix} & \frac{v_N}{R_N+h} & 0 & \frac{1}{R_E+h} & 0 & -\Omega\cdot sin(L) & 0 & -\frac{v_E}{(R_E+h)^2} \\
\begin{matrix} \Big(\Omega\cdot sin(L) \\ +\frac{v_E}{R_N+h}\cdot tan(L)\Big) \end{matrix} & 0 & \begin{matrix} \Big(\Omega\cdot cos(L) \\ +\frac{v_E}{R_E+h}\Big) \end{matrix} & -\frac{1}{R_N+h} & 0 & 0 & 0 & 0 & \frac{v_N}{(R_N+h)^2} \\
-\frac{v_N}{R_N+h} & \begin{matrix} -\Omega\cdot cos(L) \\ \\ -\frac{v_E}{R_E+h} \end{matrix} & 0 & 0 & -\frac{tan(L)}{R_N+h} & 0 & \begin{matrix} \Big(-\Omega\cdot cos(L) \\ -\frac{v_E}{(R_N+h)\cdot cos(L)^2}\Big) \end{matrix} & 0 & v_E\cdot\frac{tan(L)}{(R_N+h)^2} \\
0 & -f_D & f_E & \frac{v_D}{R_N+h} & \begin{matrix} -2\cdot\Big(\Omega\cdot sin(L) \\ +\frac{v_E}{R_N+h}\cdot tan(L)\Big) \end{matrix} & \frac{v_N}{R_N+h} & \begin{matrix} -v_E\cdot\Big(2\cdot\Omega\cdot cos(L) \\ +\frac{v_E}{(R_N+h)\cdot cos(L)^2}\Big) \end{matrix} & 0 & \begin{matrix} \frac{1}{(R_N+h)^2}\cdot \\ \Big(v_E^2\cdot tan(L) \\ -v_N\cdot v_D\Big) \end{matrix} \\
f_D & 0 & -f_N & \begin{matrix} \Big(2\cdot\Omega\cdot sin(L) \\ +\frac{v_E}{R_N+h}\cdot tan(L)\Big) \end{matrix} & \begin{matrix} \Big(\frac{v_D}{R_E+h} \\ +v_N\cdot\frac{tan(L)}{R_N+h}\Big) \end{matrix} & \begin{matrix} (2\cdot\Omega\cdot cos(L) \\ +\frac{v_E}{R_E+h}) \end{matrix} & \begin{matrix} \Big(2\cdot\Omega\cdot \\ (v_N\cdot cos(L) \\ -v_D\cdot sin(L)) \\ +v_N\cdot\frac{v_E}{(R_N+h)\cdot cos(L)^2}\Big) \end{matrix} & 0 & \begin{matrix} -\frac{v_E}{(R_E+h)^2}\cdot \\ \Big(v_N\cdot tan(L)+v_D\Big) \end{matrix} \\
-f_E & f_N & 0 & -2*\frac{v_N}{R_N+h} & -2\cdot(\Omega\cdot cos(L)+\frac{v_E}{R_E+h}) & 0 & 2\cdot\Omega\cdot v_E\cdot sin(L) & 0 & \frac{v_N^2}{(R_N+h)^2}+\frac{v_E^2}{(R_E+h)^2} \\
0 & 0 & 0 & \frac{1}{R_N+h} & 0 & 0 & 0 & 0 & -\frac{v_N}{(R_N+h)^2} \\
0 & 0 & 0 & 0 & \frac{1}{(R_E+h)\cdot cos(L)} & 0 & \frac{v_E\cdot tan(L)}{(R_E+h)\cdot cos(L)} & 0 & -\frac{v_E}{(R_E+h)^2\cdot cos(L)} \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0
\end{bmatrix} \tag{3.54}
$$

3.4 Discretization of Error State Matrices

Dynamics of a control system modeled in continuous time must be transferred to discrete time in order to implement the system in a real time digital processing unit such as FPGA, DSP etc. These processing units compute the values of states at discrete time instants. Therefore continuous time linear differential equation given in Eq. 3.49 is transfered to:

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \delta\boldsymbol{x}_k + \boldsymbol{w}_k \tag{3.55}$$

Since $\delta\dot{\boldsymbol{x}}$ represents the change on error state in an infinitesimal time interval and $\delta\boldsymbol{x}[k+1]$ directly represents the value of error state at next time step, error state transition matrix in continuous time $\boldsymbol{F}(t)$ and error state transition matrix in discrete time $\boldsymbol{F}_k$ are different.

For discretizing Eq. 3.49, we begin by taking integral over sampling period to calculate error state solution:

$$\delta\boldsymbol{x}(t) = \boldsymbol{\Phi}(t_0, t)\delta\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t_0, \tau)\boldsymbol{G}(\tau)\boldsymbol{w}(\tau)d\tau \tag{3.56}$$

where the sampled error state transition matrix is $\boldsymbol{\Phi}(t_0, t)$ is defined as the integral of error state transition matrix over sampling period. Under the assumption that $\boldsymbol{F}(t)$ and its integral over a sampling period commutes [24]:

$$\boldsymbol{F}(t)\int_{t_0}^{t} \boldsymbol{F}(\tau)d\tau = \left(\int_{t_0}^{t} \boldsymbol{F}(\tau)d\tau\right)\boldsymbol{F}(t) \tag{3.57}$$

$\boldsymbol{\Phi}(t_0, t)$ can be written as:

$$\boldsymbol{\Phi}(t_0, t) = e^{\int_{t_0}^{t} \boldsymbol{F}(\tau)d\tau} \tag{3.58}$$

Since IMU can provide outputs at high frequency, Eq. 3.57 holds for our application. Knowing that samples are taken at equal time steps, writing $t_0 = kT$, $t = (k+1)T$ and $\delta\boldsymbol{x}(kT) = \delta\boldsymbol{x}_k$ to Eq. 3.56 yields:

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}((k+1)T, kT)\delta\boldsymbol{x}_k + \int_{kT}^{(k+1)T} \boldsymbol{\Phi}(kT, \tau)\boldsymbol{G}(\tau)w(\tau)d\tau \tag{3.59}$$

The second term representing the effect of noise on error state solution in Eq. 3.59 can be written as:

$$\boldsymbol{w}_k = \int_{kT}^{(k+1)T} \boldsymbol{\Phi}(kT, \tau)\boldsymbol{G}(\tau)\boldsymbol{w}(\tau)d\tau \tag{3.60}$$

The discretized input noise covariance matrix $\boldsymbol{Q}_k$ must be calculated as the covariance of new noise sequence $\boldsymbol{w}_k$ as written below:

$$\boldsymbol{Q}_k = \int_{kT}^{(k+1)T} \boldsymbol{\Phi}(kT,\tau)\boldsymbol{G}(\tau)\boldsymbol{Q}\boldsymbol{G}^T(\tau)\boldsymbol{\Phi}^{'}(kT,\tau)d\tau \tag{3.61}$$

where $\boldsymbol{\Phi}^{'}(t_0,t)$ is the discretized transpose of error state transition matrix and shown as:

$$\boldsymbol{\Phi}^{'}(t_0,t) = e^{\int_{t_0}^{t} \boldsymbol{F}(\tau)^T d\tau} \tag{3.62}$$

Continuous time error state transition matrix and noise influence matrix can be assumed to be constant between sampling steps when sampling intervals are short. This reduces Eq. 3.58 and Eq. 3.62 to:

$$\boldsymbol{\Phi}_k^s = e^{\boldsymbol{F}(kT)T} \tag{3.63}$$

$$\boldsymbol{Q}_k^s = \int_{kT}^{(k+1)T} e^{\boldsymbol{F}(kT)[(k+1)T-\tau]}\boldsymbol{G}(kT)\boldsymbol{Q}\boldsymbol{G}(kT)^T e^{\boldsymbol{F}^T(kT)[(k+1)T-\tau]}d\tau \tag{3.64}$$

To be able to calculate the Eq. 3.63 and Eq. 3.64 in a discrete time computation unit, Taylor series approximation of matrix exponential can be used [36]. This approximation is written as:

$$\boldsymbol{F}_k = \boldsymbol{I} + \boldsymbol{F}(t_0)T + \frac{\boldsymbol{F}(t_0)^2T^2}{2!} + ... \tag{3.65}$$

$$\boldsymbol{G}_k = \boldsymbol{G}(t_0)\boldsymbol{Q}\boldsymbol{G}(t_0)^TT + \frac{(\boldsymbol{F}(t_0)\boldsymbol{G}(t_0)\boldsymbol{Q}\boldsymbol{G}(t_0)^T + \boldsymbol{G}(t_0)\boldsymbol{Q}\boldsymbol{G}(t_0)^T\boldsymbol{F}(t_0)^T)T^2}{2!} + ... \tag{3.66}$$

This concludes the derivation of discretized error state transition and input noise covariance matrices.

Note that discretization process can yield covariance matrices which are not positive semidefinite and asymmetric. In order to increase the stability of sensor fusion filter, these matrices can be forced to be positive semidefinite and symmetric. This can be achieved by using closest positive semidefinite and symmetric matrix to discretized matrix instead of directly using discretized matrix. A detailed method for calculating closest positive semidefinite and symmetric matrix is explained in Appendix B.

3.5 Modeling Inertial Sensor Measurements

IMU is a sensor combined of 3 orthogonal accelerometers and 3 orthogonal gyroscopes. Accelerometers provide translational accelerations in 3 orthogonal axes and gyroscopes provide turn rates about 3 orthogonal axes. IMU measurements are distorted by different error sources which causes error divergence of INS because of its cumulative nature. In this chapter, models of these error sources will be given.

Depending on navigation solution accuracy requirement of the application, different types IMUs can be used. Grade of an IMU specifies the magnitude of errors experienced by measurements. IMUs can be classified into 5 main categories as marine, navigation(or aviation), intermediate, tactical and automotive grade. Accuracy and cost gradually decreases from marine to automotive grade IMUs. An example table showing typical accelerometer and gyroscope biases for different grades of IMU can be found in Table 3.2

Table 3.2: Typical Accelerometer and Gyro Biases for Different Grades of IMU [18]

	Accelerometer Bias		Gyro Bias	
IMU Grade	mg	ms^{-2}	deg hr^{-1}	$rads^{-1}$
Marine	0.01	10^{-4}	0.001	5×10^{-9}
Aviation	$0.03 - 0.1$	$3 \times 10^{-4} - 10^{-3}$	0.01	5×10^{-8}
Intermediate	$0.1 - 1$	$10^{-3} - 10^{-2}$	0.1	5×10^{-7}
Tactical	$1 - 10$	$10^{-2} - 10^{-1}$	$1 - 100$	$5 \times 10^{-6} - 5 \times 10^{-4}$
Automotive	> 10	$> 10^{-1}$	> 100	$> 5 \times 10^{-4}$

Probability distributions of IMU error sources are crucial for navigation aid methods. For example Bayesian filters, like Kalman filters, makes zero mean Gaussian distribution assumption for statistical characteristics of errors, covariances representing the uncertainty on estimation is also affected by IMU measurement error distributions. Therefore, modeling IMU measurements and their error's probability distributions are important parts of VINS.

The magnitude of error sources is a major factor determining the stability and navigation solution error performance upper bounds of a navigation aid. Using IMUs with higher accuracy enables a filter to estimate navigation information with less error and have more convergent characteristic.

Tactical grade IMUs are commonly used in UAVs and aided with GPS. Hence, to be compatible with this common usage and observe the advantages offered by VINS in a more explicit way, a low cost tactical grade IMU which has specifications shown in Table 5.2 is modeled and used in simulations of this study.

Gyroscope and accelerometer errors have different models. Errors on gyroscopic measurements, $\boldsymbol{\delta\omega}$, can be written as in Eq. 3.67.

$$\begin{bmatrix} \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \end{bmatrix} = \boldsymbol{B_G} + \boldsymbol{B_g} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} + \boldsymbol{S_G} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \boldsymbol{M_G} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \boldsymbol{w}_G \qquad (3.67)$$

where $\boldsymbol{B_G}$ is fixed bias error, $\boldsymbol{B_g}$ is a matrix containing g-dependent bias coefficients relating acceleration of the mobile platform to turn rate errors, $\boldsymbol{S_G}$ is constant scale factor matrix multiplying true angular rates with a constant scale to model its effect on gyroscope errors, $\boldsymbol{M_G}$ is skew-symmetric matrix representing the mounting and cross-coupling error between axes of gyroscope. $\boldsymbol{w}_G$ is the 3x1 vector representing the gyroscope's random walk.

Errors on specific forces in other words measured accelerations, $\boldsymbol{\delta f}$ can be written as in Eq. 3.68.

$$\begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} = \boldsymbol{B_A} + \boldsymbol{S_A} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} + \boldsymbol{M_A} \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} + \boldsymbol{w}_A \qquad (3.68)$$

where $\boldsymbol{B_A}$ is fixed bias error, $\boldsymbol{S_A}$ is constant scale factor matrix multiplying true accelerations with a constant scale to model its effect on accelerometer errors, $\boldsymbol{M_A}$ is skew-symmetric matrix representing the mounting and cross-coupling error between axes of accelerometer. $\boldsymbol{w}_A$ is the 3x1 vector representing the accelerometer's random walk.

More detailed IMU error models can be found in [51]. But for this study, convention in VINS studies will be followed and only major error sources will be modeled in simulations. These error sources are bias and random walk vectors of gyroscopes and accelerometers.

Gyroscope and accelerometers bias vectors are assigned at every start-up of IMU sensor. In other words they are only drawn from a random distribution at the beginning of simulation. These vector are added to measurement as a constant error term. Because of this constant addition, IMU errors statistical distribution becomes a "non-zero" Gaussian distribution which violates the Kalman filter based sensor fusion algorithm's assumption. Another error term, random walk vector, is drawn from a zero mean Guassian distribution at every INS step and added to measurement values.

Error terms must be added to true measurements modeled by an IMU simulator in correct units. Statistics of IMU error terms provided by manufacturer in specification sheets are mostly not given in SI units [18]. Hence appropriate unit conversions must be applied to information provided by manufacturer according to units of measurements in simulation.

CHAPTER 4

FUSION OF IMAGING AND INERTIAL MEASUREMENTS

In VINS, INS is aided by visual measurements. In order to perform this, required navigation information from visual measurements must be extracted and fused with INS using a framework which is referred as sensor fusion filter in literature. A very simple example for information fusion can be using mean of the information obtained from different sources. A more elegant way can be weighting the measurements according to their uncertainties. Modern methods use uncertainties involved in the system and follow a probabilistic approach.

Recursive Bayesian Estimation, also called Bayes filter, is an information fusion method based on Bayes Rule given in Eq. 4.1. In Recursive Bayesian Estimation, distribution of the true state is tried to be estimated based on the information of the previous state, which constitutes "given" part of the equation 4.1, and current measurement. A mobile robot moving on a grid composed of landmarks can be as an example for Recursive Bayesian Estimation. When the robot starts its operation, navigation state errors and uncertainties increase until it sees first landmark. When the first landmark is detected, errors are corrected and uncertainties are decreased by using the information obtained from this landmark. If landmark is "distinctive" in terms of its position, then robot gets more accurate information which decreases its navigation information errors and uncertainties.

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{4.1}$$

A special case of Bayes Filter assuming that error sources of system have zero mean Gaussian distribution and system is linear is a well-known concept referred as Kalman filter. Kalman filter offers an optimal solution for the systems satisfying to its assumptions. This optimality involves guaranteed convergence of covariances and minimum possible state error by using available information. However, most of the real life systems are not linear. Therefore, different extensions to Kalman filter are proposed in literature like Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Guassian Mixture Filter (GMF).

Indirect filters attempt to estimate and correct navigation information errors by using the error states as explained in Chapter 3. Hence, perturbation analysis applied for deriving error dynamics. In the following section, observation model for modeling the camera measurements will be given and the perturbation analysis will be applied for using it in Indirect Bayesian filters.

4.1 Observation Model

In a control system, observation model, indicated by h_k in Eq. 3.2, establishes the mathematical relation between states and measurements. There are two commonly used observation models in VINS literature as explained in Chapter 2. In loosely coupled approach, rotation and translation are calculated from the camera measurements and used as output vector, indicated by z_k in Eq. 3.2. Second approach, referred as tightly coupled VINS, directly uses pixel values without explicitly calculating the navigation states from measurements. In this study tightly coupled approach is followed. Therefore mathematical relation between navigation states and pixel measurements must be derived.

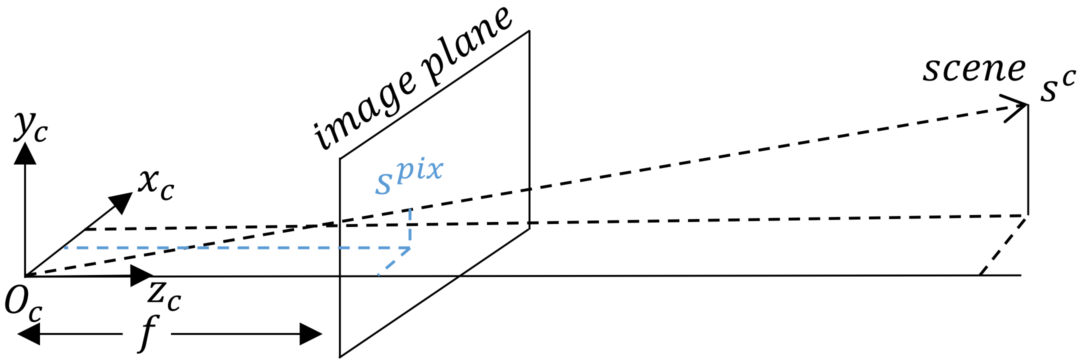


Figure 4.1: Perspective Projection Model

Basically, camera is a sensor projecting 3D environment to a 2D plane. Several different camera projection models are proposed in literature, e.g. weak perspective projection, orthographic projection, paraperspective projection and perspective projection. All projection models except perspective projection make assumptions about depth and depth variance of the imaged scene to decrease the computational complex-

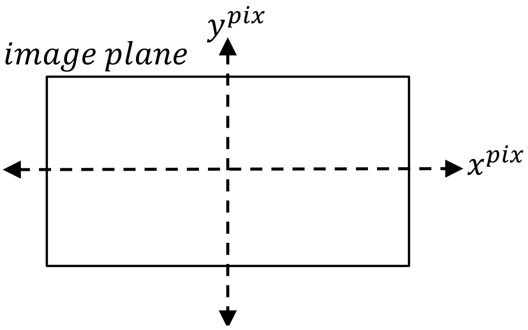


Figure 4.2: Image Plane

ity of specific applications. Hence, the accuracy of perspective projection is higher compared to the other projection models.

Fig. 4.1 shows the projection of a scene point onto image plane which forms pixel measurements. Equations defining perspective projection model which constitutes mathematical relation between position of imaged scene point and pixel measurement are given below.

$$\bar{\boldsymbol{s}}^{pix} = \frac{1}{\boldsymbol{s}^c_z}\boldsymbol{T}^{pix}_c \boldsymbol{s}^c \tag{4.2}$$

$$\boldsymbol{s}^c_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{s}^c \tag{4.3}$$

$$\boldsymbol{T}^{pix}_c = \begin{bmatrix} \frac{f}{s_x} & 0 & \frac{res_x}{2} \\ 0 & \frac{f}{s_y} & \frac{res_y}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

where $\boldsymbol{s}^c$ is the coordinates of a point in c-frame, $\boldsymbol{T}^{pix}_c$ is the intrinsic camera matrix defining camera properties, $\boldsymbol{s}^{pix}$ is unnormalized pixel coordinates of the projected point in image-frame, f is the focal length, s_x and s_y are pixel pitches, res_x and res_y are camera resolutions in x and y directions respectively and $\boldsymbol{s}^c_z$ is z coordinate of point in c-frame, in other words depth. Note that 12perspective projection is a nonlinear transformation since it contains element $\boldsymbol{s}^c_z$, from $\boldsymbol{s}_c$ operand vector.

In perspective projection, two different parameter sets are used: external and internal parameters. Intrinsic parameters are defined by $\boldsymbol{T}^{pix}_c$ which projects a point in c-frame to image plane using perspective projection equations. Extrinsic parameters define the rotation and translation of camera in 3D world. Using extrinsic parameters any point in n-frame is transferred to c-frame as follows:

$$\boldsymbol{s}^c = \boldsymbol{C}_b^c[\boldsymbol{C}_n^b(\boldsymbol{s}^n - \boldsymbol{p}^n) - \boldsymbol{T}_c^b] \tag{4.5}$$

where $\boldsymbol{p}^n$ is the position of the mobile platform in n-frame, $\boldsymbol{s}^n$ is the coordinates of a scene point in n-frame, $\boldsymbol{T}_c^b$ is the translation from b-frame to c-frame, in other words position of c-frame in b-frame, $\boldsymbol{C}_n^b$ is the DCM defining rotation from n-frame to b-frame and $\boldsymbol{C}_b^c$ is the DCM defining rotation from b-frame to c-frame. By using extrinsic and intrinsic parameters, nonlinear perspective projection equation can be finally defined as:

$$\bar{\boldsymbol{s}}^{pix} = h(\boldsymbol{s}^c, \boldsymbol{T}_c^{pix}) \tag{4.6}$$

Adding measurement noise $\boldsymbol{v}$ to $\bar{\boldsymbol{s}}^{pix}$ models real measurement as:

$$\boldsymbol{z} = \bar{\boldsymbol{s}}^{pix} + \boldsymbol{v} \tag{4.7}$$

4.1.1 Perturbation Analysis on Perspective Camera Model

Indirect filters use error states to estimate and correct the system errors. Therefore, observation model relating error states and measurements must be derived by applying perturbation analysis to perspective projection model.

Perturbation equation of the perspective projection model can be written based on the equations defined in Section 3.3 as:

$$\delta\boldsymbol{z}_k = \boldsymbol{H}_k\delta\boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.8}$$

$$\boldsymbol{H}_k = \frac{\partial h}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}_k} = \left.\begin{bmatrix} \frac{\partial h_1}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial h_1}{\partial \boldsymbol{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial h_n}{\partial \boldsymbol{x}_n} \end{bmatrix}\right|_{\bar{\boldsymbol{x}}_k} \tag{4.9}$$

where $\boldsymbol{H}_k$ is the linearized perspective projection matrix and $\boldsymbol{v}_k$ is the measurement noise. In order to derive $\boldsymbol{H}_k$, Eq. 4.6 must be linearized about states of VINS.

In robotics, Simultaneous Localization and Mapping (SLAM) is a widely used algorithm for "simultaneously" calculating the robot's navigation information and mapping its surrounding. The underlying idea of SLAM algorithm is incorporating the position of landmarks into the state vector. Thus, SLAM algorithm can estimate both navigation information and position of landmarks, in other words map, simultaneously. Both mapless and map-based VINS use SLAM approach. Hence, perspective projection equation (4.6) will be linearized about both in navigation states and position of landmarks.

Linearized perspective projection matrix in VINS is composed of 4 Jacobian matrices which are derivative of nonlinear perspective projection equation with respect to attitude error, velocity, position and landmark positions respectively.

$$\boldsymbol{H}_k = \begin{bmatrix} \frac{\partial h}{\partial \boldsymbol{\psi}} & \frac{\partial h}{\partial \boldsymbol{v}} & \frac{\partial h}{\partial \boldsymbol{p}^{llh}} & \frac{\partial h}{\partial \boldsymbol{y}^n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{\psi}} & \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{v}^n} & \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{p}^{llh}} & \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{y}^n} \end{bmatrix} \tag{4.10}$$

where $\boldsymbol{y}^n$ is position of landmark in n-frame. The definitions of landmark and platform position in n-frame are consistent and origin is fixed to a point in geodesic coordinate system.

Derivative of the nonlinear perspective projection equation with respect to attitude error is written by using chain and Quotient rules of derivative as follows:

$$\boldsymbol{H}_{\boldsymbol{z}\boldsymbol{\psi}} = \frac{\partial h}{\partial \boldsymbol{\psi}} = \frac{\boldsymbol{s}^c_z \frac{\partial \boldsymbol{s}^c}{\partial \boldsymbol{\psi}} - \boldsymbol{s}^c \left[\frac{\partial \boldsymbol{s}^c}{\partial \boldsymbol{\psi}}\right]_z}{(\boldsymbol{s}^z_c)^2} \tag{4.11}$$

Derivative of the imaged scene point with respect to attitude error, $\frac{\partial \boldsymbol{s}^c}{\partial \boldsymbol{\psi}}$, will be established based on assumption that derivative of state and its error state are approximately equal. This assumption comes from concepts explained in Section 3.3 and expressed as below:

$$\frac{\partial \delta \boldsymbol{s}^c}{\partial \boldsymbol{\psi}} \approx \frac{\partial \boldsymbol{s}^c}{\partial \boldsymbol{\psi}} \tag{4.12}$$

Imaged scene point position error $\delta \boldsymbol{s}^c$ is written as:

$$\delta \boldsymbol{s}^c = \tilde{\boldsymbol{s}}^c - \boldsymbol{s}^c \tag{4.13}$$

where $\tilde{\boldsymbol{s}}^c$ is the estimated position of the imaged scene point in c-frame and expressed as:

$$\widetilde{\boldsymbol{s}}^c = \boldsymbol{C}^c_b \widetilde{\boldsymbol{C}}^b_n [\boldsymbol{y}^n - \boldsymbol{p}^n] \tag{4.14}$$

$\widetilde{\boldsymbol{C}}^b_n$ is the estimated DCM. Under the assumption of small angles of misalignment [51], $\widetilde{\boldsymbol{C}}^b_n$ is defined as:

$$\widetilde{\boldsymbol{C}}^n_b = [\boldsymbol{I}_{3\times3} - [\times\boldsymbol{\psi}]]\boldsymbol{C}^n_b \tag{4.15}$$

where $\boldsymbol{I}_{3\times3}$ is a 3×3 identity matrix and $[\times\boldsymbol{\psi}]$ is the skew-symmetric form of attitude error vector. Transposing $\widetilde{\boldsymbol{C}}_b^n$ and applying definition of skew-symmetric matrices gives:

$$\widetilde{\boldsymbol{C}}_n^b = \boldsymbol{C}_n^b[\boldsymbol{I}_{3\times3} + [\times\boldsymbol{\psi}]] \tag{4.16}$$

Substituting Eq.4.16 into Eq. 4.14 gives:

$$\widetilde{\boldsymbol{s}}_c = \boldsymbol{C}_b^c\boldsymbol{C}_n^b[\boldsymbol{I}_{3\times3} + [\times\boldsymbol{\psi}]][\boldsymbol{y}^n - \boldsymbol{p}^n] \tag{4.17}$$

Position of the 4imaged scene point in c-frame can be redefined by assuming that translation between camera and body is, $\boldsymbol{T}_c^b$, is zero.

$$\boldsymbol{s}^c = \boldsymbol{C}_b^c\boldsymbol{C}_n^b[\boldsymbol{y}^n - \boldsymbol{p}^n] \tag{4.18}$$

This concludes definitions of $\boldsymbol{s}^c$ and $\widetilde{\boldsymbol{s}}_c$. Using these definitions, Eq. 4.13 can be explicitly written as:

$$\delta\boldsymbol{s}^c = \boldsymbol{C}_b^c\boldsymbol{C}_n^b[\times\boldsymbol{\psi}][\boldsymbol{y}^n - \boldsymbol{p}^n] \tag{4.19}$$

Assuming that 4.12 holds and using rule of cross-product in Eq. 3.53, final form of $\frac{\partial\boldsymbol{s}^c}{\partial\boldsymbol{\psi}}$ can be written as:

$$\begin{aligned}\frac{\partial\boldsymbol{s}^c}{\partial\boldsymbol{\psi}} &= \frac{\partial}{\partial\boldsymbol{\psi}}[-\boldsymbol{C}_b^c\boldsymbol{C}_n^b[\times(\boldsymbol{y}^n - \boldsymbol{p}^n)]\boldsymbol{\psi}] \\ &= -\boldsymbol{C}_b^c\boldsymbol{C}_n^b[\times(\boldsymbol{y}^n - \boldsymbol{p}^n)]\end{aligned} \tag{4.20}$$

Velocity of the platform can cause unwanted affects like "motion blur" on image. But perspective projection equations are established in discrete time and does not take image degradations into account. Hence jacobian matrix $\boldsymbol{H}_{\boldsymbol{zv}}$ representing derivative of nonlinear perspective projection equations w.r.t. velocity of the platform, is a zero matrix.

$$\boldsymbol{H}_{\boldsymbol{zv}} = \frac{\partial h}{\partial\boldsymbol{v}} = \boldsymbol{0}_{3\times3} \tag{4.21}$$

Derivative of nonlinear perspective projection equations w.r.t. position of the platform is written by using chain rule, since Eq. 4.5 is written in n-frame but position information of platform is maintained at geodesic coordinate system.

$$\boldsymbol{H}_{\boldsymbol{zp}} = \frac{\partial h}{\partial\boldsymbol{p}^{llh}} = \frac{\partial h}{\partial\boldsymbol{p}^n}\frac{\boldsymbol{p}^n}{\boldsymbol{p}^{llh}} = \frac{\boldsymbol{s}_z^c\frac{\partial\boldsymbol{s}^c}{\partial\boldsymbol{p}^n} - \boldsymbol{s}^c\left[\frac{\partial\boldsymbol{s}^c}{\partial\boldsymbol{p}^n}\right]_z}{(\boldsymbol{s}_c^z)^2}\frac{\boldsymbol{p}^n}{\boldsymbol{p}^{llh}} \tag{4.22}$$

Rightmost term of 4.22 can be calculated using Eqs. 3.30, 3.31 and 3.32 as:

$$\frac{\boldsymbol{p}^{n}}{\boldsymbol{p}^{llh}} = \begin{bmatrix} R_N + h & 0 & 0 \\ 0 & (R_E + h)cos(L) & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.23}$$

Taking derivative of Eq.4.5 w.r.t. position of platform in n-frame yields:

$$\frac{\partial \boldsymbol{s}^{c}}{\partial \boldsymbol{p}^{n}} = -\boldsymbol{C}_{b}^{c}\boldsymbol{C}_{n}^{b} \tag{4.24}$$

Since position of landmarks in n-frame is also incorporated into state vector in SLAM, observation model must also be linearized about these states. Since landmarks stands as imaged scene points the term $\boldsymbol{s}^{n}$ becomes $\boldsymbol{y}^{n}$ and $\boldsymbol{H}_{\boldsymbol{zy}}$ is expressed as:

$$\boldsymbol{H}_{\boldsymbol{zy}} = \frac{\partial h}{\partial \boldsymbol{y}^{n}} = \frac{\boldsymbol{s}_{z}^{c}\frac{\partial \boldsymbol{s}^{c}}{\partial \boldsymbol{y}^{n}} - \boldsymbol{s}^{c}\left[\frac{\partial \boldsymbol{s}^{c}}{\partial \boldsymbol{y}^{n}}\right]_{z}}{(\boldsymbol{s}_{c}^{z})^{2}} \tag{4.25}$$

Taking derivative of Eq.4.5 w.r.t. position of landmark in n-frame yields:

$$\frac{\partial \boldsymbol{s}^{c}}{\partial \boldsymbol{y}^{n}} = \boldsymbol{C}_{b}^{c}\boldsymbol{C}_{n}^{b} \tag{4.26}$$

This concludes the pertubation analysis of the perpective projection model. This equations are crucial for Kalman Filter's update step which relates the state and covariance predictions with measurements.

4.1.2 Derivation of Landmark Position Covariance

Sensor fusion filters try to fuse the information by weighting the different sources. These weights are determined according to uncertainties of the sources. Therefore landmark position covariances play a vital role on determining the weight of the visual measurements in VINS.

Derivation of landmark position covariance differs for map-based VINS and mapless-VINS. In mapless-VINS, landmark position covariance is calculated when landmark is first extracted from the imaged scene. This covariance depends on pixel measurement, depth information, most importantly the navigation solution of the mobile platform and its uncertainty. A landmark position covariance can be defined as follows:

$$\boldsymbol{P}_{\boldsymbol{yy}} = E[\delta\boldsymbol{y}\delta\boldsymbol{y}^{T}] \tag{4.27}$$

where $\delta\boldsymbol{y}$ represents the landmark position error. Number of landmarks tracked in VINS must be more than 4 in order to have an observable system. Therefore multiple landmarks are used in this study. These landmarks will be managed with the index m.

In order to derive landmark position error for calculating landmark position covariance, perturbation analysis must be applied to position of landmark. Position of landmark m in n-frame is calculated as follows:

$$\boldsymbol{y}_m^n = \boldsymbol{p}^n + \boldsymbol{C}_b^n(\boldsymbol{T}_c^b + d_m\boldsymbol{C}_c^b\boldsymbol{T}_{pix}^c\boldsymbol{z}_m) \tag{4.28}$$

where d_m represents the distance between the mobile platform and landmark m. The nonlinear relation giving the position of landmark in n-frame will be represented with the function g:

$$\boldsymbol{y}_m^n = g(\boldsymbol{p}^n, \boldsymbol{C}_b^n, \boldsymbol{T}_c^b, d_m, \boldsymbol{C}_c^b, \boldsymbol{T}_{pix}^c, \boldsymbol{z}_m) \tag{4.29}$$

Applying perturbation analysis explained in Section 3.3 to Eq. 4.28 gives:

$$\delta\boldsymbol{y}_m^n = \boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{z}}\delta\boldsymbol{z} + \boldsymbol{G}_{\boldsymbol{y}_m^n d_m}\delta d \tag{4.30}$$

where $\delta\boldsymbol{z}$ is measurement noise, $\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{x}}$, $\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{z}}$, $\boldsymbol{G}_{\boldsymbol{y}_m^n d_m}$ are derivatives of position of landmark m in n-frame w.r.t. navigation solution, pixel measurement and the distance between the mobile platform and landmark m respectively.

The derivative of position of landmark m in n-frame w.r.t. navigation solution is composed of three subparts: derivatives of position of landmark w.r.t. attitude error, velocity and position expressed in geodesic coordinate system respectively.

$$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{x}} = [\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{\psi}}\ \boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{v}}\ \boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{p}^{llh}}] \tag{4.31}$$

Derivative of the position of landmark m w.r.t. attitude error can be written as follows:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{\psi}} = \frac{\partial\boldsymbol{y}_m^n}{\partial\boldsymbol{\psi}} = \boldsymbol{C}_b^n(\times[\boldsymbol{T}_c^b + d_m\boldsymbol{C}_c^b\boldsymbol{T}_{pix}^c\boldsymbol{z}_m]) \tag{4.32}$$

Since the position of landmark m is independent from the velocity of the mobile platform. Derivate is a zero matrix:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n\boldsymbol{v}} = \boldsymbol{0}_{3\times3} \tag{4.33}$$

Eq. 4.28 gives the position of landmark m in frame. In order to calculate the effect of position of platform in geodesic coordinate system, chain rule must be used:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{p}^{llh}} = \frac{\partial \boldsymbol{y}_m^n}{\partial \boldsymbol{p}^n} \frac{\partial \boldsymbol{p}^n}{\partial \boldsymbol{p}^{llh}} \tag{4.34}$$

Differentiating Eq. 4.28 w.r.t. $\boldsymbol{p}^n$ gives identity matrix. This shows that position error in n-frame transformed to landmark position error as a shift.

$$\frac{\partial \boldsymbol{y}_m^n}{\partial \boldsymbol{p}^n} = \boldsymbol{I}_{3\times3} \tag{4.35}$$

Substituting Eq. 4.35 to Eq. 4.34 gives:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{p}^{llh}} = \frac{\partial \boldsymbol{p}^n}{\partial \boldsymbol{p}^{llh}} \tag{4.36}$$

Equation transforming geodesic coordinate system to n-frame were defined in Eq. 4.24. This results:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{p}^{llh}} = \begin{bmatrix} R_N + h & 0 & 0 \\ 0 & (R_E + h)cos(L) & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.37}$$

Differentiating Eq. 4.28 w.r.t. pixel measurement of landmark m gives:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{z}_m} = \frac{\partial \boldsymbol{y}_m^n}{\partial \boldsymbol{z}_m} \tag{4.38}$$

Eq. 4.38 can be written explicitly as follows:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{z}_m} = d_m \boldsymbol{C}_b^n \boldsymbol{C}_c^b \boldsymbol{T}_{pix}^c \begin{bmatrix} \boldsymbol{I}_{2\times2} \\ \boldsymbol{0}_{1\times2} \end{bmatrix} \tag{4.39}$$

Differentiating Eq. 4.28 w.r.t. distance between the mobile platform and landmark m can be shown as:

$$\boldsymbol{G}_{\boldsymbol{y}_m^n d_m} = \frac{\partial \boldsymbol{y}_m^n}{\partial d_m} \tag{4.40}$$

$$\boldsymbol{G}_{\boldsymbol{y}_m^n d_m} = \boldsymbol{C}_b^n \boldsymbol{C}_c^b \boldsymbol{T}_{pix}^c \boldsymbol{z}_m \tag{4.41}$$

This concludes the derivation influence matrices $\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{x}}$, $\boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{z}}$, $\boldsymbol{G}_{\boldsymbol{y}_m^n d_m}$. The covariance matrix including all landmarks uncertainties can be calculated using Eq. 4.27 and Eq. 4.30 as follows:

$$\boldsymbol{P}_{\boldsymbol{y}\boldsymbol{y}} = \boldsymbol{G}_{\boldsymbol{y}\boldsymbol{x}} \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{G}_{\boldsymbol{y}\boldsymbol{x}}^T + \boldsymbol{G}_{\boldsymbol{y}\boldsymbol{z}} \boldsymbol{R} \boldsymbol{G}_{\boldsymbol{y}\boldsymbol{z}}^T + \boldsymbol{G}_{\boldsymbol{y}d} \boldsymbol{P}_{dd} \boldsymbol{G}_{\boldsymbol{y}d}^T \tag{4.42}$$

where:

$$\boldsymbol{G}_{\boldsymbol{y}\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{G}_{\boldsymbol{y}_1^n \boldsymbol{x}} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{x}} \end{bmatrix} \tag{4.43}$$

$$\boldsymbol{G}_{\boldsymbol{y}\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{G}_{\boldsymbol{y}_1^n \boldsymbol{z}_1} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{y}_m^n \boldsymbol{z}_m} \end{bmatrix} \tag{4.44}$$

$$\boldsymbol{G}_{\boldsymbol{y}d} = \begin{bmatrix} \boldsymbol{G}_{\boldsymbol{y}_1^n d_1} \\ \vdots \\ \boldsymbol{G}_{\boldsymbol{y}_m^n d_m} \end{bmatrix} \tag{4.45}$$

$$\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{\psi}\boldsymbol{\psi}} & \boldsymbol{P}_{\boldsymbol{\psi}\boldsymbol{\psi}} & \boldsymbol{P}_{\boldsymbol{\psi}\boldsymbol{\psi}} \\ \boldsymbol{P}_{\boldsymbol{v}\boldsymbol{\psi}} & \boldsymbol{P}_{\boldsymbol{v}\boldsymbol{v}} & \boldsymbol{P}_{\boldsymbol{v}\boldsymbol{p}} \\ \boldsymbol{P}_{\boldsymbol{p}\boldsymbol{\psi}} & \boldsymbol{P}_{\boldsymbol{p}\boldsymbol{v}} & \boldsymbol{P}_{\boldsymbol{p}\boldsymbol{p}} \end{bmatrix} \tag{4.46}$$

Distance uncertainty of LiDAR can be represented by a scalar. Hence $\boldsymbol{P}_{dd}$ can be written in terms of standard deviation of distance measurement σ_d as:

$$\boldsymbol{P}_{dd} = \sigma_d^2 \tag{4.47}$$

Cross-correlation terms between landmark positions and navigation variables are calculated by taking expected value of error terms:

$$\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{y}} = E[\delta\boldsymbol{x}\delta\boldsymbol{y}^T] \tag{4.48}$$

$$\boldsymbol{P}_{\boldsymbol{y}\boldsymbol{x}} = E[\delta\boldsymbol{y}\delta\boldsymbol{x}^T] \tag{4.49}$$

Expected values of error term are zero $E[\delta\boldsymbol{x}] = \boldsymbol{0}$, $E[\delta\boldsymbol{z}] = \boldsymbol{0}$. Substituting Eq. 4.30 into Eqs. 4.50 and 4.51, cross-covariance terms can be calculated as:

$$\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{y}} = \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{G}_{\boldsymbol{y}\boldsymbol{x}}^T \tag{4.50}$$

$$\boldsymbol{P}_{\boldsymbol{y}\boldsymbol{x}} = \boldsymbol{G}_{\boldsymbol{y}\boldsymbol{x}}\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} \tag{4.51}$$

Landmark position covariance and cross correlation matrices are calculated when a new landmark is initialized to a filter. After initializing landmarks, covariance matrix in Eq. 4.46 becomes:

$$\boldsymbol{P_{xx}} = \begin{bmatrix} \boldsymbol{P_{xx}} & \boldsymbol{P_{xy}} \\ \boldsymbol{P_{yx}} & \boldsymbol{P_{yy}} \end{bmatrix} \tag{4.52}$$

$$\boldsymbol{P_{xx}} = \begin{bmatrix} \boldsymbol{P_{\psi\psi}} & \boldsymbol{P_{\psi\psi}} & \boldsymbol{P_{\psi\psi}} & \boldsymbol{P_{\psi y}} \\ \boldsymbol{P_{v\psi}} & \boldsymbol{P_{vv}} & \boldsymbol{P_{vp}} & \boldsymbol{P_{vy}} \\ \boldsymbol{P_{p\psi}} & \boldsymbol{P_{pv}} & \boldsymbol{P_{pp}} & \boldsymbol{P_{py}} \\ \boldsymbol{P_{y\psi}} & \boldsymbol{P_{yv}} & \boldsymbol{P_{yp}} & \boldsymbol{P_{yy}} \end{bmatrix} \tag{4.53}$$

Positions of landmarks in map-based VINS are loaded to the mobile platform before flight and determined using a map or an external device like an handheld GPS. Because of the position uncertainties of landmarks originated from the method used in preplannig phase of map-based VINS, positions of landmarks must be incorporate into sensor fusion filter in order to refine the positions of landmarks using SLAM approach.

Unlike mapless-VINS, errors on positions of landmarks are independent from navigation solution in map-based VINS. Therefore cross-covariance terms becomes:

$$\boldsymbol{P_{xy}} = \mathbf{0}_{3\times3} \tag{4.54}$$

$$\boldsymbol{P_{yx}} = \mathbf{0}_{3\times3} \tag{4.55}$$

If position of the visual landmarks are determined with a method having independent errors in all axes such as handheld GPS, off-diagonal terms in matrix representing position uncertainties becomes zero.

$$P_{\boldsymbol{yy}} = \begin{bmatrix} P_{\boldsymbol{y}_1^n \boldsymbol{y}_1^n} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P_{\boldsymbol{y}_m^n \boldsymbol{y}_m^n} \end{bmatrix} \tag{4.56}$$

where individual landmark position error variance matrices are:

$$P_{\boldsymbol{y}_m^n \boldsymbol{y}_m^n} = \begin{bmatrix} {\sigma_m^N}^2 & 0 & 0 \\ 0 & {\sigma_m^E}^2 & 0 \\ 0 & 0 & {\sigma_m^D}^2 \end{bmatrix} \tag{4.57}$$

4.2 Extended Kalman Filter

Kalman Filter is an information fusion method using system control model, series of measurements and statistical information of error sources. It is based on Bayes rule and aims to estimate the states in an optimal manner under several assumptions. KF assumes that the system is linear and error sources behave according to zero mean Gaussian distribution. Since linearity assumption is not valid for most real life systems, an extended version of this filter, namely Extended Kalman filter(EKF) is proposed in literature. It is based on linearizing the nonlinear system about current state and applying the KF equations. Errors are still assumed to have zero mean Gaussian distribution and independent from each other.

Extended Kalman Filter is composed of *prediction* and *innovation* steps. Prediction step propagates the state and its covariance using state transition dynamics and inertial sensor's error distribution. Innovation step have two main parts. Firstly, expected measurements are calculated using current state, previous measurements and observation model. After estimating current measurements, using difference the between estimated measurement and real measurement, correction is calculated and applied on current state. Standard EKF equations are given below.
Prediction step:
State prediction:

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k) \tag{4.58}$$

State covariance prediction:

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^T + \boldsymbol{Q}_k \tag{4.59}$$

Innovation step:
Measurement estimate:

$$\hat{\boldsymbol{z}}_{k|k-1} = h(\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.60}$$

Innovation (or measurement) residual:

$$\hat{\boldsymbol{y}}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1} \tag{4.61}$$

Innovation (or residual) covariance:

$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k \tag{4.62}$$

Kalman gain:

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1} \tag{4.63}$$

Updated state prediction:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_k\hat{\boldsymbol{y}}_k \tag{4.64}$$

Updated estimation prediction:

$$\boldsymbol{P}_{k|k} = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\boldsymbol{P}_{k|k-1} \tag{4.65}$$

In this study, an alternative form of EKF namely Indirect (Error State) Extended Kalman Filter (I-EKF) is used. I-EKF aims to estimate and correct the errors on navigation solution using error states. In order to establish an I-EKF framework, error state dynamics were derived by applying perturbation techniques on INS equations in Chapter 3 and observation model relating error states and measurements was derived by applying perturbation analysis to perspective projection model in Sec. 4.1.1.

State-space representation of error states is written in the following form:

$$\delta\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\delta\boldsymbol{x}_k + \boldsymbol{G}_k\delta\boldsymbol{u}_k \tag{4.66}$$

$$\delta\boldsymbol{z}_k = \boldsymbol{H}_k\delta\boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.67}$$

Error sources are assumed to have zero mean Gaussian distribution:

$$\delta\boldsymbol{u}_k \sim N(\boldsymbol{0}, \boldsymbol{Q}_k) \;\; \boldsymbol{v}_k \sim N(\boldsymbol{0}, \boldsymbol{R}) \tag{4.68}$$

Prediction step of I-EKF slightly differs from standard EKF. In I-EKF, initial error estimate, $\delta\boldsymbol{x}_{0|0}$, is a zero vector and the value of input noise, $\delta\boldsymbol{u}[k]$, assumed to have zero mean. Hence, error state prediction is not performed. Instead, INS equations are propagated to obtain the current navigation information.

$\boldsymbol{P}_{\boldsymbol{xx}}$ is propagated in the same way as in standard EKF using discretized error state transition matrix and discretized input noise covariance matrix $\boldsymbol{Q}_k$. Initial state covariance matrix $\boldsymbol{P}_{0|0}$ is composed of initial variances representing uncertainties of attitude, velocity and position.

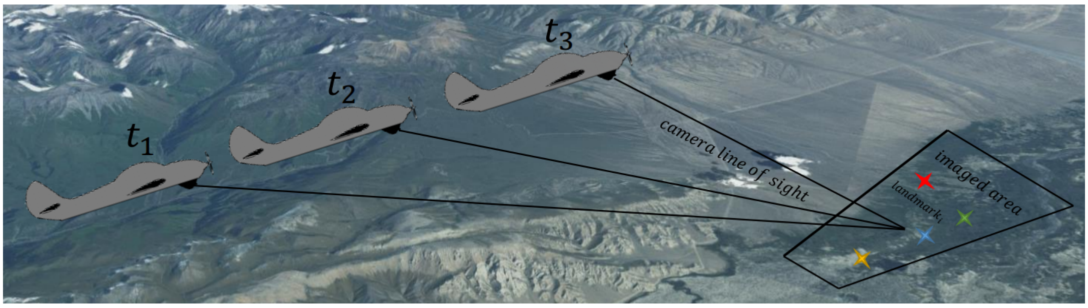


Figure 4.3: VINS Operation

Expected value of measurement error is always zero since error state mean is assumed to be zero. Hence innovation definition becomes:

$$\hat{\boldsymbol{y}}_k = \delta \boldsymbol{z}_k \tag{4.69}$$

But since $\delta \boldsymbol{z}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1}$, innovation is calculated in the same way as in standard EKF.

Since $\delta \hat{\boldsymbol{x}}_{k-1|k-1} = \boldsymbol{0}$, updated value of error estimate $\delta \hat{\boldsymbol{x}}_{k|k}$ is calculated as follows:

$$\delta \hat{\boldsymbol{x}}_{k|k} = \boldsymbol{K}_k \hat{\boldsymbol{y}}_k \tag{4.70}$$

After calculating the updated value of error estimate, navigation solution is updated accordingly. Hence error estimate, $\delta \hat{\boldsymbol{x}}_{k|k}$, again remains zero vector until the next innovation step. Fig. 4.3 shows the general operation of VINS. A UAV having a gimbaled camera is demonstrated at three different time measurement steps t_1, t_2 and t_3. Gimbaled camera is rotated downwards around pitch axis in order to observe same scene for the maximum period of time. At measurement step t_1, landmarks are extracted. If a landmark map is loaded on the platform, extracted landmark descriptors at t_1 are matched with the map. Expected pixel measurements are calculated from known positions of landmarks and estimated navigation solution of UAV. Positions of landmarks are incorporated into state vector. Landmark position covariances are incorporated into state covariance matrix. Difference between the expectations and measurements are used for updating the error state estimate. Positions of landmarks are also refined.

In mapless-VINS, positions of the landmarks are calculated and incorporated into state vector at t_1 measurement step. Since there is no previous information about landmarks, expectation cannot be calculated at t_1 measurement step. Therefore error state cannot be updated. Positions of landmarks in n-frame are calculated and incorporated into state vector. Landmark position covariance matrices are calculated as explained in Section 4.1.2 and incorporated into state covariance matrix. UAV propagates only INS equations between measurement steps t_1 and t_2. At measurement step t_2, expected pixel positions of previously extracted landmarks are calculated. After this step data association is carried out by matching landmark descriptors extracted at t_1 step and t_2. Innovation representing the difference between expected pixel positions and real pixel positions obtained with data association is calculated. Value of innovation and Kalman gain is multiplied in order to obtain updated value of error state. Updated value of error state is used for updating positions of landmarks in n-frame and navigation solution.

Measurement step t_3 is repetition of t_2 for both mapless and map-based VINS. But error updates are more consistent compared to previous step since updates are calculated from a landmark set with refined positions.

VINS can get updates from same landmark sets until physical limitations of camera gimbal is reached. Number of updates that mapless VINS gets is one step less than map-Based VINS. This originates from the fact that first measurement step of the mapless VINS only initialize the position of landmarks and can't have an expectation of pixel coordinates of landmarks.

Main flowchart of SLAM based VINS using Indirect Extended Kalman Filter (I-EKF) can be found in Fig. 4.5. Input block is elaborated at Fig. 4.4. Detailed steps regarding the propagation block are given at Fig. 4.6 and update block is expanded on Fig. 4.7.

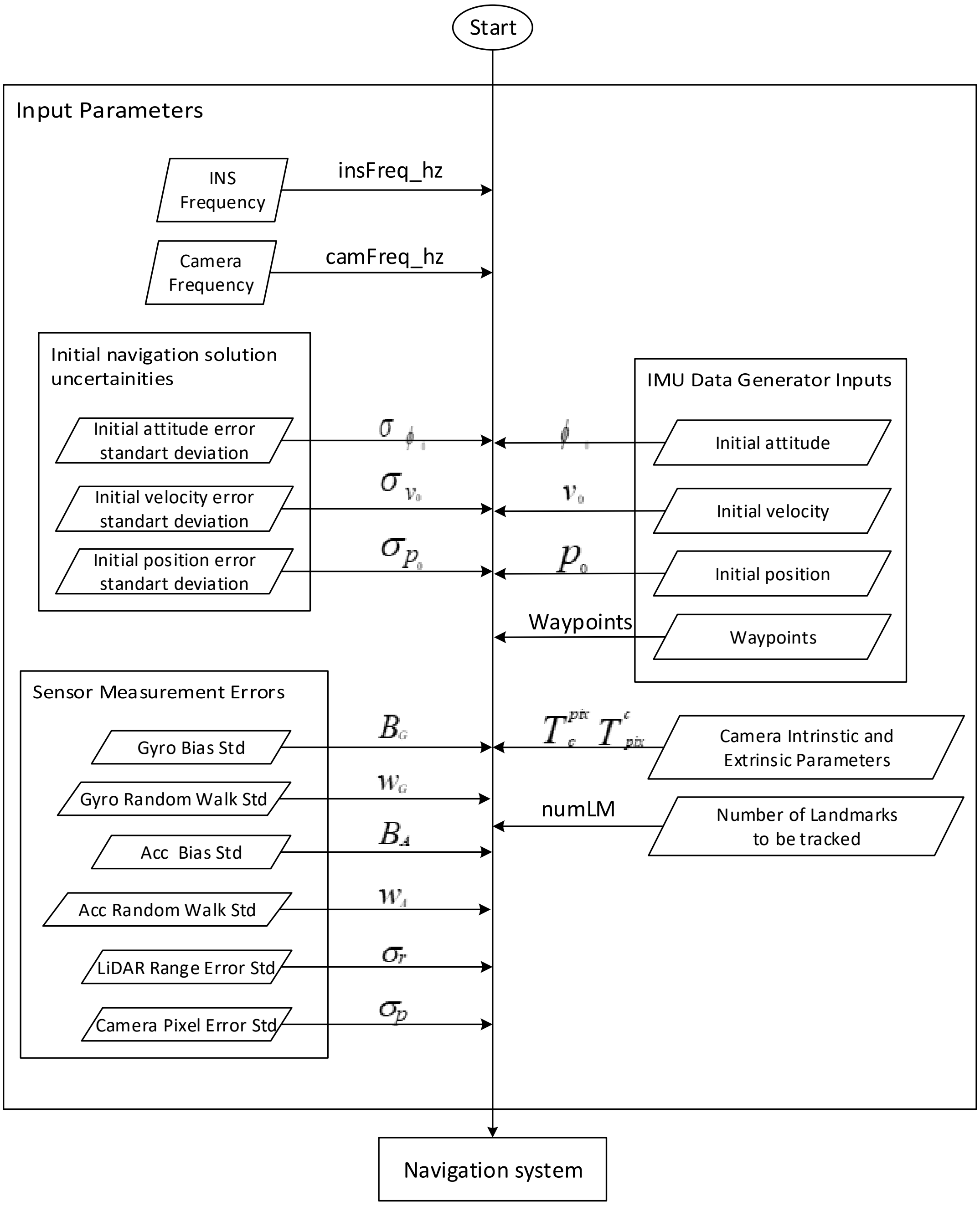


Figure 4.4: Input Parameters to Sensor Fusion

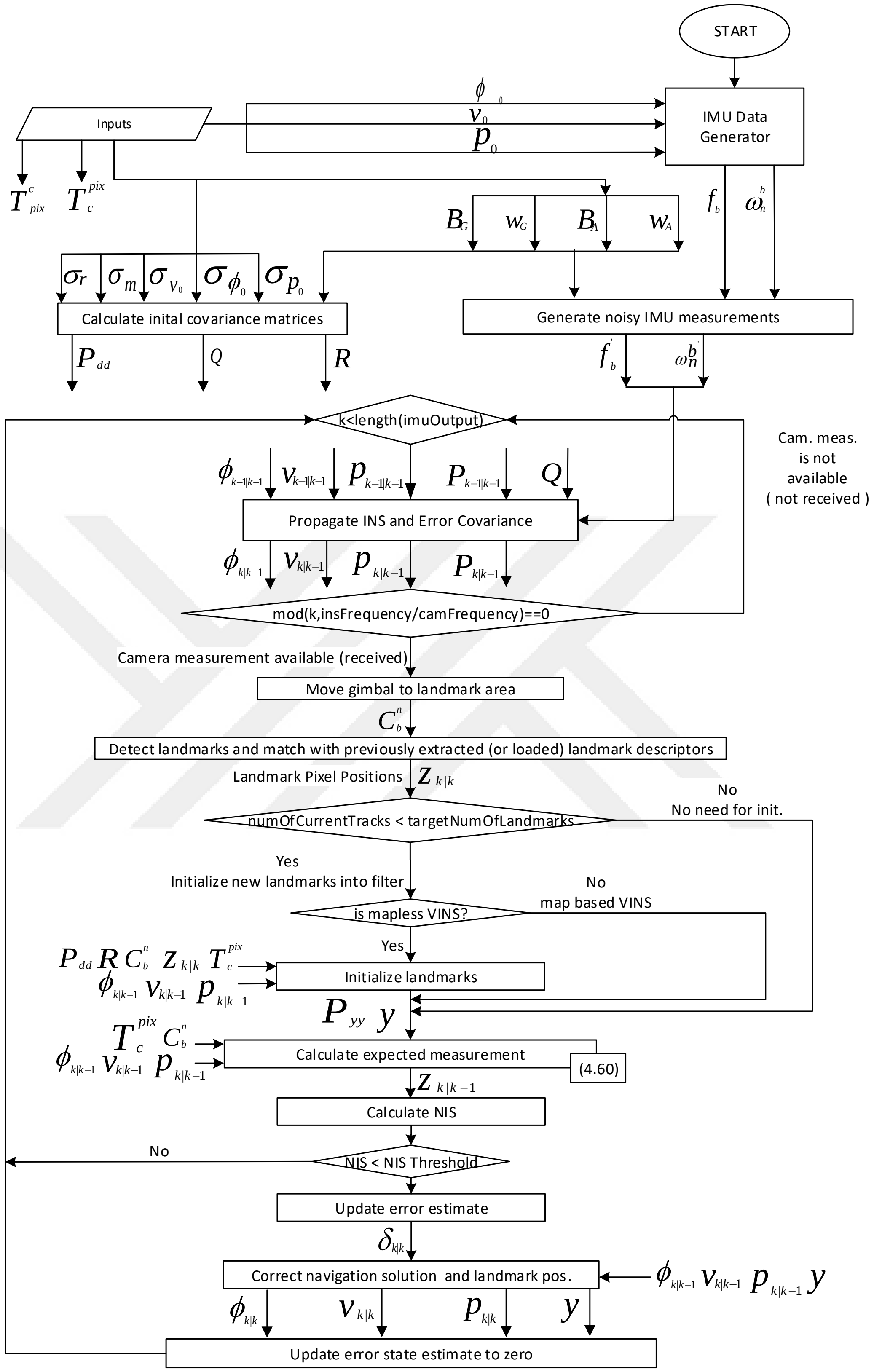

START
IMU Data Generator
Inputs
ϕ_0
v_0
p_0
T^c_{pix}
T^{pix}_c
B_G
w_G
B_A
w_A
f_b
ω^b_n
σ_r
σ_m
σ_{v_0}
σ_{ϕ_0}
σ_{p_0}
Calculate inital covariance matrices
Generate noisy IMU measurements
P_{dd}
Q
R
f'_b
$\omega^{b'}_n$
k<length(imuOutput)
Cam. meas. is not available (not received)
$\phi_{k-1|k-1}$
$v_{k-1|k-1}$
$p_{k-1|k-1}$
$P_{k-1|k-1}$
Q
Propagate INS and Error Covariance
$\phi_{k|k-1}$
$v_{k|k-1}$
$p_{k|k-1}$
$P_{k|k-1}$
mod(k,insFrequency/camFrequency)==0
Camera measurement available (received)
Move gimbal to landmark area
C^n_b
Detect landmarks and match with previously extracted (or loaded) landmark descriptors
Landmark Pixel Positions
$Z_{k|k}$
No
No need for init.
numOfCurrentTracks < targetNumOfLandmarks
Yes
Initialize new landmarks into filter
No
map based VINS
is mapless VINS?
Yes
P_{dd} R C^n_b $Z_{k|k}$ T^{pix}_c
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$
Initialize landmarks
P_{yy} y
T^{pix}_c C^n_b
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$
Calculate expected measurement
(4.60)
$Z_{k|k-1}$
Calculate NIS
No
NIS < NIS Threshold
Update error estimate
$\delta_{k|k}$
Correct navigation solution and landmark pos.
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$ y
$\phi_{k|k}$
$v_{k|k}$
$p_{k|k}$
y
Update error state estimate to zero

Figure 4.5: I-EKF Based VINS Main Flowchart

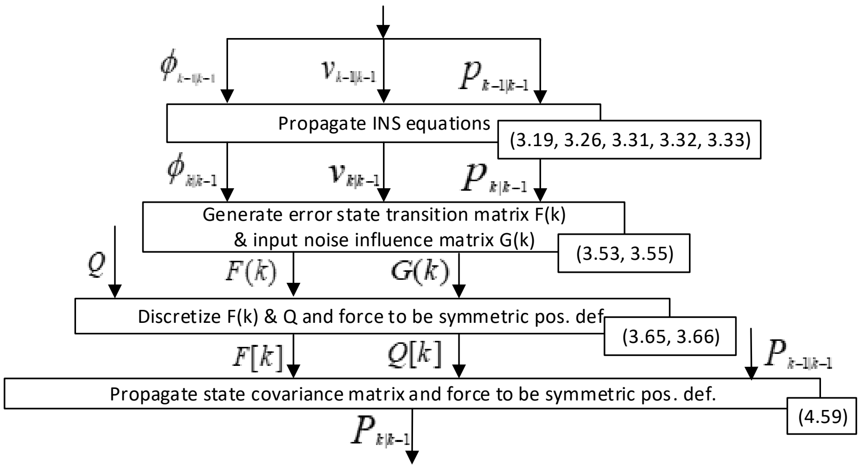


Figure 4.6: I-EKF Based VINS Prediction Flowchart

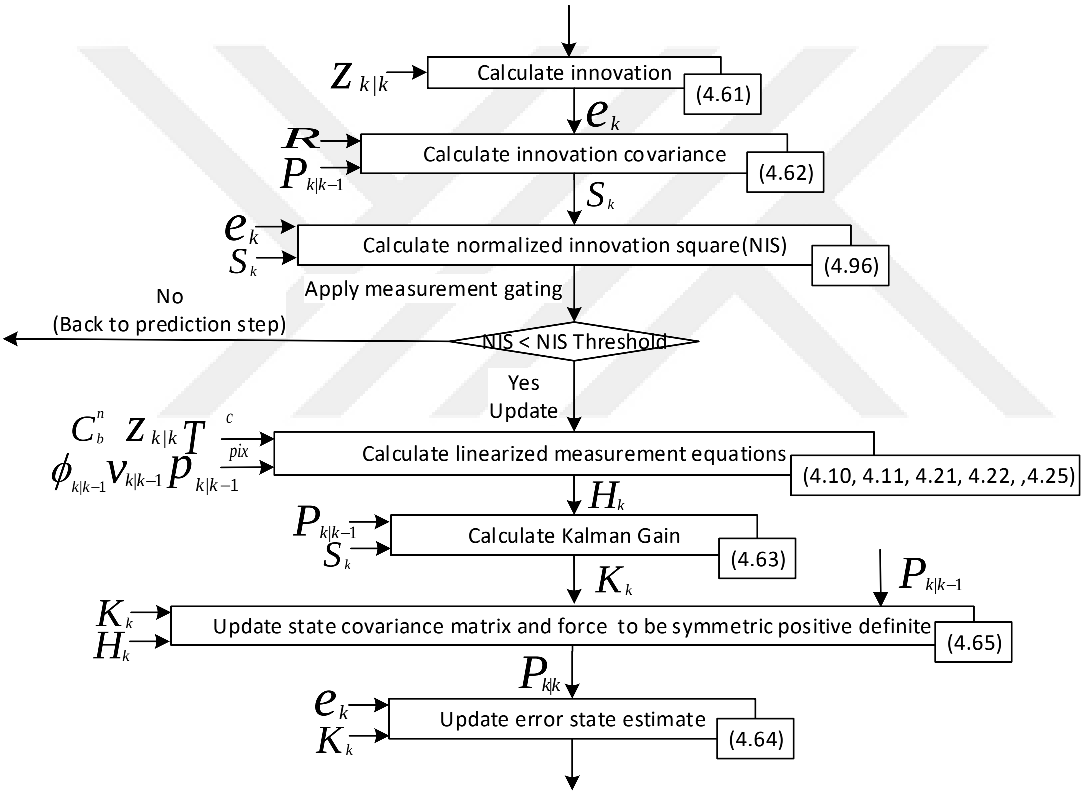

$Z_{k|k}$
Calculate innovation
(4.61)
e_k
R
$P_{k|k-1}$
Calculate innovation covariance
(4.62)
S_k
e_k
S_k
Calculate normalized innovation square(NIS)
(4.96)
Apply measurement gating
No
(Back to prediction step)
NIS < NIS Threshold
Yes
Update
C_b^n $Z_{k|k}$ T_{pix}^c
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$
Calculate linearized measurement equations
(4.10, 4.11, 4.21, 4.22, ,4.25)
H_k
$P_{k|k-1}$
S_k
Calculate Kalman Gain
(4.63)
K_k
$P_{k|k-1}$
K_k
H_k
Update state covariance matrix and force to be symmetric positive definite
(4.65)
$P_{k|k}$
e_k
K_k
Update error state estimate
(4.64)

Figure 4.7: I-EKF Based VINS Update Flowchart

4.3 Unscented Kalman Filter

EKF is a widely used sensor fusion technique for nonlinear systems. In EKF, state distribution is represented with a Gaussian random variable(GRV). Estimated mean of the distribution is propagated using linearized nonlinear system as shown in Fig. 4.8 Linearization of highly nonlinear system can cause sub-optimal performance or even divergence of state estimation error. Linearizing the system adds a computation cost to computation unit. Moreoever, linearization of highly nonlinear models increases the complexity of system. Therefore an alternative algorithm referred as Unscented Kalman Filter(UKF) proposed by Uhlmann et. al [28] offers higher accuracy and lower complexity.

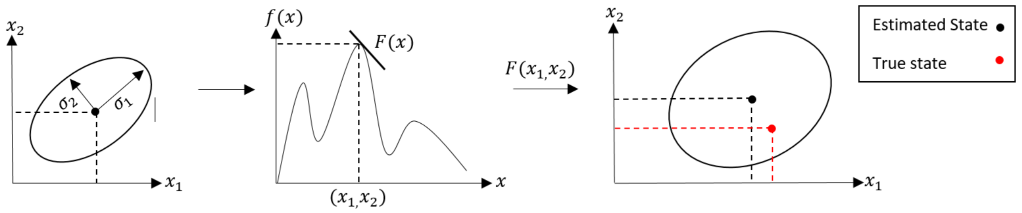


Figure 4.8: Propagation GRV with EKF

UKF is a derivative free sensor fusion technique. The name UKF refers to unscented transform(UT) which is a method for efficiently sampling the GRV. UT approximates the GRV with a deterministically calculated set of points referred as sigma points. Minimum set of sigma points contain $2L+1$ elements where L represents the number of dimensions of nonlinear system. Sigma points are propagated through nonlinear function as shown in Fig. 4.9.

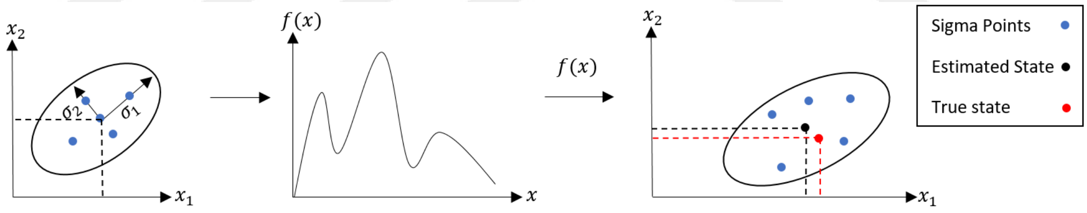


Figure 4.9: Propagation of Sigma Points through Nonlinear Function

$$\gamma_i = f(\boldsymbol{\chi}_i) \tag{4.71}$$

where χ_i and γ_i represent the i-th sigma point and the propagated i-th sigma point respectively. Sigma points also cover the mean and covariance of the distribution and calculated as follows:

$$\boldsymbol{\chi}_0 = \boldsymbol{\mu_x} \tag{4.72}$$

where χ_0 represents the central sigma point and equals to estimated mean of state distribution $\boldsymbol{\mu_x}$.

$$\boldsymbol{\chi}_i = \boldsymbol{x} + (\sqrt{(L+\lambda)\boldsymbol{P_{xx}}})_i \;\; i = 1, 2, ..., L \tag{4.73}$$

$$\boldsymbol{\chi}_i = \boldsymbol{x} + (\sqrt{(L+\lambda)\boldsymbol{P}_{\boldsymbol{xx}}})_{i-L}\ i = L+1, L+2, ..., 2L \tag{4.74}$$

where $\boldsymbol{\chi}_i$ are off-center sigma points, $(\sqrt{(L+\lambda)\boldsymbol{P}_{\boldsymbol{xx}}})_i$ is i-th column of offset matrix, λ is scaling parameter and calculated as shown below:

$$\lambda = \alpha^2(L+\kappa) - L \tag{4.75}$$

α determines the spread of sigma points around mean and usually set to a small value $1 \leq \alpha \leq 1e-04$ [53]. κ is secondary scaling parameter and usually set to $(3-L)$ or 0. Estimated state is calculated by weighted sum of propagated sigma points and state covariance is calculated by sum of weighted difference of propagated sigma points as follows:

$$\boldsymbol{\mu}_{\boldsymbol{x}} \approx \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\gamma}_i \tag{4.76}$$

$$\boldsymbol{P}_{\boldsymbol{xx}} \approx \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\gamma}_i - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{\gamma}_i - \boldsymbol{\mu}_{\boldsymbol{x}})^T + \boldsymbol{Q}_k \tag{4.77}$$

where $W_i^{(m)}$ is the weight of i-th sigma point for mean calculation and $W_i^{(c)}$ is the weight of i-th sigma point for covariance calculation. Weights for mean and covariance are calculated as follows:

$$W_0^{(m)} = \lambda/(L+\lambda) \tag{4.78}$$

$$W_0^{(c)} = \lambda/(L+\lambda) + (1-\alpha^2+\beta) \tag{4.79}$$

$$W_i^{(m)} = W_i^{(c)} = \lambda/(2\cdot(L+\lambda))\ i = 1, ..., 2L \tag{4.80}$$

where β is parameter used for incorporating prior knowledge of distribution of state. Optimal value of β equals to 2 for the systems having states with Gaussian distribution. Central component is weighted with greater values compared to off-center components since probability of central component is higher.

Innovation step of UKF fuses visual measurement and INS information without linearizing the observation model. Exptected measurements for all sigma points are calculated as follows.

$$\boldsymbol{\nu}_i = h(\boldsymbol{x}_i) \tag{4.81}$$

where $\boldsymbol{\nu}_i$ is expected measurement of i-th sigma point. Expected measurement is calculated by weighted sum of expected measurements from sigma points.

$$\hat{\boldsymbol{z}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\nu}_i \tag{4.82}$$

Innovation covariance is calculated as follows:

$$\boldsymbol{S}_k = \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\nu}_i - \hat{\boldsymbol{z}}_{k|k-1})(\boldsymbol{\nu}_i - \hat{\boldsymbol{z}}_{k|k-1})^T + \boldsymbol{R} \tag{4.83}$$

Cross-covariance between state and measurement is calculated as shown below:

$$\boldsymbol{P}_{\boldsymbol{x}\hat{\boldsymbol{z}}} = \sum_{i=0}^{2L} W_i^{(c)} (\boldsymbol{\chi}_i - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{\nu}_i - \hat{\boldsymbol{z}}_{k|k-1})^T \tag{4.84}$$

Kalman gain of UKF utilizes from cross covariance and innovation covariance and it is written as:

$$\boldsymbol{K}_k = \boldsymbol{P}_{\boldsymbol{x}\hat{\boldsymbol{z}}} \boldsymbol{S}_k{}^{-1} \tag{4.85}$$

Updated state is calculated as follows:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_k \hat{\boldsymbol{y}}_k \tag{4.86}$$

where $\hat{\boldsymbol{y}}_k$ is the innovation as explained before and calculated same as in EKF as:

$$\hat{\boldsymbol{y}}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_{k|k-1} \tag{4.87}$$

where $\boldsymbol{z}_k$ is the measurement at time step k.

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k-1|k-1} - \boldsymbol{K}_k \boldsymbol{S}_k{}^{-1} \boldsymbol{K}_k^T \tag{4.88}$$

While offering higher level of approximation and lower complexity, UKF also has disadvantages. Number of dimensions in SLAM based VINS is calculated as $(9 + 3 \times num\ of\ landmarks)$. Since minimum number of landmarks has to be used in order to have an observable VINS is equals to 4, minimum number of sigma points propagated on SLAM based VINS with UKF can be calculated as 43 which roughly corresponds to 43 times greater computational cost compared to EKF.

In [20] Gustafsson et. al. gives counterexample for the hypothesis of Uhlmann et. al in [28], which proposes: "Any set of sigma points that encodes the mean and covariance correctly calculates the projected mean and covariance correctly to the second order ". They show that UT does not give the correct covariances even for quadratic function $g(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{x}^T$ with GRV. This covariance error can cause suboptimal performance.

Modification on prediction and innovation steps for using error states which are explained in Section 4.2 can be easily adapted to UKF. Hence details will not be repeated. Main flowchart of SLAM based VINS using Indirect Unscented Kalman Filter(I-UKF) can be found in Fig. 4.11. Inputs block is elaborated at Fig. 4.4. Propagation block's detailed steps are given at Fig. 4.10 and update block is expanded on Fig. 4.12.

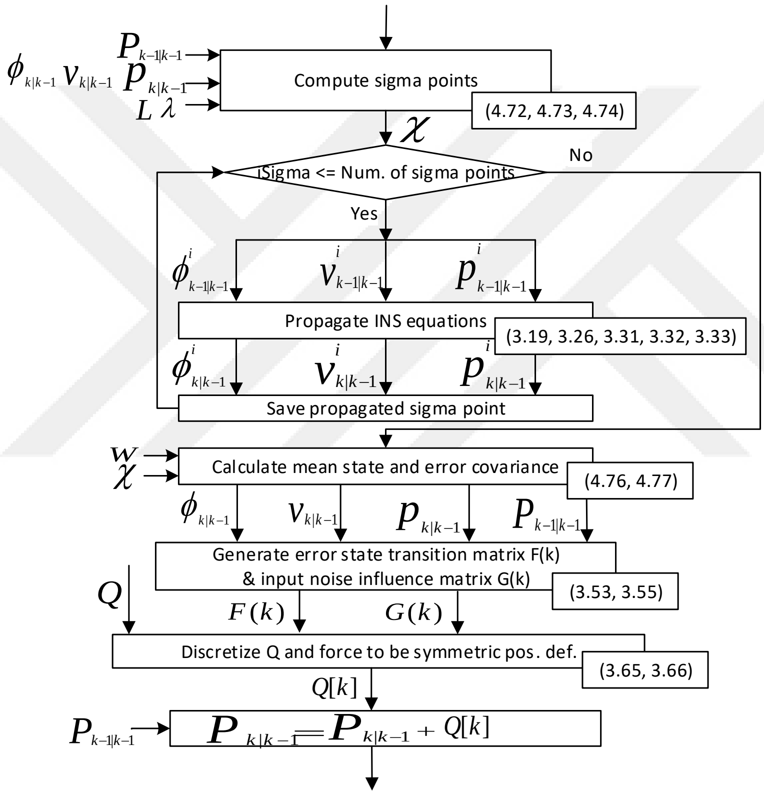


Figure 4.10: I-UKF Based VINS Prediction Flowchart

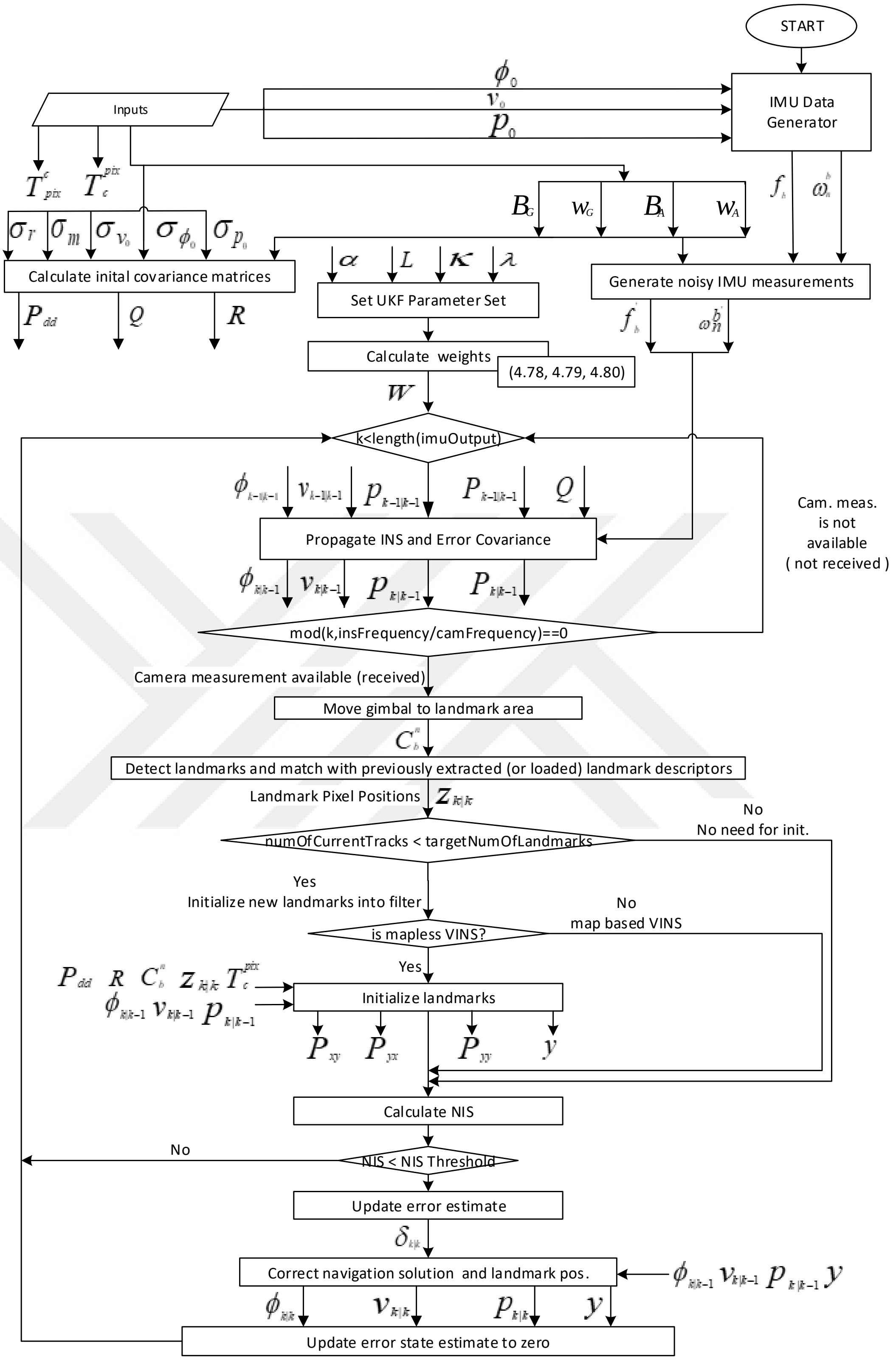

START
IMU Data Generator
Inputs
ϕ_0
v_0
p_0
T^c_{pix}
T^{pix}_c
σ_r σ_m σ_{v_0} σ_{ϕ_0} σ_{p_0}
B_G w_G B_A w_A
f_b ω^b_n
Calculate inital covariance matrices
P_{dd} Q R
α L κ λ
Set UKF Parameter Set
Calculate weights
(4.78, 4.79, 4.80)
W
Generate noisy IMU measurements
f'_b $\omega^{b'}_n$
k<length(imuOutput)
$\phi_{k-1|k-1}$ $v_{k-1|k-1}$ $p_{k-1|k-1}$ $P_{k-1|k-1}$ Q
Propagate INS and Error Covariance
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$ $P_{k|k-1}$
Cam. meas. is not available (not received)
mod(k,insFrequency/camFrequency)==0
Camera measurement available (received)
Move gimbal to landmark area
C^n_b
Detect landmarks and match with previously extracted (or loaded) landmark descriptors
Landmark Pixel Positions $z_{k|k}$
numOfCurrentTracks < targetNumOfLandmarks
No
No need for init.
Yes
Initialize new landmarks into filter
No
map based VINS
is mapless VINS?
Yes
P_{dd} R C^n_b $z_{k|k}$ T^{pix}_c
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$
Initialize landmarks
P_{xy} P_{yx} P_{yy} y
Calculate NIS
No
NIS < NIS Threshold
Update error estimate
$\delta_{k|k}$
Correct navigation solution and landmark pos.
$\phi_{k|k-1}$ $v_{k|k-1}$ $p_{k|k-1}$ y
$\phi_{k|k}$ $v_{k|k}$ $p_{k|k}$ y
Update error state estimate to zero

Figure 4.11: I-UKF Based VINS Main Flowchart

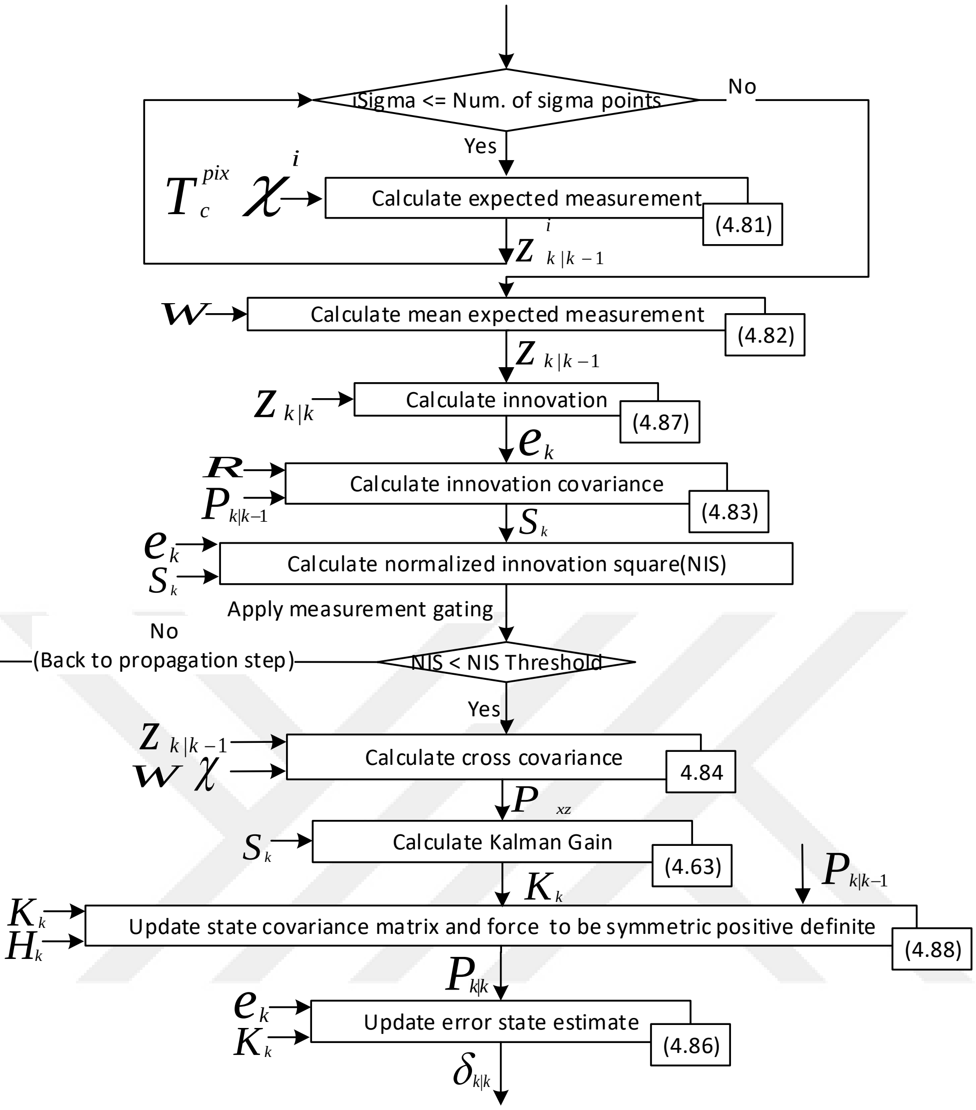


Figure 4.12: I-UKF Based VINS Update Flowchart

4.4 Gaussian Mixture Filter

Bayesian approaches which represent the state with a single Gaussian variable may end up with unsatisfactory results for highly nonlinear systems. An alternative approach namely Gaussian Mixture Filter (GMF) or Gaussian Sum Filter (GSF) in literature, offering a better representation of state density is proposed by Alspach et. al. in [1]. GMF approximates the posterior density of system state with the weighted sum of Gaussian components. Moreover it eliminates the assumption that posterior density of state is Gaussian.

Subproblems associated with the GMF can be summarized as follows:

- Initialization of Gaussian components and their weights,
- Choosing number of Gaussian components,
- Choosing weighting method of Gaussian components,
- Choosing weight update method.

Initial density of state must be sampled in a way that first samples efficiently represents the initial density and have enough Gaussian components for the state densities at proceeding steps. Unscented Transform (UT) offers an efficient approximation to Gaussian density if an appropriate parameter set is used. Weight can be assigned uniformly or UT weights can be used directly.

Number of Gaussian components must be choosen high enough to efficiently represent the state but computational complexity must be held low enough to implement and run the system in a real time with a computation unit. Studies in literature proposes using minimum 2^{2n+1} particles to efficiently represent the state density for particle filter which has exactly the same problem. However, this proposal cannot be applied to SLAM based VINS since system is high dimensional. In this study, 200 components are used in GMF. Initial components are composed of sigma points of unscented transform and random components chosen around mean according to initial covariances. Altought starting with similar components by following this approach, UKF and GMF differs after first INS step, since components are not dynamically generated at GMF.

In mapless VINS where all landmarks are initialized with respect to mobile platform's position and attitude, UKF and GMF follow different strategies. Sigma points are dynamically generated at every INS step of UKF and they also cover probability density function of position of landmarks but initialized with respect to one state which is weighted sum of sigma points. Landmarks are initialized individually for all components in GMF since components are not dynamically generated and their hypothesis about navigation state is different.

State prediction of GMF propagates mean and covariance of Gaussian components until a measurement is received. After initialization, all GMF components are propagated via nonlinear system dynamics individually as follows:

$$\hat{\boldsymbol{x}}^{i}_{k|k-1} = f_k(\hat{\boldsymbol{x}}^{i}_{k-1|k-1}, \boldsymbol{u}_k) \tag{4.89}$$

where $\hat{\boldsymbol{x}}^{i}_{k-1|k-1}$ represents the mean of i-th Gaussian component. Similarly propagation of covariance of i-th Gaussian component $\boldsymbol{P}^{i}_{k-1|k-1}$ is written as below:

$$\boldsymbol{P}^{i}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}^{i}_{k-1|k-1} \boldsymbol{F}^{T}_{k} + \boldsymbol{Q}_k \tag{4.90}$$

In order to obtain a final decision about system state and its covariance, weighted sum of components is used.

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=1}^{N} w_i \hat{\boldsymbol{x}}^{i}_{k|k-1} \tag{4.91}$$

$$\boldsymbol{P}_{k|k-1} = \sum_{i=1}^{N} w_i [\boldsymbol{P}^{i}_{k|k-1} + (\hat{\boldsymbol{x}}_{k|k-1} - \hat{\boldsymbol{x}}^{i}_{k|k-1})(\hat{\boldsymbol{x}}_{k|k-1} - \hat{\boldsymbol{x}}^{i}_{k|k-1})^T] \tag{4.92}$$

Innovation step updates state, weights and components of GMF when measurement is received. The expected measurement for i-th Gaussian component is calculated as follows:

$$\hat{\boldsymbol{z}}^{i}_{k|k-1} = h(\hat{\boldsymbol{x}}^{i}_{k|k-1}) \tag{4.93}$$

where $\hat{\boldsymbol{z}}^{i}_{k|k-1}$ represents the predicted measurement calculated from i-th Gaussian component. Innovation (or measurement) residual for i-th component is calculated as follows:

$$\hat{\boldsymbol{y}}^{i}_{k} = \boldsymbol{z}^{i}_{k} - \hat{\boldsymbol{z}}^{i}_{k|k-1} \tag{4.94}$$

Innovation (or residual) covariance of i-th component can be written as:

$$\boldsymbol{S}^{i}_{k} = \boldsymbol{H}^{i}_{k} \boldsymbol{P}^{i}_{k|k-1} {\boldsymbol{H}^{i}_{k}}^{T} + \boldsymbol{R}_k \tag{4.95}$$

where $\boldsymbol{H}^{i}_{k}$ is the linearized perspective projective matrix w.r.t i-th Gaussian component. Normalized innovation square used for filtering and weighting the components is written as:

$$NIS = \hat{\boldsymbol{y}}^{T}_{i} {\boldsymbol{S}_i}^{-1} \hat{\boldsymbol{y}}_i \tag{4.96}$$

Selection of weighting method for GMF components plays an important role on error characteristic and stability of filter. This study follows the traditional approach in literature proposed for both particle filters and GMFs. GMF components are weighted according to normalized innovation square (NIS) value of components. Weights of the components are calculated as:

$$w_i = w_i^p * \frac{1}{\sqrt{(2\pi)^l \left|\boldsymbol{S}_i\right|}} e^{-0.5(\hat{\boldsymbol{y}}_i^T \boldsymbol{S}_i{}^{-1} \hat{\boldsymbol{y}}_i)} \tag{4.97}$$

where w_i is the unnormalized new weight, w_i^p weight before update, l is the number of dimensions, $|S_i|$ is the determinant of matrix S_i, S_i is innovation covariance and $\hat{\boldsymbol{y}}_i$ is innovation of i-th component. GMF is a system with memory as one can see immediately from Eq. 4.97. The NIS value represents the possibility of having a true state under observation (measurement) and vice versa. Therefore it provides an information about how close a state estimate to its true value. NIS values are calculated for all Gaussian component individually. Means and covariances of components which cannot pass Chi-Square test are not updated. This is because of they increase navigation solution errors even if their weights are small and there is not an optimal way of weighting. Therefore they are not allowed to contribute Gaussian sum.

Kalman gain of components which can pass Chi-Square test are calculated as follows:

$$\boldsymbol{K}_k^i = \boldsymbol{P}_{k|k-1}^i {\boldsymbol{H}_k^i}^T {\boldsymbol{S}_k^i}^{-1} \tag{4.98}$$

State and covariance prediction of i-th Gaussian component is updated as follows:

$$\hat{\boldsymbol{x}}_{k|k}^i = \hat{\boldsymbol{x}}_{k-1|k-1}^i + \boldsymbol{K}_k^i \hat{\boldsymbol{y}}_k^i \tag{4.99}$$

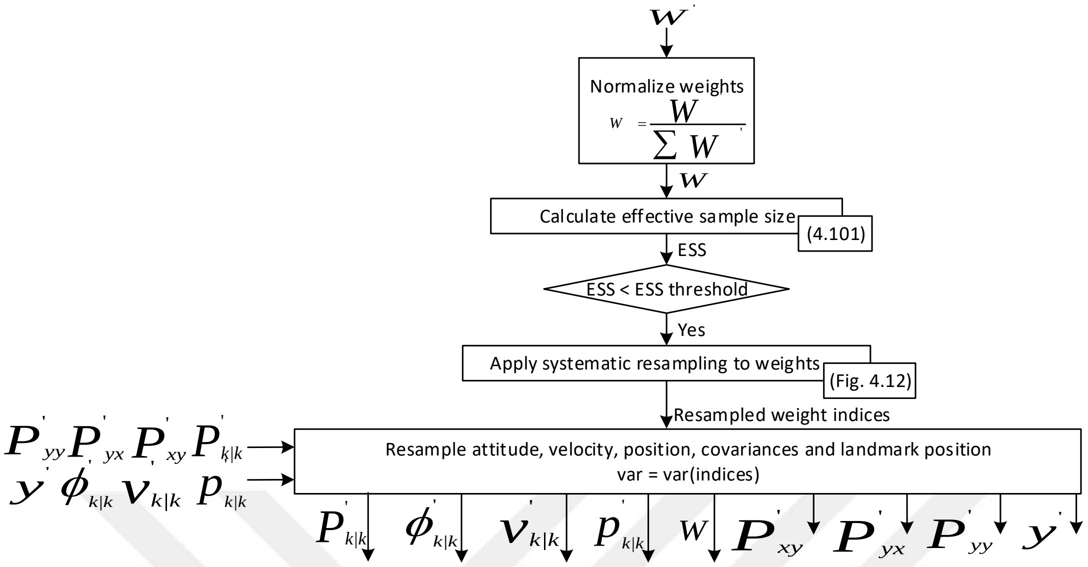
$$\boldsymbol{P}_{k|k}^i = [\boldsymbol{I} - \boldsymbol{K}_k^i \boldsymbol{H}_k^i] \boldsymbol{P}_{k|k-1}^i \tag{4.100}$$

Decreasing the effect of components with higher NIS values causes lack of components for efficiently representing the state density. This is a well-known phenomenon in literature and referred as "sample degeneracy" problem.

Solution proposed in the literature for sample degeneracy problem is replacing the ineffective components with useful ones. This is referred as "resampling". Resampling for GMF is proposed in several studies (i.e. [40], [37]) and also used in this study.

Main flowchart of SLAM based VINS using Indirect Gaussian Mixture Filter(I-GMF) can be found in Fig. 4.14. Since propagation and update steps for individual Gaussian components are same with I-EKF, detailed steps can be found in Figs. 4.6 and

4.7 respectively. Inputs block is elaborated in Fig. 4.4 and resampling algorithm 's detailed flowchart is given in Fig. 4.13.



Normalize weights
Calculate effective sample size
(4.101)
ESS
ESS < ESS threshold
Yes
Apply systematic resampling to weights
(Fig. 4.12)
Resampled weight indices
Resample attitude, velocity, position, covariances and landmark position
var = var(indices)

Figure 4.13: Resampling Main Flowchart

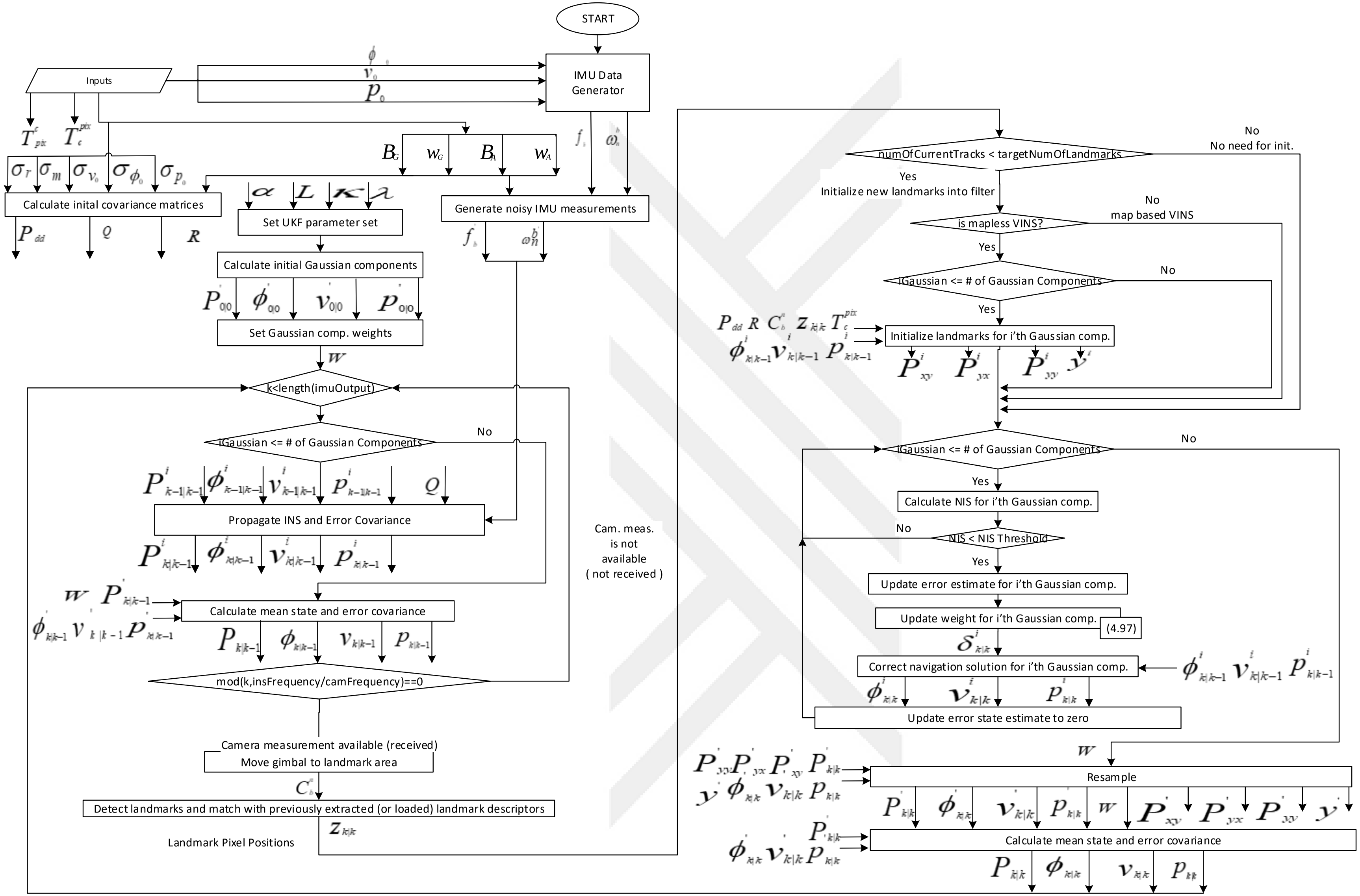


Figure 4.14: I-GMF Based VINS Main Flowchart

4.4.1 Resampling

Filter weights tend to be dominated by fewer number of components having higher weights. This makes most of the components ineffective and cause loosing advantage of using multiple components in filter. This problem is referred as sample depletion or sample degeneracy in literature. In order to solve this problem, components with lower weights can be replaced with the components with the higher weights. This approach is referred as "resampling".

Weight of a component represents its probability of being true state. Therefore components having higher weights replicated more on resampled component set. Different resampling schemes such as multinominal resampling, residual resampling, stratified resampling and systematic resampling are proposed in literature. These methods follows different strategies for determining number of component repetitions in resampled set.

Resampling algortihms are applied when effective number of components falls below a predetermined threshold. This threshold is typically selected as a ratio of number of components such as $\%90$.

Effective number of components can be determined with the help of variance of component weights as below [19]:

$$N_{eff} = \frac{N}{1 + N^2 var(W)} \tag{4.101}$$

where N represents the number of components and W represents the set of weights. Eq. 4.101 reaches its maximum value when all weights are equal which makes variance term zero and $N_{eff} = N$ and it is indirectly proportional with the variance of weight set.

Results of papers where resampling methods are comparatively studied shows that systematic resampling outperforms in term of mean square errors of states compared to other methods [26], [33]. This result is also verified in this study. Therefore systematic resampling is used. In order to keep the integrity of the subject, only systematic resampling will be elaborated in this section.

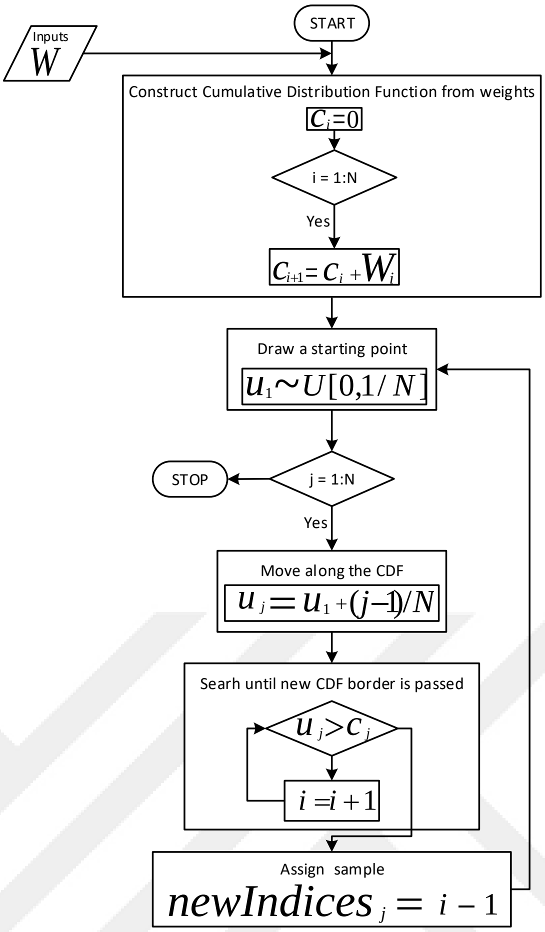


Figure 4.15: Systematic Resampling Flowchart [3]

Systematic resampling aims to minimize the variance term in the denominator of Eq. 4.101 to maximize the effective number of samples [42]. It moves along CDF of weights and samples distribution. While moving along CDF, if sampled point corresponds to a new CDF interval, corresponding interval's component is replicated otherwise deleted. Systematic resampling flowchart [3] is provided in Fig. 4.15 .

Although resampling is the most common method, there are also other methods proposed in literature in order to solve sample degeneracy problem. Some examples are; adding independent zero mean and constant variance Gaussian noise to components after resampling which is referred as jittering [46], modifying noise covariances used in filter and regularization method proposed in [38] convolves each component with a kernel after resampling to diversify the components.

4.5 An Efficient Feature Region Selection Method for Simultaneously Localization and Mapping

Visual Aided Inertial Navigation Systems use landmark based maps. These may consist of low-level visual landmark points such as blobs and corners or may be landmarks based on an image patch that can be consistently matched between frames. Feature matching both refines the position, velocity and attitude of the mobile platform as well as refining or estimating from scratch the positions of these visual landmarks in the global navigation frame. Tracking a visual landmark as long as possible enables the system to get updates from this landmark hence improving all joint estimates. In order to track visual landmarks as long as possible, we propose a feature region selection method that considers both the quality as well the position of the landmark within the image frame.

On fast moving mobile platforms, observing the same scene for a long time is not possible. In order to obtain high estimation performance, these landmarks must be observed for multiple frames. This may be possible only with a very high frame-rate (with matching computational performance) or with a gimbaled (movable) camera system that can keep the features inside the frame longer. Because of prohibitive computational burden of the former approach, the latter one is the considered approach in our study.

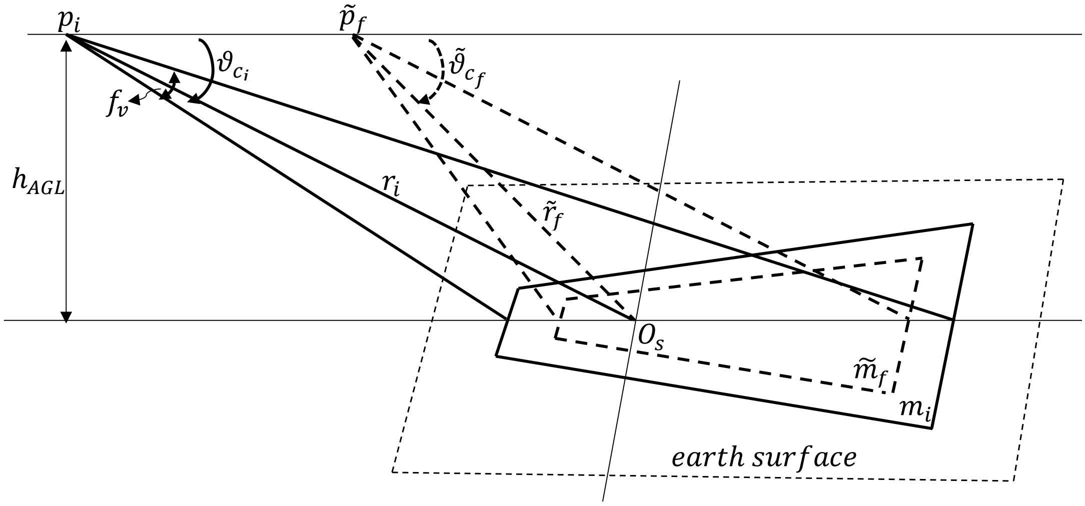


Figure 4.16: Camera Gimbal Angle Control and Imaging Geometry in terms of Vertical Field of View

Fig. 4.16 shows a possible camera gimbal control method for observing the same scene for a long time. Camera gimbal can be moved along roll axis of c-frame define in Fig. 3.3. In this way, center of the scene defined by O_s in Fig. 4.16 is tracked and similar scene is observed for a long time.

If ground plane assumption is made, imaged scene with an angled camera corresponds to a trapezoid on the Earth. If the imaged scene at i-th step shown by m_i trapezoid in Fig. 4.16 is tracked by moving the camera gimbal along roll axis of c-frame, m_i trapezoid shrinks along both perpendicular and horizontal axis as the platform moves towards $\tilde{p}_f$. The final imaged scene at f-th step shown by $\tilde{m}_f$ and m_i trapezoid overlaps as shown in Fig. 4.16. Note that this assumption holds when mobile platform moves along same direction from i-th step to f-th step.

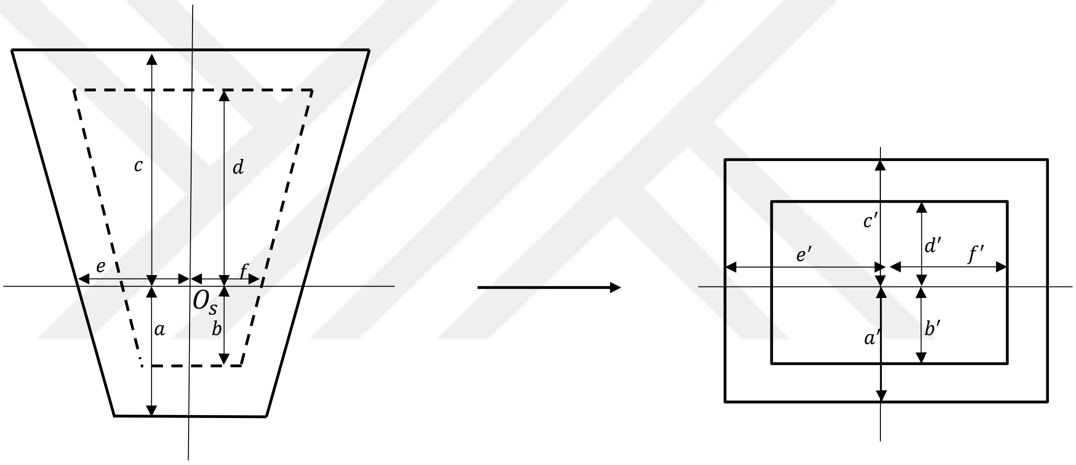


Figure 4.17: Images are projection of trapezoid areas in real world to 2D image plane.

Images of these trapezoid areas actually transferred to pixels by optoelectronic systems. This transfer is shown in Fig. 4.17. Borders of final image which overlaps with i-th image can be estimated if the last angle of gimbal in roll axis is known. This corresponds physical gimbal angle limit in our case.

This estimation can be used in data association as a feature region selection constraint. Since only the feature points inside of the estimated borders will be visible in final image, feature points started to be tracked in i-th step can be selected by using these estimated borders. A filter selecting the features estimated to be visible in the final image is written in terms of the quantities defined in Fig. 4.17 as:

$$f(pix_x, pix_y) = \begin{cases} (pix_x, pix_y) & [(1-\frac{b^{'}}{a^{'}})\frac{res_x}{2} \leq pix_x \leq \frac{res_x}{2} + \frac{d^{'}}{c^{'}}\frac{res_x}{2}]\& \\ & [(1-\frac{f^{'}}{e^{'}})\frac{res_y}{2} \leq pix_y \leq \frac{res_y}{2} + \frac{f^{'}}{e^{'}}\frac{res_y}{2}] \\ delete\, feature\, point, & otherwise \end{cases} \tag{4.102}$$

The imaged area of a pixel in real world is referred as pixel footprint. The footprint of each pixel is different and increases along x-axis of image plane. The length ratio of m_i and $\tilde{m}_f$ trapezoid under and above the O_s are also different. Parts of trapezoids in Fig. 4.17 are defined and Eq. 4.102 is written accordingly. Lengths of variables defined on trapezoid is proportional to lengths defined on image plane. Therefore it is possible to write:

$$\frac{b^{'}}{a^{'}} = \frac{b}{a} \ , \ \frac{d^{'}}{c^{'}} = \frac{d}{c} \ , \ \frac{f^{'}}{e^{'}} = \frac{f}{e} \tag{4.103}$$

Fig. 4.16, p_i shows the position of the mobile platform and θ_{c_i} represents the gimbal angle in roll axis at i-th step. The camera field of view along vertical axis of image plane is f_v. r_i shows the slant range between origin of c-frame and O_s. $\tilde{\theta}_{c_f}$ is the camera gimbal limit in roll axis of the camera. r_f shows calculated the slant range between origin of c-frame and O_s at f-th frame. In terms of these variables, lengths on Fig. 4.17 constituting elements of Eq. 4.102 can be written as:

$$a = r_i cos(\theta_{c_i}) - \frac{r_i sin(\theta_{c_i})}{sin(\theta_{c_i} + \frac{f_v}{2})} cos(\theta_{c_i} + \frac{f_v}{2}) \tag{4.104}$$

$$b = \tilde{r}_f cos(\tilde{\theta}_{c_f}) - \frac{\tilde{r}_f sin(\tilde{\theta}_{c_f})}{sin(\tilde{\theta}_{c_f} + \frac{f_v}{2})} cos(\tilde{\theta}_{c_f} + \frac{f_v}{2}) \tag{4.105}$$

$$c = \frac{r_i sin(\theta_{c_i})}{sin(\theta_{c_i} - \frac{f_v}{2})} cos(\theta_{c_i} - \frac{f_v}{2}) - r_i cos(\theta_{c_i}) \tag{4.106}$$

$$d = \frac{\tilde{r}_f sin(\tilde{\theta}_{c_f})}{sin(\tilde{\theta}_{c_f} - \frac{f_v}{2})} cos(\tilde{\theta}_{c_f} - \frac{f_v}{2}) - \tilde{r}_f cos(\tilde{\theta}_{c_f}) \tag{4.107}$$

Although trapezoid extents along vertical axis, in order to simply the calculations, length of trapezoid along horizontal axis assumed to be constant and equal to it length at O_s as shown in Fig. 4.18.

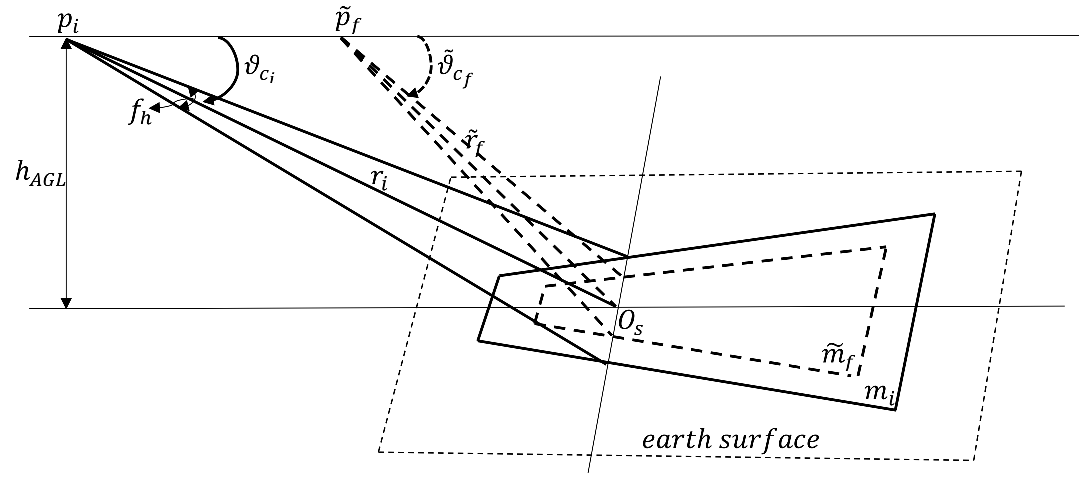


Figure 4.18: Imaging Geometry in terms of Horizontal Field of View

In Fig. 4.18, f_h represent the camera field of view along horizontal axis of image plane. The lengths defined along horizontal axis in Fig. 4.17 is written as:

$$e = r_i tan(f_h/2) \tag{4.108}$$

$$f = \tilde{r}_f tan(f_h/2) \tag{4.109}$$

The slant range between origin of c-frame and O_s at f-th frame is calculated as:

$$\tilde{r}_f = r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f})} \tag{4.110}$$

Substituting $\tilde{r}_f$ into Eqs. 4.105, 4.107 and 4.109 yields:

$$b = r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f})} cos(\tilde{\theta}_{c_f}) - r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f} + \frac{f_v}{2})} cos(\tilde{\theta}_{c_f} + \frac{f_v}{2}) \tag{4.111}$$

$$d = r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f} - \frac{f_v}{2})} cos(\tilde{\theta}_{c_f}) - r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f})} cos(\tilde{\theta}_{c_f} + \frac{f_v}{2}) \tag{4.112}$$

$$f = r_i \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f})} tan(\frac{f_h}{2}) \tag{4.113}$$

Elements of Eq. 4.102 can be calculated by substituting derived lengths into Eq. 4.102:

$$\frac{b}{a} = \frac{cot(\tilde{\theta}_{c_f}) - cot(\tilde{\theta}_{c_f} + \frac{f_v}{2}))}{cot(\theta_{c_i}) - cot(\theta_{c_i} + \frac{f_v}{2}))} \tag{4.114}$$

$$\frac{d}{c} = \frac{cot(\tilde{\theta}_{c_f} - \frac{f_v}{2}) - cot(\tilde{\theta}_{c_f})}{cot(\theta_{c_i} - \frac{f_v}{2}) - cot(\theta_{c_i})} \tag{4.115}$$

$$\frac{f}{e} = \frac{sin(\theta_{c_i})}{sin(\tilde{\theta}_{c_f})} \tag{4.116}$$

This concludes the derivation of elements of Eq. 4.102. The method can also be referred as visibility filter.

Proposed method not only maximizes the number of track frames, it also selects the feature points affected minimally by imaging sensor's measurement errors since optic errors increase as moved away from center of optic elements.

CHAPTER 5

COMPARATIVE EVALUATION OF VINS ALGORITHMS IN A SIMULATION ENVIRONMENT

5.1 Methodology

In order to construct a simulation experimental environment to comparatively test VINS scenarios and data fusion algorithms, one needs to have the following simulation components:

1. A selection of flight scenarios (trajectories and ground feature points),
2. An IMU data simulator, generating IMU sensor data consistent with flight trajectory,
3. A camera simulator to transform ground feature points into image sensor measurements,
4. A camera gimbal simulator to simulate camera motion to determine its line-of-sight,
5. Implementation of an INS-only navigation sub-system,
6. Implementation of alternative existing and proposed VINS feature detection and Bayesian data fusion algorithms.

Due to the lack of available facilities and real-data, we are required to work in a simulation environment. Therefore, the reliability of our results depend on the fidelity of the simulation environment as well as on the algorithms tested. To increase our confidence in the simulation implementation, we followed a unit testing approach where we have tried to validate each enumerated stage of the simulation independently as described below.

1. We have selected a number of scenarios covering meaningful operational test cases. These will be described briefly in this section,

2. The IMU data generator was supplied by the institution supporting this study and was beyond our focus. We consider this sub-system to be fairly reliable as it has been extensively tested apriori to the present study.

3. The camera simulator is a simple pinhole camera model, implementing the perspective projection of 3D world feature points into point measurements on the sensor. Lens distortion is neglected due to the small field-of-view of the optics in our application,

4. The camera gimbal is modeled with a simple pitch-yaw motion that can follow a feature "scene" on the ground to keep the feature point content as constant as possible. In practice, the content will vary due to changing distance (camera field-of-view) and imaging geometry.

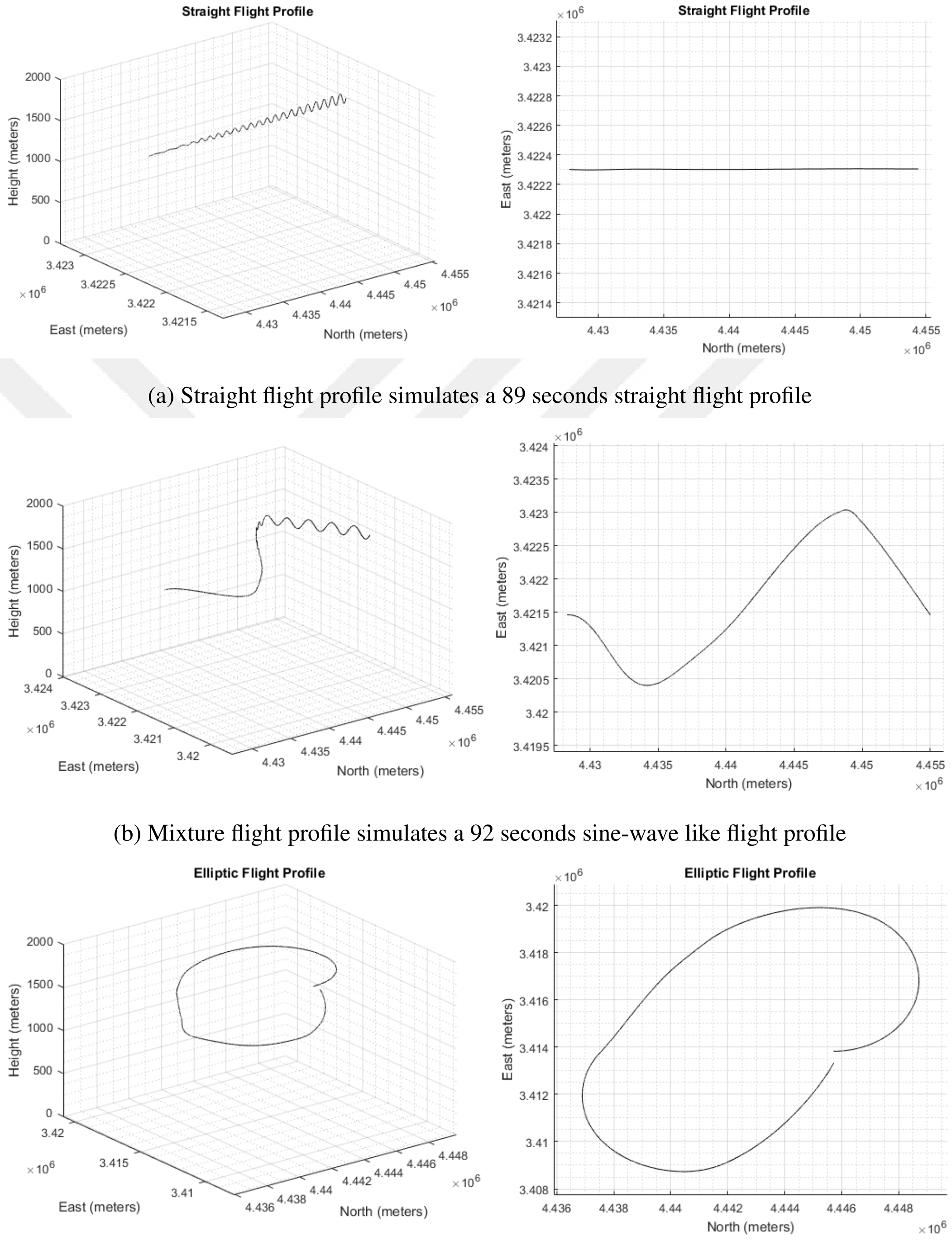
5. We have implemented ourselves a full INS-only navigation sub-system by implementing well established strapdown equations explained in Chapter 3.2. The accuracy of this sub-system is validated by comparing its output at certain waypoints with the ground truth supplied by the external IMU data generator in the form of a validation file.

6. Considering the Bayesian fusion algorithms in the study, we have firstly implemented a fairly standard GPS-INS fusion algorithm following the Indirect Kalman Filtering approach. This is less complex than the VINS problem, hence gives us opportunity to validate our approach.

7. In order to verify the integration of camera model and gimbal control model to sensor fusion filter, Map-Based VINS which is less complex compared to mapless VINS is implemented. Performance versus critical design parameters are analyzed. Since expectation of performance analyses are satisfied, camera integration is determined to be successful.

8. As the final step, mapless VINS is implemented and analyzed. A Gaussian Filter Framework and a feature region selection method were proposed.

Since gyroscope and accelerometer of the IMU have different error characteristics, performance differs for each flight profiles. Therefore analyses are conducted on 3 flight profiles given in Fig. 5.1. Length of all paths approximately equal and 27 kilometers.



(a) Straight flight profile simulates a 89 seconds straight flight profile

(b) Mixture flight profile simulates a 92 seconds sine-wave like flight profile

(c) Elliptic flight profile simulates a 116 seconds flight profile containing circular and straight sub profiles

Figure 5.1: Tested Flight Profiles

All tests are performed on a simulated UAV having configurations given in Table 5.1.

Table 5.1: Tested UAV Configuration

UAV Parameters	Values
Flight Speed	300m/s (∼0.88 Machs)
Flight Altitude Above Ground Level	1500 m
Depth Sensor	LIDAR
Depth Sensor Accuracy	0.1 m

The navigation solution error mean and standard deviation of aided system in VINS depends on specifications of IMU. In this study a low cost tactical grade IMU is chosen and IMU error sources are modeled according to datasheet provided by manufacturer.

Table 5.2: IMU Specifications

IMU Specifications	Value
IMU Frequency	100 Hz
Accelerometer Bias	$1 \times 10^{-3}\ g$
Accelerometer Random Walk	$85 \times 10^{-6}\ g$
Gyroscope Bias	$1°/hr$
Gyroscope Random Walk	$0.125°/\sqrt{hz}$

A hypothetical camera with a realizable field of view is considered and used in our experiments. Distortion parameters were neglected due to having a narrow viewing angle.

Table 5.3: Simulated Camera Configurations

Camera Parameters	Values
Camera Resolution	1280 x 720
Camera Processing Frame Rate	1 Hz
Camera Field of View	10° x 10 °
Measurement Noise	$N(0, 0.5)$

In order to improve estimation performance, one needs to keep the same landmarks longer within the frame. For this the camera gimbal tracks the set of landmarks in view until it reaches its angular limit. When the limit is reached, the gimbal moves back to its starting angle. This angular sweep corresponds in our simulations to a number of image frames.

Measurements are generated by projecting the positions of landmark points which are blob-based features like SIFT and SURF feature points onto image plane pixel points and adding pixel position noise on top. Positions of landmarks are assigned in the vicinity of ground plane but in random manner.

Initial navigation solution errors have zero mean Gaussian distribution with standard deviations given in Table 5.4:

Table 5.4: Initial Navigation Solution Error Standard Deviations

Variable	**Std**
Latitude	$50\ m$
Longitude	$50\ m$
Height	$100\ m$
Roll/Pitch/Yaw	$0.005°$
Lat. / Lon./ Height Velocity	$0.5\ m/s$

GMF used in simulations is composed of linear combination of 200 Gaussian components. Systematic resampling is applied when effective number of Gaussian components fall below 180.

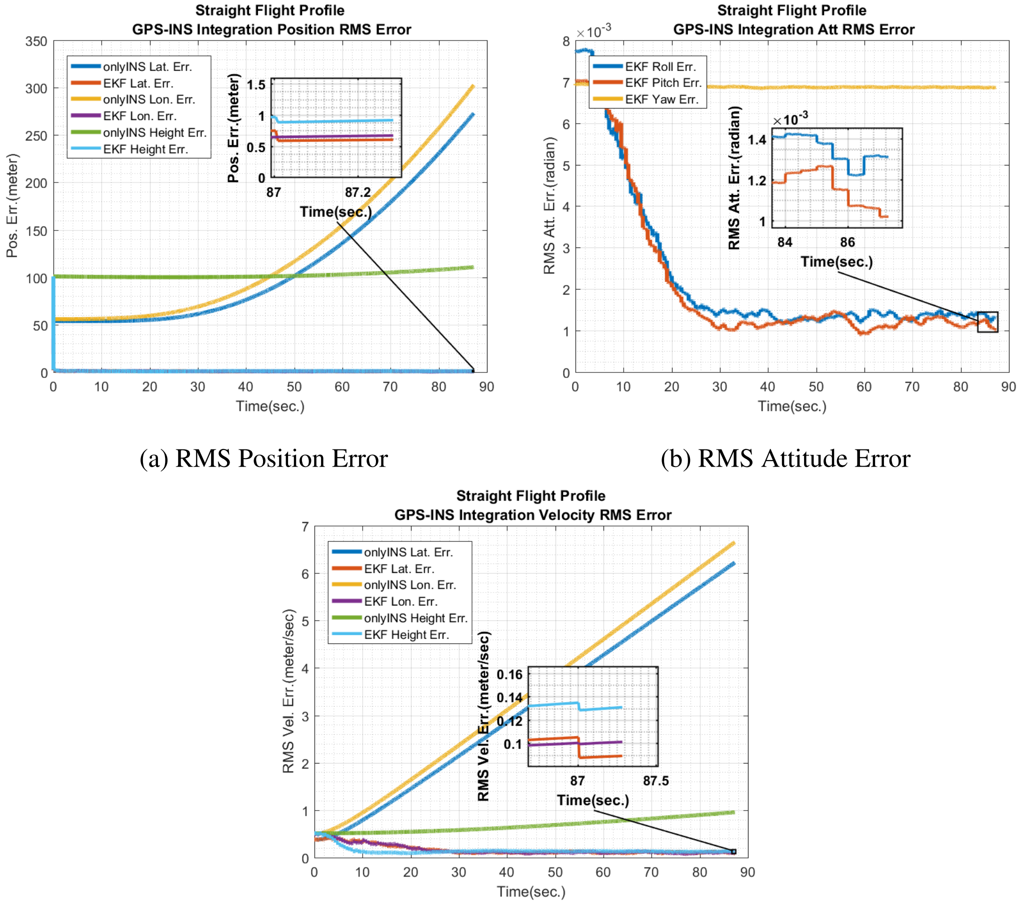
Number of dimensions in UKF, L equals to 45 since 12 landmarks are used. Since $L = 45$, 91 sigma points are propagated. Parameters of UKF is set to $\alpha = 10^{-1}$, $\beta = 2$, $\kappa = 10000$. Using this parameter set yields weighting central component of sigma points 110 times more than off-central sigma points.

Because of space considerations, only straight flight profile's analysis results will be shown in figures. Mean of last 200 INS step's navigation solution errors and standard deviations will be given in tables for showing the results of mixture and elliptic flight profiles.

Finally, all Monte Carlo simulations were run on a computer with hardware setup; i7-5700HQ @ 2.7 GHz CPU, 8GB-DDR3 RAM using MATLAB 2016a environment.

5.1.1 Evaluation of EKF based GPS-INS Integration

GPS-INS integration is a navigation technique that integrates the external position information provided by the satellite networks to navigation solution of INS using a sensor fusion filter. It is relatively a simpler system compared to VINS, since less number of sub-systems and operations are included. Therefore before implementing a VINS, an EKF based loosely coupled GPS-INS integration system is implemented in order to verify the Bayesian sensor fusion filter, linearization and discretization of error state dynamics and other operations that will establish a basis for VINS. GPS sensor errors are modelled as zero mean 5 meter standard deviation Gaussian random variable. 50 Monte Carlo runs are carried out.

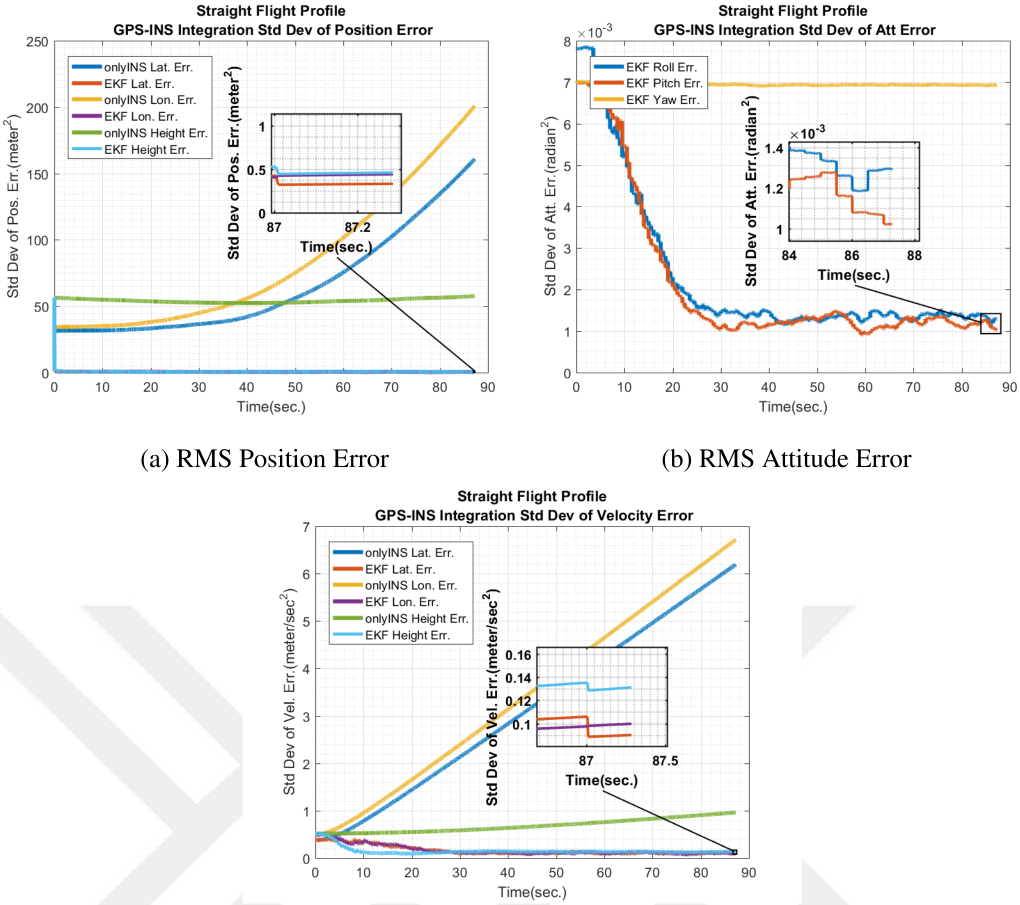


(a) RMS Position Error

(b) RMS Attitude Error

(c) RMS Velocity Error

Figure 5.2: Straight Flight Profile: GPS-INS Integration RMS Navigation Solution Errors



(a) RMS Position Error

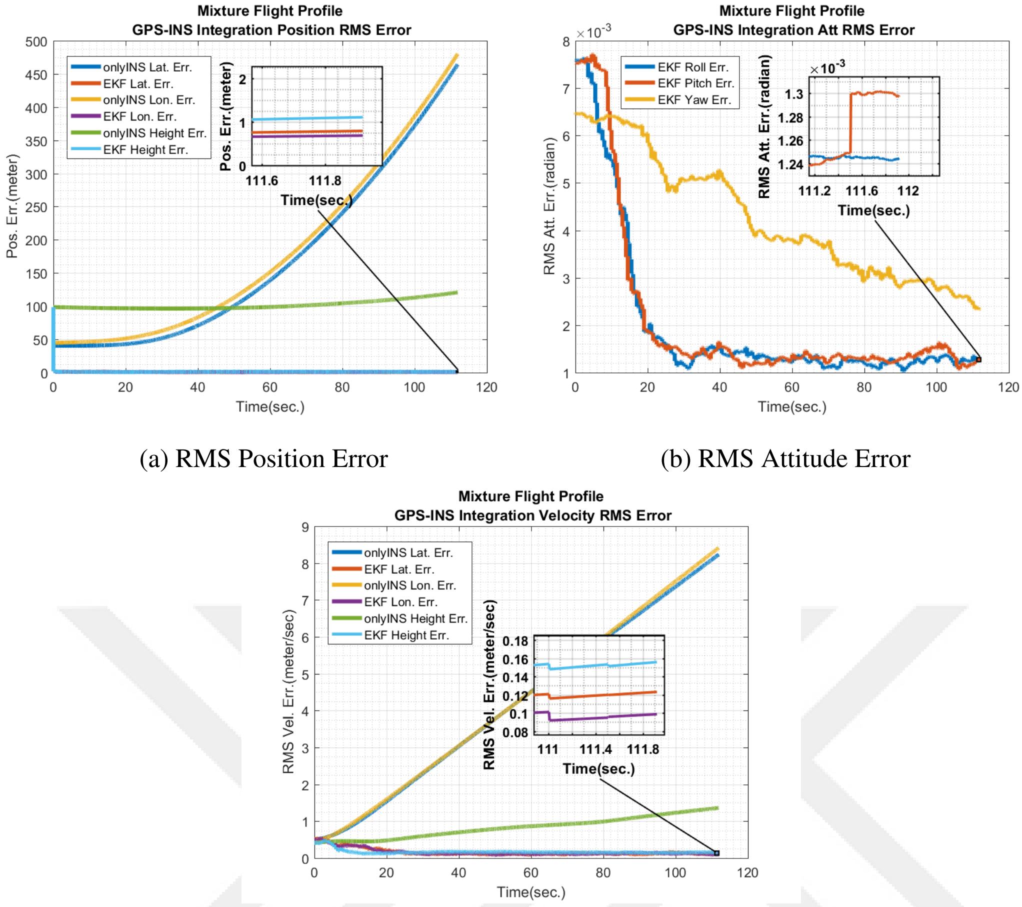
(b) RMS Attitude Error

(c) RMS Velocity Error

Figure 5.3: Straight Flight Profile: GPS-INS Integration Std. Dev. of Navigation Solution Errors

Fig. 5.2 and 5.3 shows the RMS and standard deviation of navigation solution errors. Since GPS provides an external position information it is possible to decrease the position and velocity errors to vicinity of zero. This expectation is also verified in our simulations as shown in Fig. 5.2.

EKF based loosely coupled GPS-INS integration is able to refine the roll and pitch errors since they remain observable under all circumtances but yaw axis become unobservable when the mobile platform has constant flight velocity [13] [17]. Under the straight flight profile scenario of this study, the mobile platform almost moves at constant velocity since it doesn't have to accelerate for slowing down or speeding up for reaching the defined flight velocity at the beginning of flight.



(a) RMS Position Error

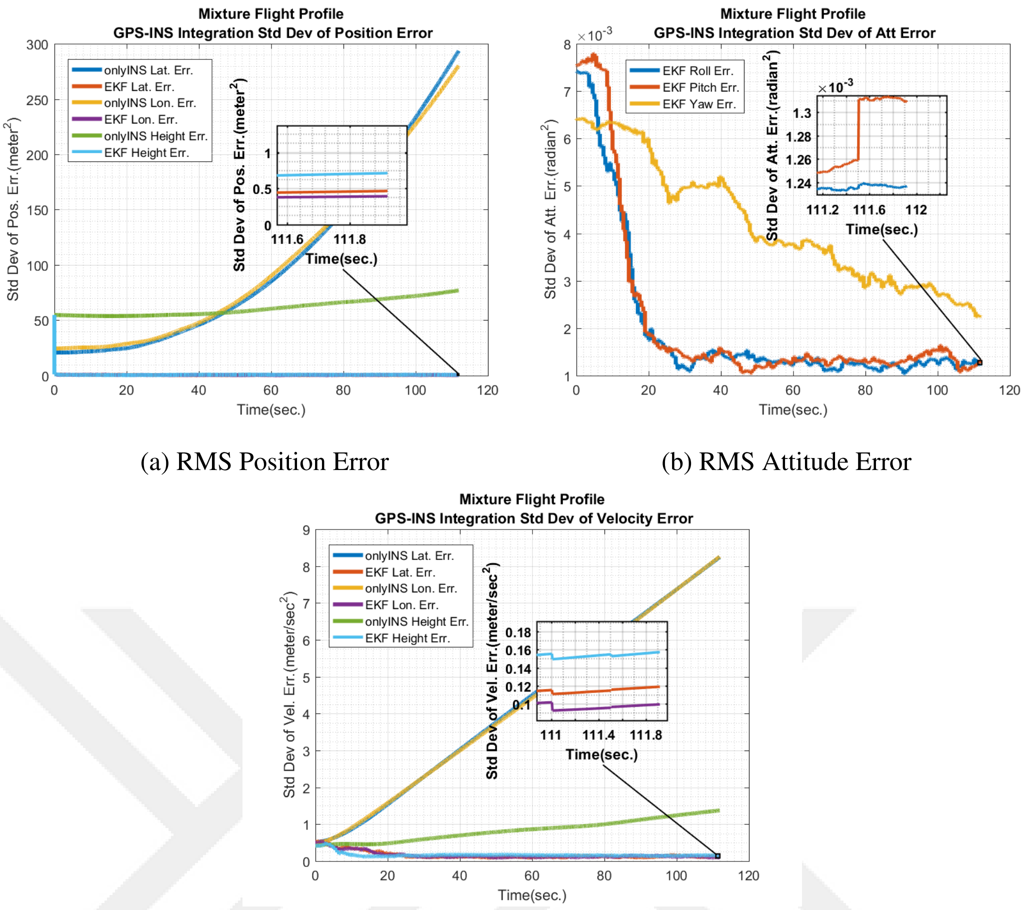
(b) RMS Attitude Error

(c) RMS Velocity Error

Figure 5.4: Mixture Flight Profile: GPS-INS Integration RMS Navigation Solution Errors

Attitude errors about all axes are able to be refined in straight flight profile however because of the low observability, refinement about roll axis is lower. In mixture flight profile, mobile platform have to accelerate for turning maneuvers, therefore refinements in yaw axis are higher because of higher observability as shown in Fig. 5.4 and 5.5.

These results show that our Bayesian filter is not getting any spurious information and other sub-systems are integrated correctly. Therefore they are decided to be suitable to establish a more complex system which is the VINS in our study.



(a) RMS Position Error

(b) RMS Attitude Error

(c) RMS Velocity Error

Figure 5.5: Mixture Flight Profile: GPS-INS Integration Std. Dev. of Navigation Solution Errors

5.2 Simulations on Map Based VINS

The performance in map based VINS are determined by many factors. Some of these factors are given below:

- Error sources; IMU errors, pixel measurement error, range error etc.,
- Data association accuracy,
- Slant range between landmark and the mobile platform,
- Selection of sensor fusion filter,
- Parameters related with sensor fusion filter such as number of landmarks, number of tracked frames etc..

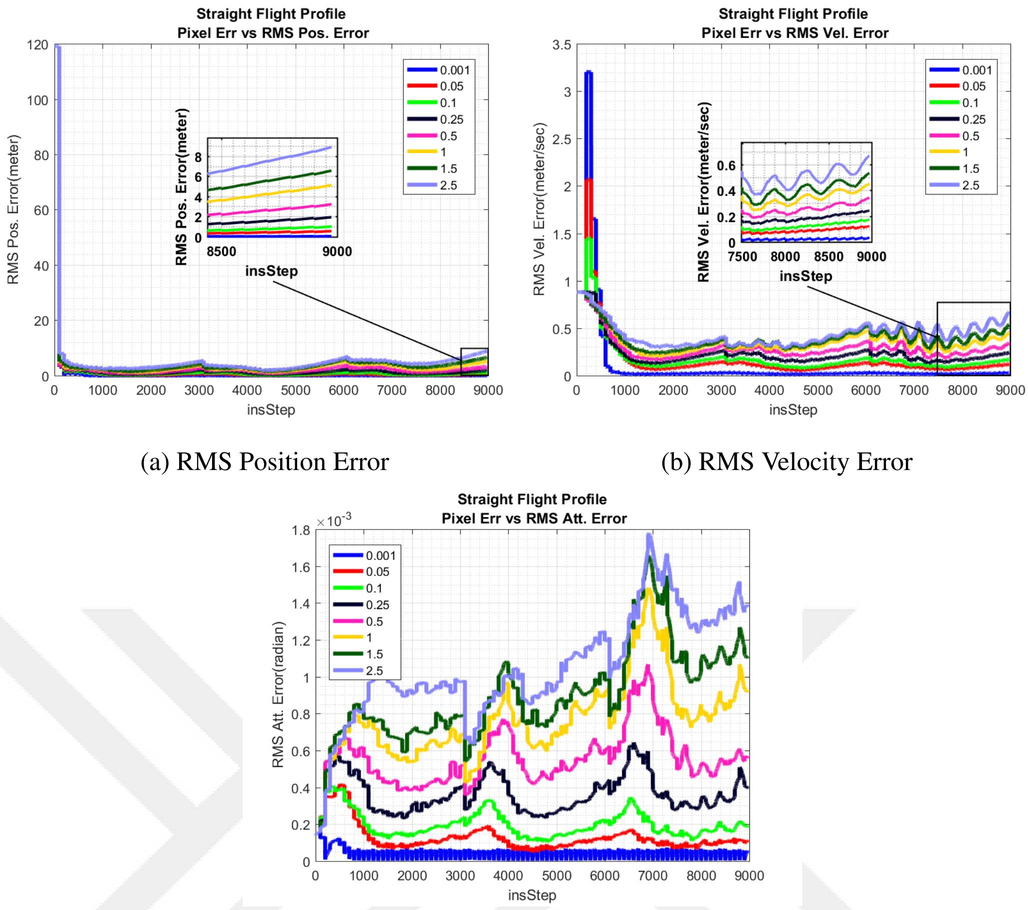
In order to study these parameters and sensor fusion filter on map-based VINS, simulations are conducted and presented in this section.

5.2.1 Effect of Measurement Noise on Map-Based VINS Performance

Pixel measurement error is one of the key parameters that determines performance upper bounds of VINS in term of navigation solution error. In order to investigate the sensivity of VINS on UAV applications to this variable, simulations are conducted.

The effect of pixel measurement error on the performance of VINS is also related to slant range between UAV and landmark in linear fashion. The back projected position error of landmark is directly proportional to the distance between UAV and landmark. Hence uncertainty of the information which will be provided to sensor fusion filter by landmark increases with this distance. In this study, distance between UAV and the region where landmarks are initially observed is approximately 2600 meters and decreases as UAV approaches to this region.

Pixel measurement error's standard deviation is set to 8 different values between 0.001 and 2.5 pixels. 50 monte carlo runs for each value are carried out. Norm of mean and standard deviation of navigation solution error are analyzed for different flight profiles. It is observed that minimum, intermediate and maximum values of simulated errors are enough to represent the results. Legends of the figures show the values of simulated pixel measurement error standard deviations.



(a) RMS Position Error

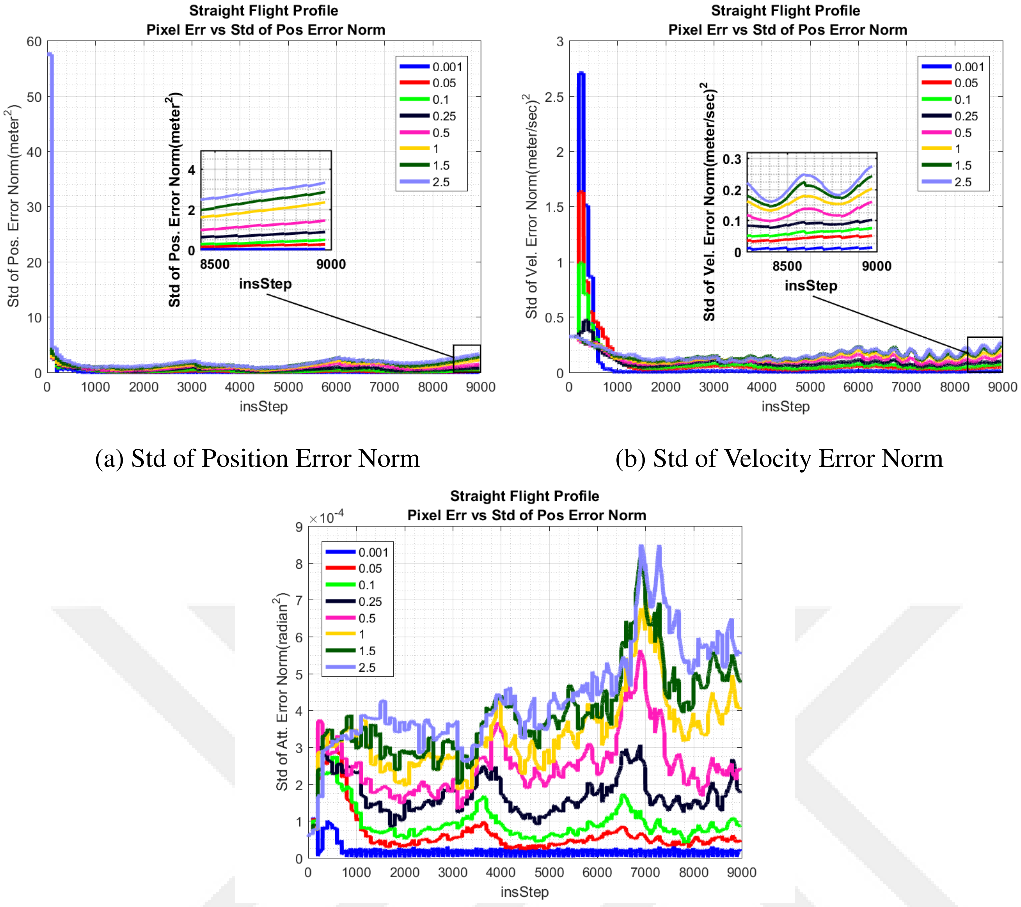
(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.6: Straight Flight Profile: RMS Navigation Solution Errors versus Pixel Errors(color traces)

Figure 5.6 shows norm of navigation solution error means versus different values of measurement error standard deviations for straight flight profiles. Results show that error levels and magnitudes of error oscillations increases exponentially with the increase of pixel measurement error for all navigation variables.

An interesting outcome is overshoot characteristics. Although having lower error bound for small measurement error standard deviations, overshoot of velocity error is higher. This originates from the fact that navigation solution error is higher in transition region and true states are needed to be searched in wider space. In that region, relying more on an external information might not provide accurate information. Therefore higher values of measurement error covariance smooths the errors on this transition region.



(a) Std of Position Error Norm

(b) Std of Velocity Error Norm

(c) Std of Attitude Error Norm

Figure 5.7: Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms versus Pixel Errors(color traces)

Figure 5.7 shows standard deviation of navigation solution error norms versus different values of measurement error standard deviations for all flight profiles. Results show same characteristics with the mean of navigation error solution. Hence implications from error norm results are valid for standard deviation given in Fig. 5.7

Observations are valid for all flight profiles. Therefore only norm of mean and standard deviation of error levels are provided for mixture and ellipsis flight profiles at Tables 5.5 and 5.6. All monte carlo runs have convergent error characteristic. Therefore stability results are not given as individual figures.

Table 5.5: Mixture Flight Profile: RMS and Std of Navigation Solution Errors

	Attitude Error		Velocity Error		Position Error	
Pixel Error Cov.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
pixel	rad	rad^2	m/s	$(m/s)^2$	m	m^2
0.001	3.09E-05	1.2E-05	0.02523	0.0109	0.029496	0.014695
0.25	0.00035	0.0002	0.25510	0.1017	1.5824	0.639092
2.5	0.00116	0.0006	0.58231	0.2331	5.6209	2.367605

Table 5.6: Ellipsis Flight Profile: RMS and Std of Navigation Solution Errors

	Attitude Error		Velocity Error		Position Error	
Pixel Error Cov.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
pixel	rad	rad^2	m/s	$(m/s)^2$	m	m^2
0.001	2.69E-05	8.32E-06	0.02573	0.00968	0.0276	0.01408
0.25	0.00042	0.00017	0.30499	0.09258	2.61412	1.01298
2.5	0.00122	0.00053	0.54954	0.18545	8.04655	3.17083

5.2.2 Joint Analysis of Landmarks Position Uncertainty and Number of Tracked Landmarks on Map-Based VINS Performance

In map based VINS, landmark's position informations must be loaded to mobile platform before flight. Because of the uncertainties of measurement devices and errors on fixing position information to visual data at post processing step of preplaning phase, landmark positions can only be known with an uncertainty. Because of having non-zero mean and non-Gaussian distribution, having landmark position uncertainty violates the assumption of EKF. Therefore it is one of the reasons causing sub optimal performance in sensor fusion.

A method for decreasing the effect of landmark position uncertainty can be increasing the number of landmarks used in filter. EKF shares its "belief" to every landmark. It relies less on every landmark for the case where more landmarks are used. Hence effect of getting an update from a landmark having higher position error decreases. Moreover, fusing more external information can provide more accurate information to filter. Therefore by increasing the number of landmarks, it is possible to compensate the performance degradation caused by landmarks having more erroneous position information.

In Fig. 5.8, an example case is presented. $info_1$, $info_2$ and $info_3$ show the information provided by 3 landmarks indicated with lm_1, lm_2 and lm_3 respectively. Ellipses show the uncertainty around the information provided by landmark to filter. The information used by filter lies in intersection on ellipses. Intersection area represents the uncertainty of information. Number of ellipses increases with number of landmarks used in filter and intersection area decreases with number of ellipsis. This means that uncertainty on information provided to filter decreases with the increase on number of landmarks.

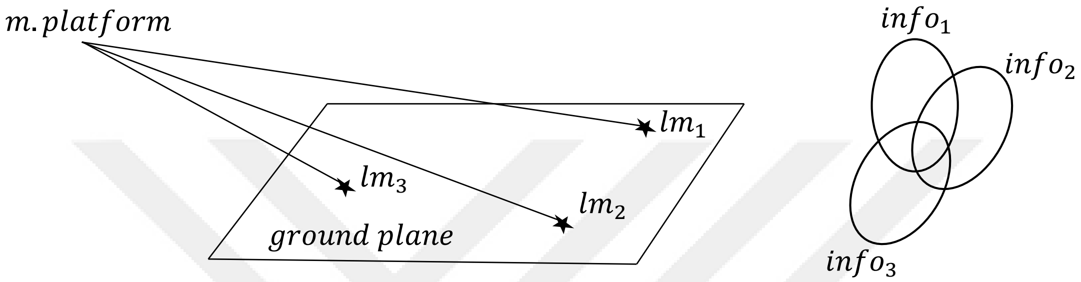
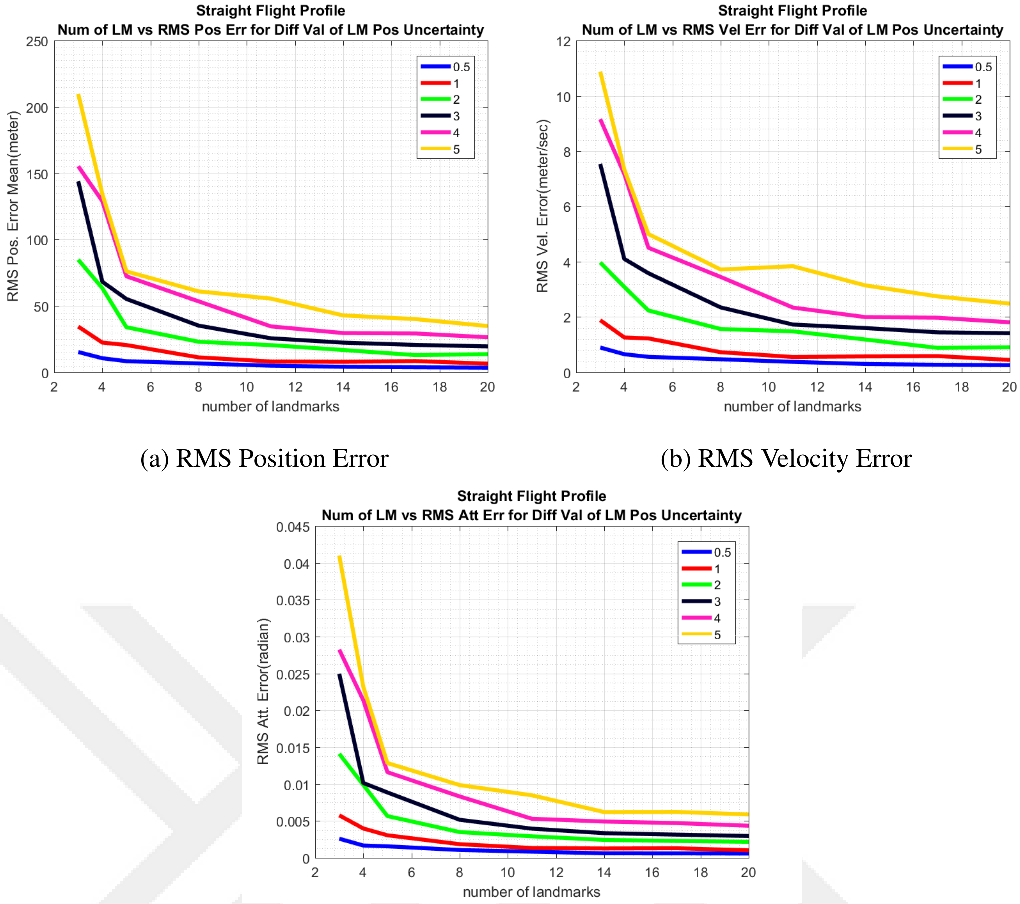


Figure 5.8: Advantage of Using Multiple Landmarks

In order to gain a deeper insight to concept explain above, number of landmark and landmark position uncertainty were jointly studied. 50 monte carlo runs for each standard deviation value were carried out. Norm of mean and standard deviation of navigation solution errors for each simulated value were analyzed for different flight profiles.

Simulations were conducted for 5 different values of landmark position uncertainties. It is observed that minimum, intermediate and maximum of simulated values are enough to represent the results. Legends of the Fig. 5.9 shows the values of simulated landmark position uncertainties.



(a) RMS Position Error

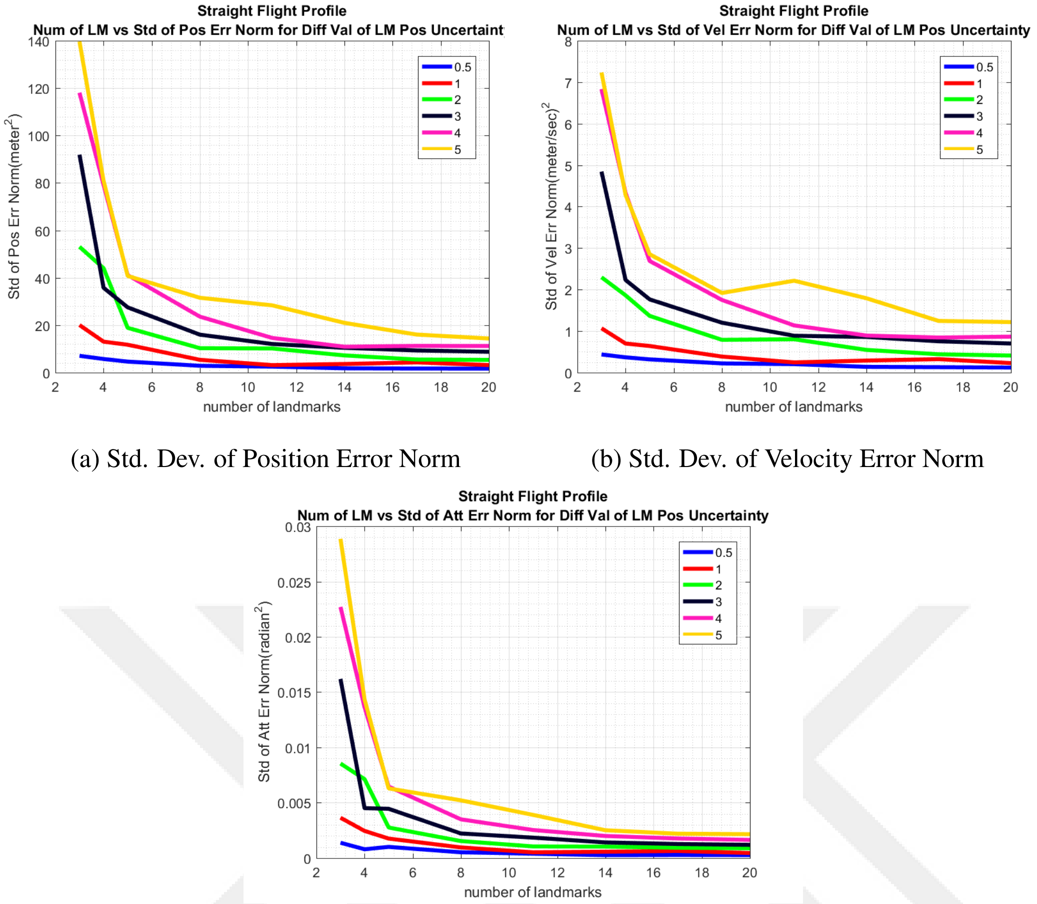
(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.9: Straight Flight Profile: RMS Navigation Solution Errors versus Number of Landmarks (x-axis) and Std. Dev. of Landmark Position Errors (color traces)

Figure 5.9 shows RMS navigation solution error versus number of landmark used in filter for straight flight profile.

Results verifies the expectations. Using more landmarks in filter can provide lower RMS navigation solution error for different values of landmark position uncertainties. RMS navigation solution error of 3 landmarks is noticeably higher compared to higher number of landmarks. This result also verifies the literature which proposes to use minimum 4 landmarks in order to have full observability.



(a) Std. Dev. of Position Error Norm

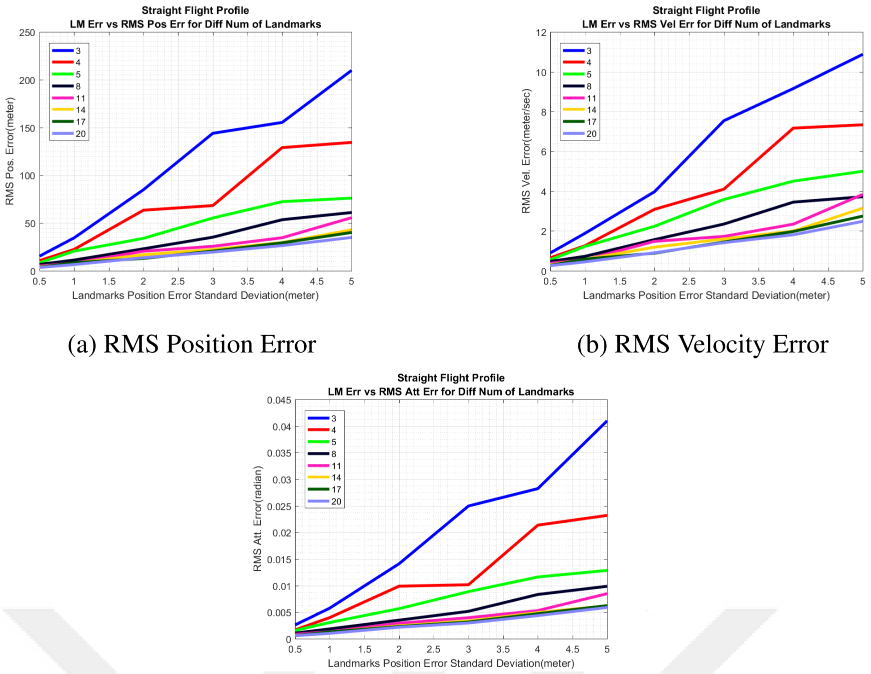
(b) Std. Dev. of Velocity Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.10: Straight Flight Profile: Std. Dev. of Navigation Solution Errors versus Num. of Landmarks (x-axis) and Std. Dev. of Landmark Pos. Errors (color traces)

Standard deviation of navigation solution versus different number of landmarks is given Fig. 5.10. Results show that using more landmarks in filter also decreases the standard deviation of navigation solution error. As the number of landmarks increases, error characteristic becomes more linear under increase of landmark position uncertainty.

Landmark position uncertainty increases the error levels and standard deviations exponentially as shown in Fig. 5.11 and 5.12 when number of landmarks used in filter is low. Using more landmarks damps the exponential increase of error under increase of landmark position uncertainty and can provide an almost linear error characteristic.

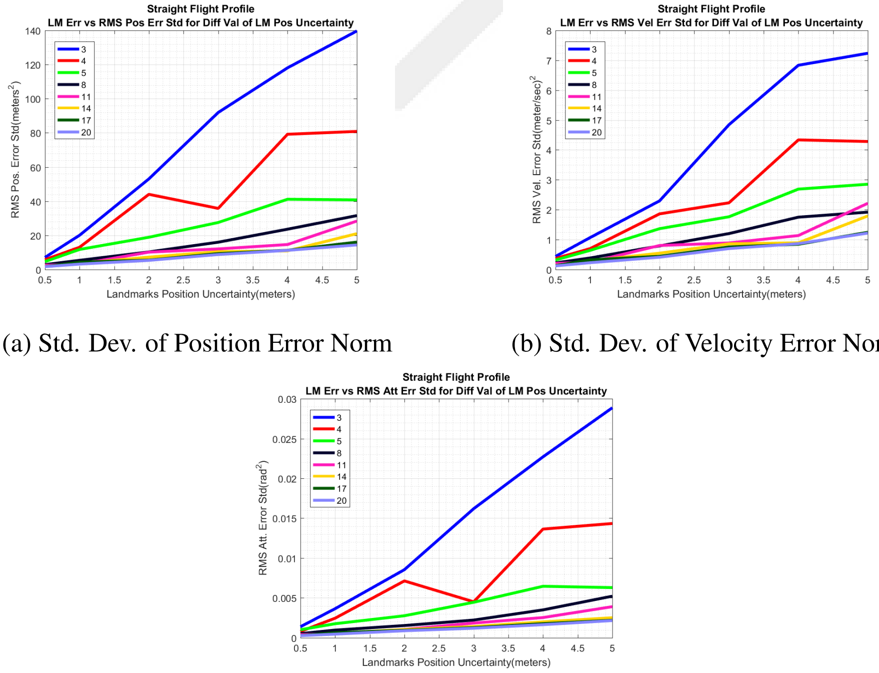


(a) RMS Position Error

(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.11: Straight Flight Profile: RMS Navigation Solution Errors versus Std. Dev. of Landmark Pos. Errors(x-axis) and Number of Landmarks(color traces)



(a) Std. Dev. of Position Error Norm

(b) Std. Dev. of Velocity Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.12: Straight Flight Profile: Std. Dev. of Navigation Solution Errors versus Std. Dev. of Landmark Pos. Errors(x-axis) and Num. of Landmarks(color traces)

Same conclusions are appeared to be valid for all flight profiles. Therefore RMS and standard deviation of errors at the end of mixture and ellipsis flight profiles will be presented in tabular form at Tables 5.7 - 5.12. All monte carlo runs have convergent error characteristic. Therefore stability results are not presented individually.

Table 5.7: Mixture Flight Profile: RMS and Std. Dev. of Position Error

Pos. Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	21.24043	9.7063	60.9462	32.1794	95.5557	47.7832
11	8.717538	4.2935	26.1794	10.7895	44.2156	21.4653
20	6.060835	2.7603	18.3662	8.64023	28.3750	12.3956

Table 5.8: Mixture Flight Profile: RMS and Std. Dev. of Velocity Error

Pos.Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	1.537985	0.73593	4.27804	2.25976	6.83874	3.43857
11	0.693274	0.32468	2.03675	0.87308	3.62635	1.81679
20	0.488725	0.21203	1.52227	0.69689	2.28525	0.99907

Table 5.9: Mixture Flight Profile: RMS and Std. Dev. of Attitude Error

Pos. Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	0.00420	0.00204	0.00977	0.00542	0.01525	0.00831
11	0.00178	0.00089	0.00521	0.00214	0.00817	0.00409
20	0.00123	0.00054	0.00379	0.00199	0.00565	0.00250

Table 5.10: Ellipsis Flight Profile: RMS and Std. Dev. of Position Error

Pos. Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	32.0159	21.0312	90.0498	52.2506	148.329	80.9260
11	8.8679	3.33418	32.1461	18.1507	47.7569	22.5724
20	6.8768	2.85356	21.3924	9.01141	34.5266	13.9994

Table 5.11: Ellipsis Flight Profile: RMS and Std. Dev. of Velocity Error

Pos. Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	1.7015	1.1646	4.6787	2.7027	7.5793	4.2698
11	0.4958	0.1940	1.5958	0.8598	2.4020	1.2058
20	0.3905	0.1798	1.0557	0.4724	1.7097	0.7147

Table 5.12: Ellipsis Flight Profile: RMS and Std. Dev. of Attitude Error

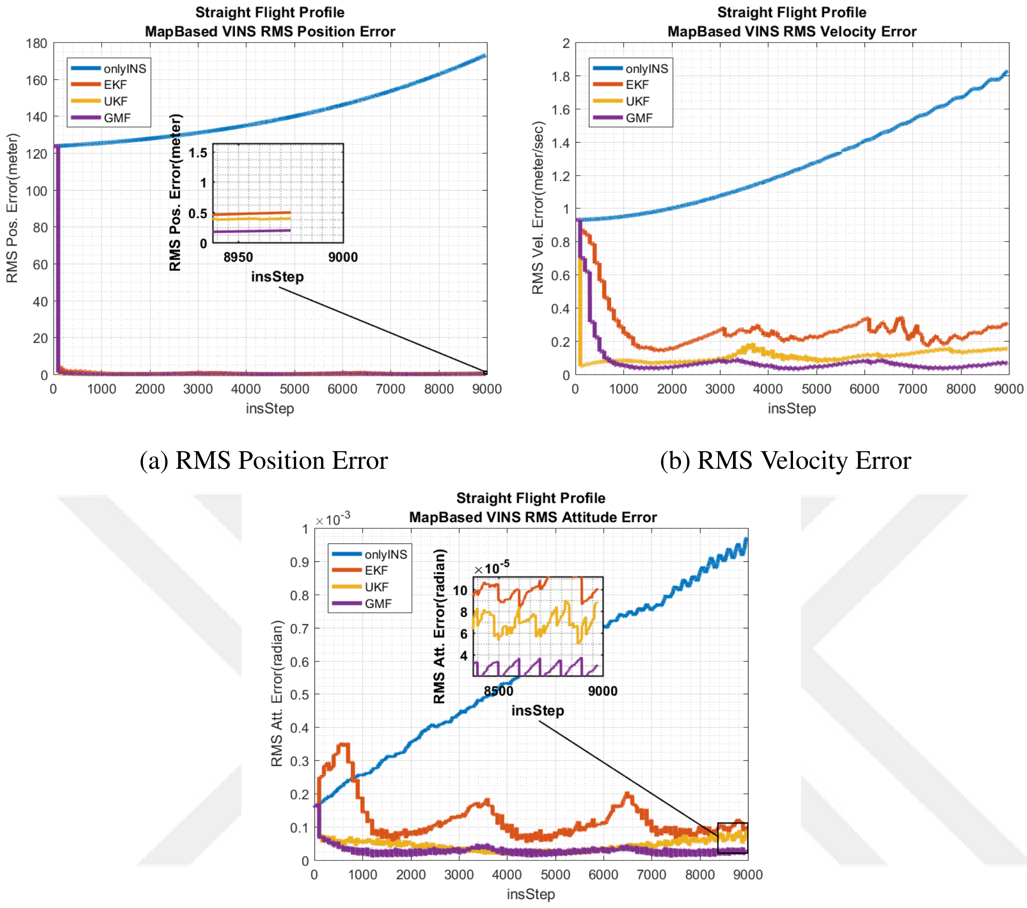
Pos. Unc.(m)	1		3		5	
Unit	rad	rad^2	m/s	$(m/s)^2$	m	m^2
Num. Landm.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
4	0.0047	0.0031	0.0139	0.0079	0.0244	0.0143
11	0.0014	0.0005	0.0049	0.0028	0.0074	0.0035
20	0.0010	0.0004	0.0032	0.0013	0.0052	0.0022

5.2.3 Comparative Evaluation of Sensor Fusion Filters on Map-Based VINS

The core framework that determines the stability and error of the fused information is sensor fusion filter. In chapter 4, detailed information about sensor fusion filters were presented.

Sensor fusion filters performs optimally when all assumptions of filter is satisfied. However both map-based and mapless VINS violates the assumptions of studied Bayiesian filters because of non-zero mean and non-Gaussian error sources. These assumptions were presented in 4. Some other factors such as linearizion and discretization also causes performance degradation in terms of stability and navigation solution error.

In order to study the effect of sensor fusion filter selection on Map-Based VINS, simulations are conducted. Results of these simulations are analyzed in term of RMS and standard deviation navigation solution error under different sensor fusion filters. In these simulations, a landmark map is assumed to be available before navigation operation. 12 landmarks with known positions are tracked during 30 frames. New landmarks with known position are initialized to filter after 30 frames. A landmark position error with 1 meter standard deviation is added to known positions of landmarks on each axes individually to increase the fidelity of simulation. 50 Monte Carlo runs for each filter are carried out.



(a) RMS Position Error

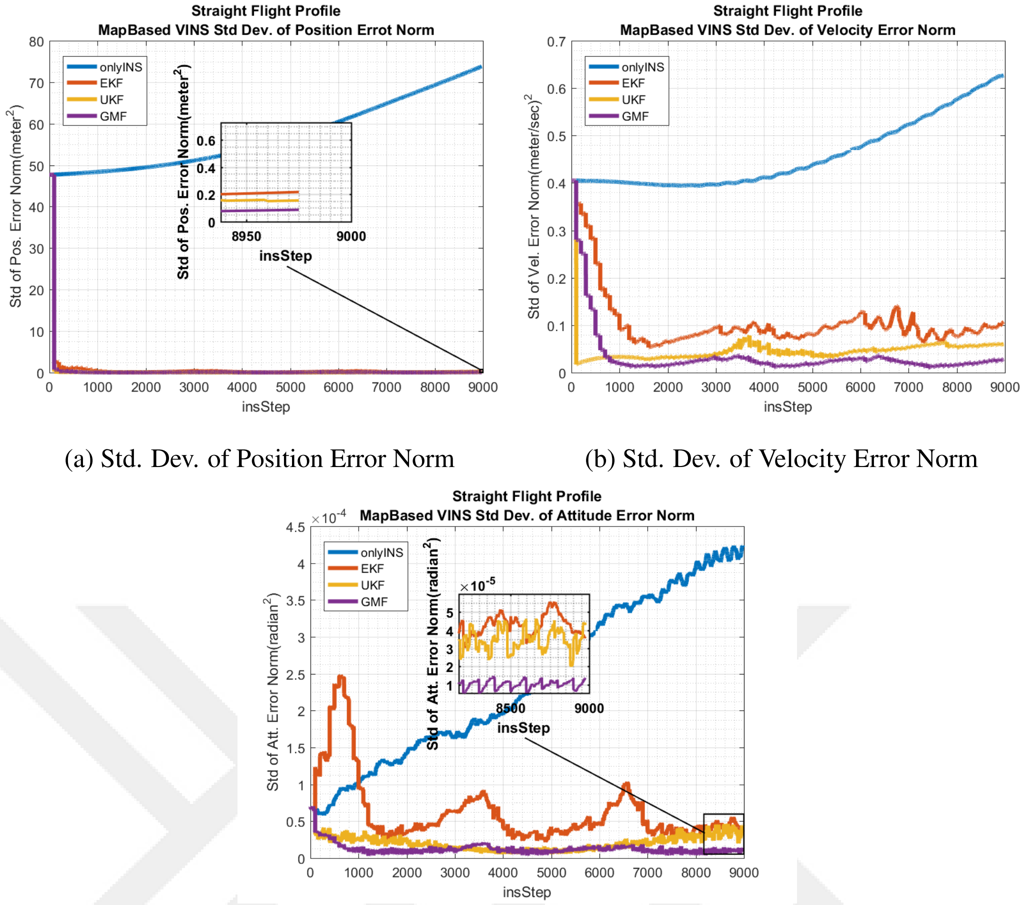
(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.13: Straight Flight Profile: RMS Navigation Solution Errors of Different Sensor Fusion Filters and only INS

Figure 5.13 shows the comparison of RMS navigation solution error under different sensor fusion filters and the only INS for straight flight profile. Results show that all aiding methods are able to refine the navigation solution and have better performance compared to only INS. Since map based VINS provides an external position information to UAV, it is possible to reduce the position, velocity and attitude errors.

The UKF and proposed GMF take the advantage of better approximation to state distribution and settles down to lower error level compared to EKF. The proposed GMF overperforms compared to other filters in terms of all navigation variables. This result is followed by UKF and EKF respectively.



(a) Std. Dev. of Position Error Norm

(b) Std. Dev. of Velocity Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.14: Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms of Different Sensor Fusion Filters and only INS

Figure 5.14 shows comparison of standard deviation of navigation solution error under EKF, UKF, GMF and only INS for straight flight profile. Results are compatible with RMS error. The proposed GMF method can also provide lower standard deviation compared to other filters. This result is followed by UKF and EKF.

Peak of error ripples corresponds to steps where new landmarks are initialized to filter. The positions of these landmarks are also refined during track sequence.

Observations are approximately valid for all flight profiles. Therefore only RMS and standard deviation of error levels are provided for mixture and ellipsis flight profiles at tables 5.14 and 5.15.

Table 5.13: Straight Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	9.97E-05	3.63E-05	0.308882	0.106726	0.48896	0.214605
UKF	8.76E-05	4.37E-05	0.15476	0.060401	0.389503	0.152762
GMF	2.99E-05	1.3E-05	0.072173	0.028361	0.192261	0.08401
INS	0.00097	0.000423	1.827929	0.626795	173.0205	73.78661

Table 5.14: Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	0.000111	4.57E-05	0.300653	0.126049	0.476419	0.21741
UKF	7.73E-05	3.54E-05	0.130628	0.054579	0.472584	0.195584
GMF	2.72E-05	1.15E-05	0.070553	0.029127	0.197727	0.097527
INS	0.00103	0.000419	1.704476	0.678559	143.5549	48.99185

Table 5.15: Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Errors of Different Sensor Fusion Filters and only INS

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	0.000135	6.26E-05	0.345452	0.136547	0.886044	0.45479
UKF	7.84E-05	3.1E-05	0.086232	0.033472	0.367314	0.197827
GMF	2.35E-05	1.16E-05	0.03991	0.018377	0.240945	0.122842
INS	0.000867	0.000345	1.679603	0.793392	180.4135	74.34071

All filters show 100 percent convergence. Therefore an extra table is not provided for stability results.

The proposed GMF and UKF propagates multiple components and represents instantaneous state density better compared to EKF. The proposed GMF takes the advantage of using more components and other improvements at initialization and update steps and it is able to performs better in term of all navigation variables at all flight profiles. But these filters increase the computational cost of the system.

Although faster systems can be implemented using more efficient software and hardware architectures, in order to provide an insight for the computational cost of each filter, the average INS step intervals are measured using MATLAB's timer functions. Results are presented in Table 5.16.

Table 5.16: The Average Time Intervals for Each INS Step of VINS

	UKF	EKF	GMF
Step Interval (ms)	28.63	0.29809	61.945119

Since 12 landmarks are used in simulations, while EKF propagates single component, UKF and GMF propagate 91 and 200 components respectively. Ratio between number of components approximately holds for the ratio between measured times of EKF, UKF and GMF in Table 5.16. Since extra operations like resampling and dynamic sigma point generation included in GMF and UKF, ratio of time intervals are slightly above the ratio of components.

5.3 Simulations on Mapless VINS

In order to study the sensor fusion filters and performance critical parameters of mapless VINS, simulations are conducted and results are presented in this section. In these simulations, landmarks set's positions and covariances are initialized using instantaneous navigation solution during flight in a standalone manner.

The performance of mapless VINS is determined by similar factors with the map-based VINS. These factors were given in Sec. 5.2. But in map-based VINS, initial position uncertainty of landmarks are independent from the mobile platform's navigation solution. However in mapless VINS, since landmarks are initialized with respect to mobile platform's navigation solution, it is strongly correlated with the position uncertainty of landmarks as explained in Sec. 4.1.2. These causes difference in performance of the map-based and mapless VINS.

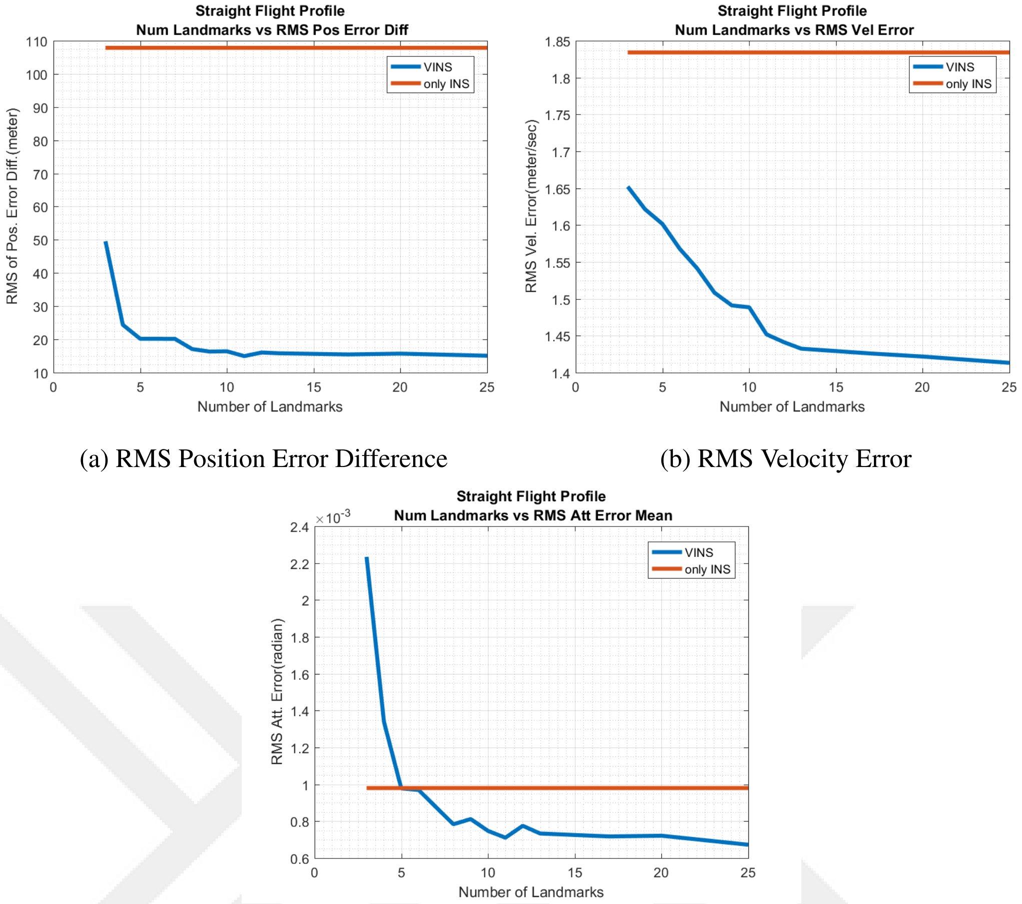
Since mapless VINS cannot provide an external and independent global position information to UAV, it is not possible to reduce the position error of the mobile platform in mapless-VINS but position error divergence can be damped. Another words, position error difference of initial and final position error where flight path is completed should be low. Therefore for mapless VINS position error difference will be evaluated as the performance criteria.

5.3.1 Effect of Number of Landmarks on Mapless VINS Performance

In mapless VINS, positions of landmarks are calculated with respect to navigation solution of the platform and don't have to be loaded to mobile platform before flight. Therefore a pre-planning phase is not required in mapless VINS. This makes mapless VINS favorable concept under many circumstances. However not providing an external information comes with the cost of decrease on accuracy/stability of navigation solution and increase on landmark position uncertainty. Errors on landmarks positions are correlated with; slant range between landmark and the mobile platform, errors on mobile platform's navigation solution and measurements. Since every landmark is initialized once, landmark position error have non-zero and non-Gaussian distribution. This violates the assumptions of Bayesian filters used in this study, it is one of the reasons causing suboptimal performance in sensor fusion of mapless VINS.

As explained in Sec. 5.2.2, it is possible to compensate the effect of position uncertainty by applying SLAM and using higher number of landmarks in sensor fusion filter. Since every landmark also increases the computational cost because of increasing dimensions of filter, a limit must be introduced to number of landmarks for implementing a real time system.

In order to investigate the effect of number of landmarks used in sensor fusion filter of mapless VINS, simulation are conducted. 50 monte carlo runs for each number of landmark value are carried out. RMS and standard deviation of navigation solution error are analyzed for different flight profiles.



(a) RMS Position Error Difference

(b) RMS Velocity Error

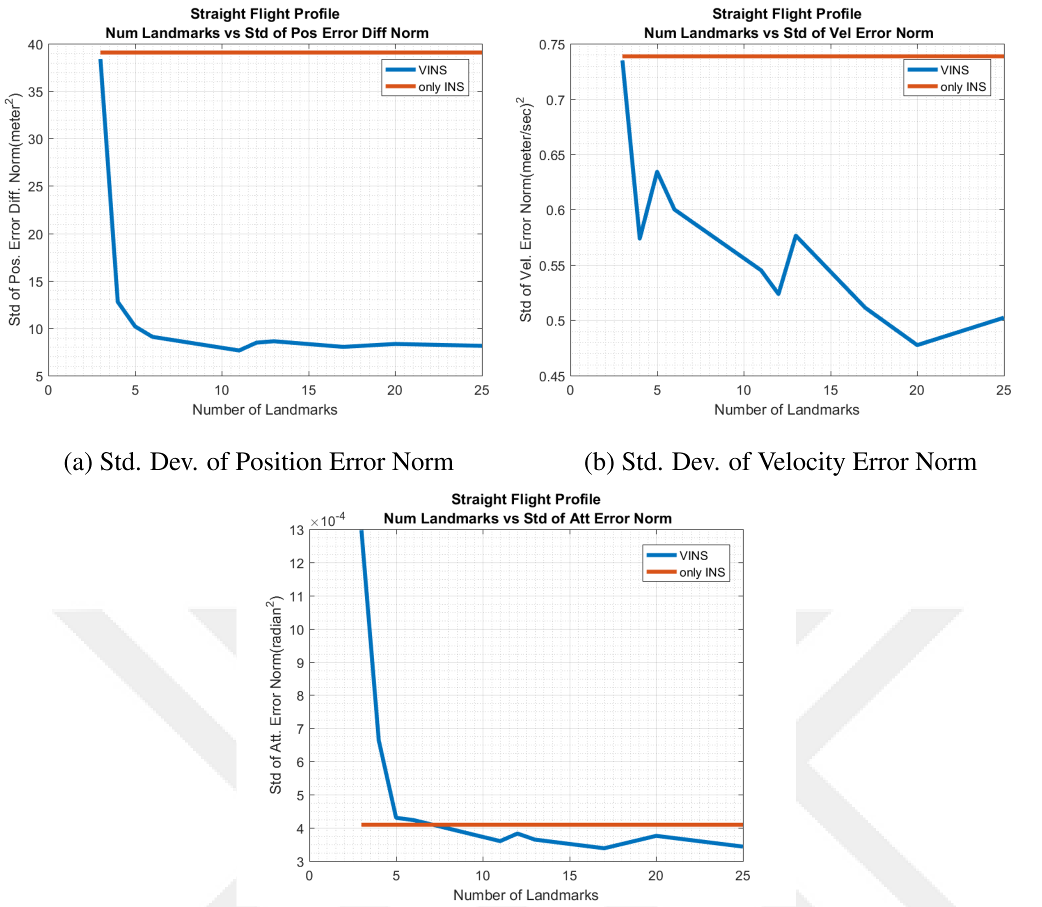
(c) RMS Attitude Error

Figure 5.15: Straight Flight Profile: RMS Navigation Solution Error versus Number of Landmarks

Fig.5.15 shows the norm of navigation solution error mean under different number of landmarks. Red constant line shows the RMS error of only INS .

Results show that number of landmarks must be held above a number to perform better than only INS in terms of attitude error. Refinements of attitude error is observed to be successful after using 8 landmarks.

RMS position error performance is saturated after a certain number of landmarks used in filter. However RMS velocity and attitude error continues to decrease with the increase on number of landmarks.



(a) Std. Dev. of Position Error Norm

(b) Std. Dev. of Velocity Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.16: Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms versus Number of Landmarks

Similar results are observed for the standard deviation of navigation solution error given in Fig. 5.16. Advantage of using higher number of landmarks can be observed better in terms of standard deviation of velocity and attitude errors. Similar to RMS error characteristic, standard deviation of attitude error can be decreased below the only INS if more than 8 landmarks are used in filter.

Characteristics of RMS and standard deviation of navigation solution error under increase on number of landmarks is similar for all flight profiles. Simulations are conducted for 14 different number of landmarks values. It is observed that minimum, intermediate and maximum of simulated values are enough to represent the results. Therefore only RMS and standard deviation of error levels are provided for mixture and ellipsis flight profiles at Tables 5.17 and 5.18.

Table 5.17: Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Landmarks

	Attitude Error		Velocity Error		Position Error	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
Num LM.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
3	0.00226	0.001458	3.941911	3.221569	40.98149	27.22951
12	0.001058	0.000514	1.293103	0.657861	23.86867	12.07363
25	0.000817	0.000353	1.229557	0.612692	16.77811	7.416489

Table 5.18: Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Landmarks

	Attitude Error		Velocity Error		Position Error	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
Num LM.	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
3	0.00129	0.000721	1.493344	0.537917	31.98809	20.37654
12	0.000614	0.000312	1.363193	0.458553	13.39592	6.334202
25	0.000307	0.000228	1.303281	0.164384	11.61019	5.947553

Table 5.19 shows the convergence percentages of all paths under different number of landmarks. Although simulations are conducted for 14 different values, because of space considerations stability of 10 selected values are presented in tabular form. Results show that in order to have a stable EKF based mapless VINS under all flight scenarios, more than 8 landmarks are required.

Table 5.19: Convergence Percentages under Different Number of Landmarks for All Flight Profiles

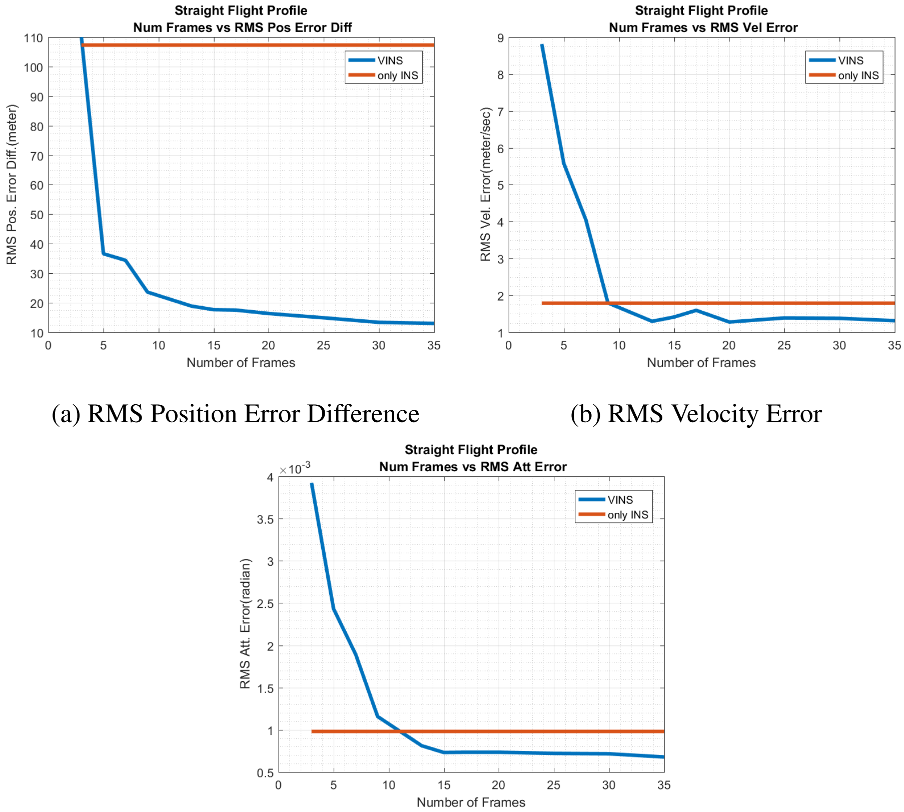
Num. of Landm	3	4	5	7	8	9	10	15	20	25
Straight Flight Prof.	64	90	98	100	98	100	100	100	100	100
Mixture Flight Prof.	60	84	90	98	96	94	92	96	94	96
Ellipsis Flight Prof.	62	74	84	88	92	96	98	90	92	98

5.3.2 Effect of Number of Tracked Frames on Mapless VINS Performance

Visual Aided Inertial Navigation Systems jointly refine the navigation solution and positions of landmarks if SLAM approach is applied. SLAM integrates the positions of landmarks into sensor fusion filter by including landmark positions into state vectors and landmark position covariance matrix into state covariance matrix. As long as system is observable, landmark positions are refined at every frame if landmarks are visible and successfully matched to their pixel coordinates in previous frame. Therefore number of frames that landmarks are tracked is crucial for getting an update from a landmark with higher position accuracy. Since landmark positions in mapless VINS are calculated with respect to mobile platform's navigation solution, their position uncertainty is generally greater than landmarks in map-based VINS. Hence refining landmark positions is crucial in mapless VINS.

Number of tracked frames also determines many parameters of the system related with mobile platform's mechanical design, electronics hardware and flight speed, flight altitude etc. Hence an optimum number of frames must determined at designing phase of VINS.

In order to investigate the effect of number of frames that landmarks are tracked on the performance of sensor fusion filter of mapless VINS, simulation are conducted. 50 monte carlo runs for each number of frame are carried out. RMS and standard deviation of navigation solution error are analyzed for different flight profiles.



(a) RMS Position Error Difference

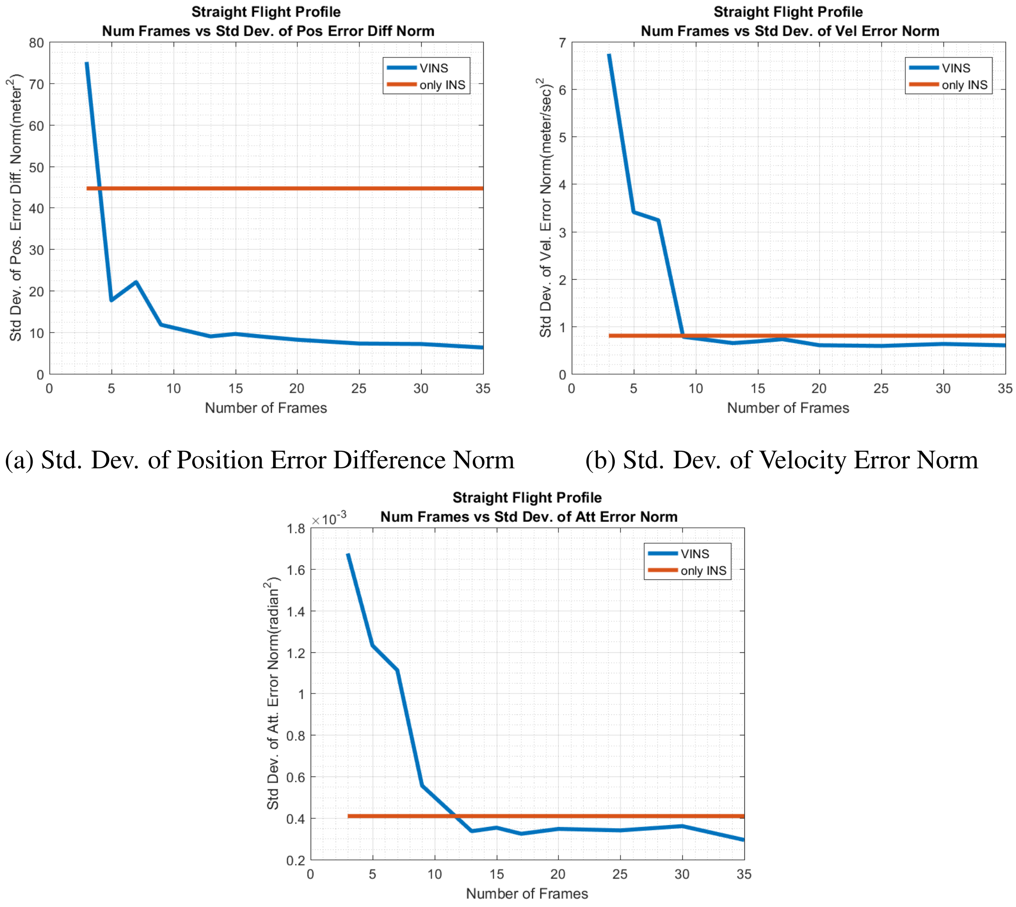
(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.17: Straight Flight Profile: RMS Navigation Solution Error versus Number of Tracked Frames

Fig. 5.17 shows the norm of navigation solution error mean under different number of frames that landmarks are tracked for straight flight profile. Red constant line in each figure shows the only INS error. Results show that RMS error can only be decreased below only INS error after a certain number of tracked frames which is determined to be 4 for position, 10 for velocity and 12 for attitude error on our scenario. After specified number of tracked frames, performance is observed as almost saturated for attitude and velocity error but RMS position error continuous to decrease.

Fig. 5.18 shows the standard deviation of navigation solution error norm under different number of tracked frames for straight flight profile. Specified numbers of tracked frames required for refining RMS navigation solution error in our scenarios are appeared to be same for standard deviation of navigation solution error.



(a) Std. Dev. of Position Error Difference Norm

(b) Std. Dev. of Velocity Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.18: Straight Flight Profile: Std. Dev. of Navigation Solution Error versus Number of Tracked Frames

Characteristics of RMS and standard deviation of navigation solution error under increase on number of tracked is observed to be similar for all flight profiles. Therefore only RMS and standard deviation of navigation solution error at the end of mixture and ellipsis flight profiles are presented at Tables 5.20 and 5.21.

Table 5.20: Mixture Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Tracked Frames

	Attitude Error		Velocity Error		Position Error	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
Num Frames	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
3	0.0026	0.0008	2.4473	1.1123	36.8551	8.4353
17	0.0009	0.0004	1.3361	0.5606	17.71747	8.1298
35	0.0007	0.0003	1.2305	0.4845	13.75221	7.1161

Table 5.21: Ellipsis Flight Profile: RMS and Std. Dev. of Navigation Solution Error versus Number of Tracked Frames

	Attitude Error		Velocity Error		Position Error	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
Num Frames	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
3	0.00682	0.002303	3.2570	1.5417	61.6663	30.7052
17	0.00013	8.35E-05	0.5719	0.5017	25.0770	16.4692
35	9.3E-05	5.14E-05	0.3364	0.1808	9.2469	4.8240

Table 5.22 shows the convergence percentages of all flight profiles under different number of landmarks. Results show that in order to have a stable EKF based mapless VINS under all flight scenarios, landmarks are required to be tracked more than 12 frames.

Table 5.22: Convergence Percentages under Different Number of Frames for All Flight Profiles

Num. of Fr.	3	5	7	9	13	15	17	20	25	30	35
Straight Flight Prof.	4	22	56	92	92	96	96	100	100	100	100
Mixture Flight Prof.	40	82	82	80	85	90	96	100	100	96	100
Ellipsis Flight Prof.	22	32	16	28	78	76	80	92	96	100	100

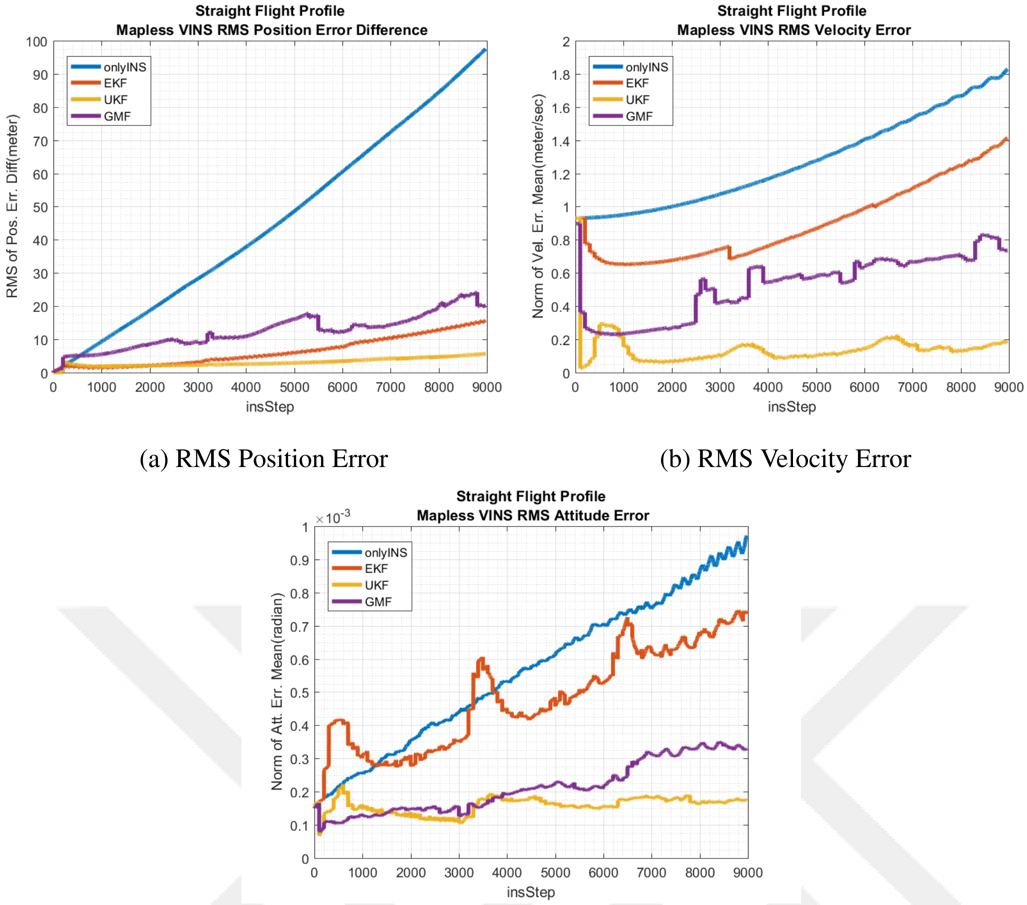
5.3.3 Comparative Evaluation of Sensor Fusion Filters on Mapless VINS

The EKF, UKF and proposed GMF follows different strategies to represent the instantaneous state density as explained before. In mapless VINS, performance is strongly related with the incorporation of landmark positions into state vector. This incorporation step differs for every Bayesian filter used in this study. For example, EKF follows the basic strategy since it only consist of single component. However UKF and GMF propagates and benefit from multiple components and follows different strategies to initialize the landmark positions and their uncertainties.

Since UKF calculates the sigma points at every INS step, landmark positions and covariances must be initialized according to mean state and mean covariance. Otherwise it is meaningless to initialize landmarks according to temporary components (sigma points) since these components will be lost in the upcoming steps. Initialized landmark positions are diversified at every propagation step with the dynamically generated sigma points according to landmark position covariances.

Components of GMF are not dynamically generated and they carry different hypotheses for navigation solution. Since landmarks positions are calculated with respect to navigation solution of the mobile platform, all components have different hypotheses about initial landmark positions and their uncertainties. If landmark positions are calculated with respect to only mean navigation solution and mean covariances, all components except mean component would have high normalized innovation square(NIS) values, which is a measure of difference between expected measurement and actual measurement, and eventually these components would stop getting updates from measurement because of NIS threshold.

Differences in SLAM approach is crucial for mapless VINS because of landmark positions are unknown. These filters makes also different assumption as explained in Chapter 4. In order to study these differences, sensor fusion filters are comparatively studied in this section. Results of these simulations are analyzed in term of RMS and standard deviation navigation solution errors using 50 Monte Carlo runs for each filter. In this simulations, a landmark map is assumed to be not available before navigation operation. 12 landmarks are tracked during 30 frames and their positions are calculated during flight. New landmarks are initialized to filter after 30 frames.



(a) RMS Position Error

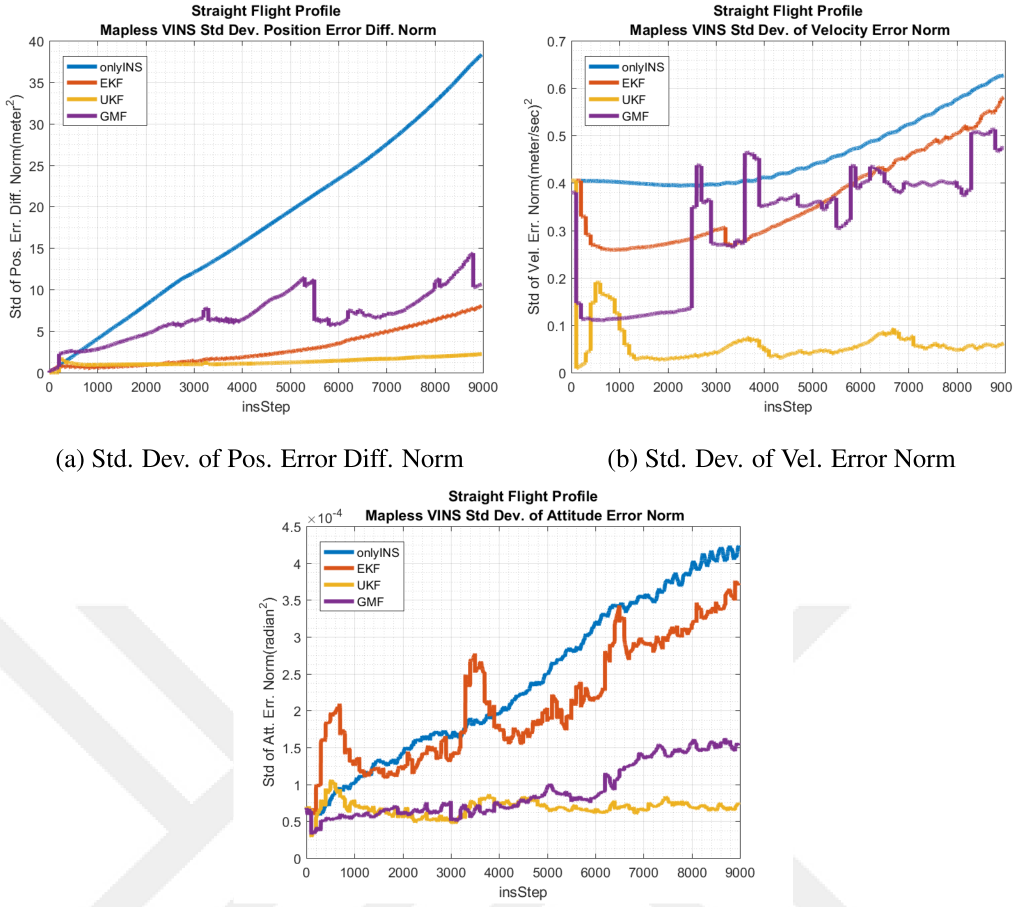
(b) RMS Velocity Error

(c) RMS Attitude Error

Figure 5.19: Straight Flight Profile: RMS Navigation Solution Error under Different Sensor Fusion Filters and Only INS

Fig. 5.19 and 5.20 shows comparison of different sensor fusion filters in terms of RMS and standard deviation of navigation solution error respectively. Results show that implemented filters are successful at aiding INS.

Nonlinearity in Mapless VINS problem is higher compared to map-based VINS. Therefore UKF shows better results compared to other filters in terms of all navigation variables. UKF and GMF are observed to less susceptible to landmark initialization compared to EKF by taking the advantage of using more components and making less assumptions about state density.



(a) Std. Dev. of Pos. Error Diff. Norm

(b) Std. Dev. of Vel. Error Norm

(c) Std. Dev. of Attitude Error Norm

Figure 5.20: Straight Flight Profile: Std. Dev. of Navigation Solution Error Norms under Different Sensor Fusion Filters and Only INS

Unlike in map-based VINS, the advantages of proposed GMF cannot be observed in mapless VINS. This results from the fact that there is not an optimal way of weighting different components and solving other problems such as sample degeneracy and component initialization in GMF. However GMF still performs betters than EKF for velocity and attitude errors and it can show best performance in terms of certain navigation variables in other flight profiles as shown in at tables 5.24 and 5.25.

Because of the error sources which violate the assumption of UKF, error levels of UKF slightly increases with INS steps. But these error levels still remain noticeably lower compared to other filters.

Table 5.23: Straight Flight Profile: RMS and Std of Navigation Solution Error

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	0.000743	0.000372	1.412809	0.580519	15.67982	8.053901
UKF	0.000177	7.39E-05	0.188418	0.061531	5.670387	2.219307
GMF	0.000329	0.000155	0.736536	0.473376	20.23883	10.62102
INS	0.00097	0.000423	1.827929	0.626795	97.45366	38.31953

Table 5.24: Mixture Flight Profile: RMS and Std of Navigation Solution Error

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	0.000727	0.00035	1.84306	0.70051	16.03049	7.535479
UKF	0.00014	5.65E-05	0.172759	0.071476	4.585546	2.23276
GMF	0.000311	0.000149	0.521651	0.293783	19.6097	10.54954
INS	0.00103	0.000419	1.704476	0.678559	84.34644	34.57703

Table 5.25: Ellipsis Flight Profile: RMS and Std of Navigation Solution Error

	Att. Err.		Vel. Err.		Pos. Err.	
Unit	rad	rad^2	m/s	m/s^2	m	m^2
	μ_e	σ_e^2	μ_e	σ_e^2	μ_e	σ_e^2
EKF	0.000472	0.000262	0.967186	0.522699	10.21357	4.951266
UKF	0.000393	0.000183	0.262077	0.134253	4.154502	1.836126
GMF	0.000236	8.01E-05	0.58151	0.451823	19.64156	10.97029
INS	0.000867	0.000345	1.679603	0.793392	115.4948	53.27032

While calculating RMS and standard deviation of navigation solution error, only Monte Carlo runs satisfying the stability conditions explained in Appendix A are taken into account. Stability of compared sensor fusion filters are given in Table 5.26.

Table 5.26: Stability Percentages of Different Sensor Fusion Filters on Mapless VINS

Stability (%)	UKF	EKF	GMF
Straight Flight Profile	100	100	84
Mixture Flight Profile	100	94	78
Ellipsis Flight Profile	100	90	70

UKF shows superior stability performance compared to other filters. This results also verifies the literature. Unfortunately, the proposed GMF is observed to be less stable because of the reasons explained above.

Although faster systems can be implemented using efficient software and hardware architectures, in order to provide an insight for the computational cost of each filter, the average step intervals are measured using MATLAB's timer functions. Results are presented in Table 5.27.

Table 5.27: The Average Time Intervals for Each Step of VINS

	UKF	EKF	GMF
Step Interval (ms)	31.54874	0.29809	65.57381615

Computational cost is higher compared to map-based VINS, since landmark positions and their covariances are also calculated during flight.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

In this thesis, we investigated the effect of performance critical parameters and sensor fusion filters on the stability and navigation solution error of VINS. Study was conducted on a simulation environment which was composed of simulated IMU, camera, camera gimbal, INS only navigation sub-system and sensor fusion filters. Monte Carlo simulations were run in order to characterize the VINS performance under the tested parameters and sensor fusion filters. Results of Monte Carlo runs were evaluated in terms of RMS and standard deviation of navigation solution error and stability. All simulations were conducted for three different flight profiles.

The problem was investigated in two different domains. First domain was the map-based VINS where a landmark set with known positions is available before flight. Positions of landmarks carry uncertainties because of visual position-fixing and measurement errors. In this study, statistics of the landmarks position uncertainties were assumed to be available before flight and SLAM approach was used for refining the position of landmarks in map-based VINS. In order to study the effect of performance critical parameters of map-based VINS, simulations were conducted with EKF. The effect of measurement error was studied under selected flight and imaging geometry. RMS and standard deviation of navigation solution error and their oscillations were determined to be increased under the increase of pixel measurement error. The effect of the number of landmarks and landmark position uncertainty were jointly studied and it was found out that increasing number of landmarks used in the filter can compensate and linearize the increase in navigation solution error caused by high landmark position uncertainty. All simulations which were conducted with EKF have shown convergent error characteristics in map-based VINS.

The second domain was mapless-VINS where a landmark set is not available before flight and landmark positions and the uncertainties are calculated with the help of instantaneous navigation solution of the mobile platform during flight. Position error in mapless VINS cannot be decreased due to unavailability of independent global position information. However attitude and velocity errors can be refined. Since landmark positions are initialized with respect to navigation solution, their initial position uncertainty is higher compared to landmarks of map-based VINS. Hence, the number of frames where the same landmarks are refined carries an important role in mapless VINS. Number of frames also determines many important electronic and mechanical parameters while designing a VINS. It is shown that a certain number of frames must be exceeded for performing better than only INS. It was shown that after a certain number of tracked frames, performance is saturated for our scenarios. The effect of the number of landmarks used in sensor fusion filter was also analyzed for mapless VINS. Increasing number of the landmarks especially refined the velocity and the attitude of the mobile platform in our simulations.

After revealing the need of exceeding a number of tracked frames for maximizing the performance of VINS, we proposed a feature region selection algorithm that contributes maximizing number of frames which landmarks are tracked. Since some feature points will become out of sight due to imaging geometry during flight, these landmarks should not be tracked. We built on these ideas and proposed a feature region selection algorithm which uses gimbal angles and camera parameters for selecting the features that will remain visible during track.

A Gaussian Mixture Filter was proposed. Proposed filter make use of unscented transform and other components sampled around mean according to initial covariances for initialization of components and applies resampling scheme when effective number of components falls below a certain level. Unlike UKF, it also initializes landmarks individually for all components. In order to characterize the performance of proposed GMF, UKF and EKF on mapbased and mapless VINS, simulations were conducted. For both types of VINS, all filters are able to refine the navigation solution compared to only INS. UKF and GMF perform better in terms of the RMS and standard deviation of navigation solution error compared to EKF. GMF overperforms for all navigation variables and flight profiles in map-based VINS. However looses its advantage in mapless VINS. This result originates from the fact that there is not an optimal way of weighting and resampling the Guassian components.

As a future direction of this study, real flight tests can be conducted. During real flight tests, synchronized visual data, range information and IMU data can be collected and processed in offline manner in order to verify the theoretical outcomes of this study.

In this study, range information was assumed to be available for all landmarks using a LiDAR sensor. An interesting study would be comparison of different range sensors and range estimation methods under different sensor fusion algorithms. With such a study, the range estimation accuracy needed for having a stable VINS can be determined.

Performance of the GMF can be improved by propagating weighted sum of UKFs which can further diversify the state hypotheses of GMF. A drawback of UKF based GMF would be increase on computational cost which makes real time implementation not realizable. This problem can be solved by applying a marginalization method like Rao-Blackwellization which decreases the number of components by applying conditional independence assumption between navigation states and positions of landmarks. Considering these potential improvements, an extension to this study can be increasing the stability and the accuracy of the navigation solution on GMF based mapless VINS by applying more refinements on the filtering algorithm and finding an appropriate component weighting method.

As a further analysis, a loosely coupled VINS, which calculates the rotation and the translation of the mobile platform explicitly from measurements, can be implemented and compared with the results of this study.

REFERENCES

[1] D. Alspach and H. Sorenson. Nonlinear bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.

[2] J. K. Bingham. Vision-aided, cooperative navigation for multiple unmanned vehicles. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT, 2009.

[3] H. E. Bingöl. Bearings-only tracking. Master's thesis, METU, 2011.

[4] J. I. Bowditch. *American practical navigator*, page 799. Defense Mapping Agency Hydrographic Center, 2003.

[5] J. R. Carr and J. S. Sobek. Digital scene matching area correlator (DSMAC). *Image Processing For Missile Guidance*, 1980.

[6] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein. GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. *Journal of Field Robotics*, 30(3):415–438, 2013.

[7] C.-C. Chu, F. A. P. Lie, L. Lemay, and D. Gebre-Egziabher. Performance comparison of tight and loose ins-camera integration. In *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011)*, page 3516, 2001.

[8] G. Conte and P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal on Advances in Signal Processing*, 2009:10, 2009.

[9] D. Dardari, E. Falletti, and M. Luise. *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2012.

[10] M. U. de Haag, D. Venable, M. Smearcheck, J. Campbell, and M. Miller. Flash-ladar inertial navigator aiding. In *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, pages 677–683. IEEE, 2006.

[11] M. U. de Haag, D. Venable, and A. Soloviev. Implementation of a flash-ladar aided inertial navigator. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 560–567. IEEE, 2008.

[12] S. Ebcin. Tightly integrating optical and inertial sensors for navigation using the ukf. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT, 2008.

[13] G. Falco, M. Pini, and G. Marucco. Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios. *Sensors*, 17(2):255, 2017.

[14] E. E. K. Filter. Vision-aided navigation for small UAVs in GPS-challenged environments. 2007.

[15] M. G. Giebner. Tightly-coupled image-aided inertial navigation system via a kalman filter. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT, 2003.

[16] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[17] P. D. Groves. Principles of GNSS. *Inertial, and Multisensor Integrated Navigation Systems*, 521, 2008.

[18] P. D. Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[19] F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.

[20] F. Gustafsson and G. Hendeby. Some relations between extended and unscented Kalman filters. *IEEE Transactions on Signal Processing*, 60(2):545–555, 2012.

[21] M. Hazewinkel. *Encyclopaedia of Mathematics: Volume 6: Subject Index—Author Index*. Springer Science & Business Media, 2013.

[22] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*, pages 559–574. Springer, 2013.

[23] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 33(1):182–201, 2014.

[24] J. P. Hespanha. *Linear systems theory*. Princeton university press, 2009.

[25] N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear algebra and its applications*, 103:103–118, 1988.

[26] J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.

[27] W. Jiang, L. Wang, X. Niu, Q. Zhang, H. Zhang, M. Tang, and X. Hu. High-precision image aided inertial navigation with known features: Observability analysis and performance evaluation. *Sensors*, 14(10):19371–19401, 2014.

[28] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[29] Y. Kim, D. Lee, and H. Bang. Vision-only UAV navigation aided by terrain elevation map. In *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, pages 1729–1733. IEEE, 2012.

[30] X. Kong, W. Wu, L. Zhang, and Y. Wang. Tightly-coupled stereo visual-inertial navigation using point and line features. *Sensors*, 15(6):12816–12833, 2015.

[31] K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *Robotics research*, pages 201–212. Springer, 2010.

[32] D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Experimental Robotics*, pages 303–317. Springer, 2013.

[33] P. Kozierski, M. Lis, and J. Ziętkiewicz. Resampling in particle filtering-comparison. 2013.

[34] N. Kwok, Q. P. Ha, S. Huang, G. Dissanayake, and G. Fang. Mobile robot localization and mapping using a gaussian sum filter. *International Journal of Control, Automation, and Systems*, 2007.

[35] K. Lee, J. M. Kriesel, and N. Gat. Autonomous airborne video-aided navigation. *Navigation*, 57(3):163–173, 2010.

[36] F. L. Lewis, L. Xie, and D. Popa. *Optimal and robust estimation: with an introduction to stochastic control theory*, volume 29. CRC press, 2007.

[37] M. Murata, H. Nagano, and K. Kashino. Gaussian sum resampling filter. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 4338–4343. IEEE, 2015.

[38] C. Musso, N. Oudjane, and F. Le Gland. Improving regularised particle filters. In *Sequential Monte Carlo methods in practice*, pages 247–271. Springer, 2001.

[39] R. Niu, P. K. Varshney, M. Alford, A. Bubalo, E. Jones, and M. Scalzo. Curvature nonlinearity measure and filter divergence detector for nonlinear tracking problems. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[40] M. L. Psiaki. Gaussian mixture nonlinear filtering with resampling for mixand narrowing. *IEEE Transactions on Signal Processing*, 64(21):5499–5512, 2016.

[41] S. Ren and W. Chang. A fast and robust scene matching method for navigation. *Progress In Electromagnetics Research M*, 36:57–66, 2014.

[42] B. Ristic, S. Arulampalam, and N. Gordon. Beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 19(7):37–38, 2004.

[43] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4326–4333. IEEE, 2002.

[44] M. Scalzo, G. Horvath, E. Jones, A. Bubalo, M. Alford, R. Niu, and P. K. Varshney. Adaptive filtering for single target tracking. In *SPIE Defense, Security, and Sensing*, pages 73360C–73360C. International Society for Optics and Photonics, 2009.

[45] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011.

[46] N. Shephard. Learning and filtering via simulation: smoothly jittered particle filters. 2009.

[47] S. Sirtkaya, B. Seymen, and A. A. Alatan. Loosely coupled kalman filtering for fusion of visual odometry and inertial navigation. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 219–226. IEEE, 2013.

[48] J. Soeder and J. Raquet. Image-aided navigation using cooperative binocular stereopsis. *Navigation*, 62(3):239–248, 2015.

[49] A. Soloviev, D. Bates, and F. GRAAS. Tight coupling of laser scanner and inertial measurements for a fully autonomous relative navigation solution. *Navigation*, 54(3):189–205, 2007.

[50] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz. A new approach to vision-aided inertial navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4161–4168. IEEE, 2010.

[51] D. Titterton and J. L. Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.

[52] M. J. Veth. Fusion of imaging and inertial sensors for navigation. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT, 2006.

[53] E. Wan and R. Van Der Merwe. Chapter 7: The unscented kalman filter. *Kalman Filtering and Neural Networks*, pages 221–280, 2001.

[54] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[55] A. Wu, E. N. Johnson, M. Kaess, F. Dellaert, and G. Chowdhary. Autonomous flight in GPS-denied environments using monocular vision and inertial sensors. *J. Aerospace Inf. Sys.*, 10(4):172–186, 2013.

[56] C. Xu, L. Zhang, L. Cheng, and R. Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1209–1222, 2017.

[57] D. J. Yates. Monocular vision localization using a gimbaled laser range sensor. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING, 2010.

[58] S. Zhao, F. Lin, K. Peng, B. M. Chen, and T. H. Lee. Homography-based vision-aided inertial navigation of UAVs in unknown environments. In *AIAA Guidance, Navigation, and Control Conference*, page 5033, 2012.

APPENDIX A

Chi-Square Goodness of Fit Test

Pearson's Chi Square test investigates how well the hypothesis that states value or distribution of observation fits the expected distribution of observation. It categorizes this hypothesis as true or false. Chi Square test can be used when tested variable follows Chi-Square distribution.

Chi Square distribution is the distribution of sum of squares of k independent normal random variables. [21]. If $Y_1, ..., Y_k$ are independent, standard normal random variables, then the sum of their squares:

$$Q = \sum_{i=1}^{k} Y_i^2 \tag{A.1}$$

is distributed according to the Chi-Squared distribution with k-degrees of freedom and denoted as: $Q \sim \chi^2(k)$ or $Q \sim \chi_k^2$. Probability density function of Chi Square distribution has an essential role in goodness of fit tests and it is calculated as:

$$f(x;k) = \begin{cases} \frac{x^{k/2-1}e^{-x/2}}{2^{k/2}\Gamma(k/2)}, & x > 0 \\ 0, & otherwise \end{cases} \tag{A.2}$$

Where $\Gamma()$ denotes the Gamma Function. Chi Square probability density function is given in Figure A.1

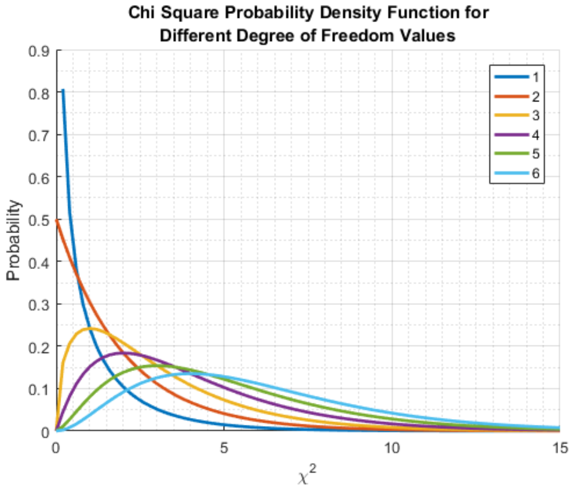


Figure A.1: PDF of Chi Square Distribution

A test statistics is a measure used in statistical hypothesis testing. It is believed to have Chi Square distribution in Pearson's Chi Square test. Chi Square goodness of fit test is conducted simply by comparing the test statistics with the value of Chi-Square density with k degree of freedom at selected value of level of confidence. Level of confidence determines the threshold for the probability of test statistics. If test statistics probability is lower than level of confidence, hypothesis is classified as wrong. It is typically selected as 0.05. Degree of freedom is the number of independent dimensions at the observation that specifies the state of the system. In VINS, dimension of measured pixel coordinates determines the degree of freedom which is two times number of tracked landmarks.

Violating assumptions of a sensor fusion filter can yield divergence of the state errors. Monte Carlo runs where divergence is detected should be excluded while calculating mean and standard deviation of state errors in order to characterize the system correctly. Converge percentage of Monte Carlo runs should be presented separately.

In [39] ,Niu et. Al. proposed that Normalized Innovation Square(NIS) of Kalman Filter based sensor fusion methods follows Chi Square distribution. Normalized innovation squared is defined as:

$$NIS_k = z_k^T S_k^{-1} z_k \tag{A.3}$$

Here z_k is the innovation another words difference between measurement and prediction and S_k is the innovation covariance matrix.In [44], Niu et. al. proposed that applying Chi Square test to every individual update step can mislead the divergence check because of the temporal error peaks. Hence applying an averaging filter on NIS values was proposed. Degree of freedom is calculated as the multiplication of output dimension and averaging window size of the filter.

It was verified that using divergence checking method proposed in [44] provides suitable divergence check results. p-value was selected as suggested value 0.05 and window size is selected as 3.

APPENDIX B

Computing Closest Symmetric and Positive Semidefinite Matrix

Theorem: *Every covariance matrix is a symmetric positive semi-definite matrix.*

Proof: Let the vector set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k\}$ is composed of random vectors where $\boldsymbol{x}_i \,\epsilon\, \boldsymbol{R}^n$. Mean vector $m_{\boldsymbol{x}}$ is calculated as:

$$m_{\boldsymbol{x}} = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{x}_i \tag{B.1}$$

The covariance matrix $\{\boldsymbol{C} \,\epsilon\, \boldsymbol{R}^n \times \boldsymbol{R}^n\}$ of vector set $\boldsymbol{X}$ is defined as:

$$\boldsymbol{C} = \frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T \tag{B.2}$$

Definition: *$\{\boldsymbol{A} \,\epsilon\, \boldsymbol{R}^n \times \boldsymbol{R}^n\}$ is called positive semi-definite if $\boldsymbol{uAu}^T \geq 0$ for $\{\forall\, \boldsymbol{u} \,\epsilon\, \boldsymbol{R}^n\}$*

$$\begin{aligned}\boldsymbol{u}^T\boldsymbol{C}\boldsymbol{u} &= \boldsymbol{u}^T(\frac{1}{k}\sum_{i=1}^{k}(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T)\boldsymbol{u}\\ &= \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{u}^T(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T\boldsymbol{u}\\ &= \frac{1}{k}\sum_{i=1}^{k}((\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T\boldsymbol{u})^T((\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T\boldsymbol{u})\\ &= \frac{1}{k}\sum_{i=1}^{k}((\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})^T\boldsymbol{u})^2 \geq 0\end{aligned} \tag{B.3}$$

Therefore the covariance matrix $\boldsymbol{C}$ is positive semi-definite.

If y^{th} element of $\boldsymbol{x}_i$ is defined as $\boldsymbol{x}_i(y)$, it is possible to write:

$$E\{(\boldsymbol{x}_i(y) - \boldsymbol{m}_{\boldsymbol{x}}(y))(\boldsymbol{x}_i(z) - \boldsymbol{m}_{\boldsymbol{x}}(z)) = E\{(\boldsymbol{x}_i(z) - \boldsymbol{m}_{\boldsymbol{x}}(z))(\boldsymbol{x}_i(y) - \boldsymbol{m}_{\boldsymbol{x}}(y)) \tag{B.4}$$

Therefore covariance matrix is also symmetric.

Covariance matrices are symmetric positive definite when they first initialized to filter. But propagating the covariance matrices removes symmetric positive definiteness mainly because of discretization and computational accuracy. In order to increase the numerical stability of a filter, covariance matrix can be checked at every propagation step and forced to be positive semidefinite.

There are different methods for checking symmetric positive definiteness of a matrix. The most computationally efficient method is proposed to be checking if Cholesky decomposition of a matrix exists. Cholesky decomposition decomposes a matrix $\boldsymbol{A}$ by representing this matrix as the product of a lower triangular matrix $\boldsymbol{L}$ with real and positive diagonal entries and its conjugate transpose $\boldsymbol{L}^*$

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^* \tag{B.5}$$

Writing Cholesky decomposition explicitly yields:

$$\begin{bmatrix} \boldsymbol{L}_{11} & 0 & 0 \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} & 0 \\ \boldsymbol{L}_{31} & \boldsymbol{L}_{32} & \boldsymbol{L}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{21} & \boldsymbol{L}_{31} \\ 0 & \boldsymbol{L}_{22} & \boldsymbol{L}_{32} \\ 0 & 0 & \boldsymbol{L}_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11}^2 & \boldsymbol{L}_{21}\boldsymbol{L}_{11} & \boldsymbol{L}_{31}\boldsymbol{L}_{11} \\ \boldsymbol{L}_{21}\boldsymbol{L}_{11} & \boldsymbol{L}_{21}^2 + \boldsymbol{L}_{22}^2 & \boldsymbol{L}_{31}\boldsymbol{L}_{21} + \boldsymbol{L}_{32}\boldsymbol{L}_{22} \\ \boldsymbol{L}_{31}\boldsymbol{L}_{11} & \boldsymbol{L}_{31}\boldsymbol{L}_{21} + \boldsymbol{L}_{32}\boldsymbol{L}_{22} & \boldsymbol{L}_{31}^2 + \boldsymbol{L}_{32}^2 + \boldsymbol{L}_{33}^2 \end{bmatrix} \tag{B.6}$$

Diagonal entries of this matrix can be written as[16]:

$$\boldsymbol{L}_{jj} = \sqrt{\boldsymbol{A}_{jj} - \sum_{k=1}^{j-1} \boldsymbol{L}_{jk}^2} \tag{B.7}$$

Since matrix $\boldsymbol{L}$ has to have real and positive diagonal entries, the expression in square root in Eq. B.7 is always greater than or equal to zero if such decomposition exists. Matrix $\boldsymbol{A}$ is real and positive semi-definite if such decomposition exist. Thi as shown below:

$$\begin{aligned} \boldsymbol{x}\boldsymbol{A}\boldsymbol{x}^* &= \boldsymbol{x}\boldsymbol{L}\boldsymbol{L}^*\boldsymbol{x}^* \\ &= (\boldsymbol{x}\boldsymbol{L})(\boldsymbol{x}\boldsymbol{L})^* \\ &= \|\boldsymbol{x}\boldsymbol{L}\|^2 \\ &\geq 0 \end{aligned} \tag{B.8}$$

A covariance matrix can be forced to be positive semi-definite by using nearest positive semi-definite matrix to original covariance matrix. A method is proposed by

Higham in [25]. Higham states that: "The nearest symmetric positive semidefinite matrix in the Frobenius norm to an arbitrary real matrix $\boldsymbol{A}$ is shown to be $(\boldsymbol{B}+\boldsymbol{H})/2$, where $\boldsymbol{H}$ is the symmetric polar factor of $\boldsymbol{B} = (\boldsymbol{A} + \boldsymbol{A}^T)/2$". Algorithm consist of following steps:

1. Symmetrize matrix $\boldsymbol{A}$ by:

$$\boldsymbol{B} = (\boldsymbol{A} + \boldsymbol{A}^T)/2 \tag{B.9}$$

2. Calculate symmetric polar factor of matrix $\boldsymbol{B}$

   - Calculating the singular value decomposition of $\boldsymbol{B}$:

$$\boldsymbol{B} = \boldsymbol{U\Sigma V}^T \tag{B.10}$$

     where $\boldsymbol{\Sigma}$ is diagonal and $\boldsymbol{V}$ is unitary matrices.

   - Calculate symmetric polar factor:

$$\boldsymbol{H} = \boldsymbol{V\Sigma V}^T \tag{B.11}$$

3. Find closest positive semi-definite matrix to matrix $\boldsymbol{A}$:

$$\hat{\boldsymbol{A}} = (\boldsymbol{B} + \boldsymbol{H})/2 \tag{B.12}$$

4. Ensure the symmetry:

$$\tilde{\boldsymbol{A}} = (\hat{\boldsymbol{A}} + \hat{\boldsymbol{A}}^T)/2 \tag{B.13}$$