CRANFIELD UNIVERSITY

VASKO SAZDOVSKI

INERTIAL NAVIGATION AIDED BY SIMULTANEOUS LOCALIZATION AND MAPPING

CENTRE FOR AUTONOMOUS SYSTEMS
DEPARTMENT OF INFORMATICS AND SYSTEMS ENGINEERING

PhD THESIS
Academic year: 2011 – 2012

Supervisor: Dr. Peter M.G. Silson
April 2012

CRANFIELD UNIVERSITY

CENTRE FOR AUTONOMOUS SYSTEMS
DEPARTMENT OF INFORMATICS AND SYSTEMS ENGINEERING

PhD THESIS

Academic year: 2011 – 2012

VASKO SAZDOVSKI

INERTIAL NAVIGATION AIDED BY SIMULTANEOUS LOCALIZATION AND MAPPING

Supervisor: Dr. Peter M.G. Silson

April 2012

This thesis is submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

I would like to dedicate this thesis to my family...

Acknowledgements

You see things; and you say "Why?" But I dream things that never were; and I say "Why not?"

George Bernard Shaw

I would like first to express my sincere and heartfelt gratitude to my Supervisor **Dr. Peter M.G. Silson** for his precious help and guideness throughout the research study. His supports without reservation and critics have brought bright light in this research and are still present motivating me for searching beyond the expected.
I due appreciation and dedication to **Prof. Antonios Tsourdos**, Head of the Autonomous Systems Group. Prof. Antonios thank you very much for being always there for me, giving me the right critics, directions and support.
Next I express my deepest appreciation to our respected **Prof. Emeritus Brian A. White** for all the experience, wisdom and understanding that he brought in this thesis.
Last but not the least I would like to thank to **Dr. Nabil Aouf** and **Dr. John Economou** for their valuable support, guidance and suggestions given during the initial research period.
My greatest depth belongs to my bellowing family and friends that supported me throughout the research study.

Abstract

Unmanned aerial vehicles technologies are getting smaller and cheaper to use and the challenges of payload limitation in unmanned aerial vehicles are being overcome. Integrated navigation system design requires selection of set of sensors and computation power that provides reliable and accurate navigation parameters (position, velocity and attitude) with high update rates and bandwidth in small and cost effective manner. Many of today's operational unmanned aerial vehicles navigation systems rely on inertial sensors as a primary measurement source. Inertial Navigation alone however suffers from slow divergence with time. This divergence is often compensated for by employing some additional source of navigation information external to Inertial Navigation. From the 1990's to the present day Global Positioning System has been the dominant navigation aid for Inertial Navigation. In a number of scenarios, Global Positioning System measurements may be completely unavailable or they simply may not be precise (or reliable) enough to be used to adequately update the Inertial Navigation hence alternative methods have seen great attention. Aiding Inertial Navigation with vision sensors has been the favoured solution over the past several years. Inertial and vision sensors with their complementary characteristics have the potential to answer the requirements for reliable and accurate navigation parameters.

In this thesis we address Inertial Navigation position divergence. The information for updating the position comes from combination of vision and motion. When using such a combination many of the difficulties of the vision sensors (relative depth, geometry and size of objects, image blur and etc.) can be circumvented. Motion grants the vision sensors with many cues that can help better to acquire information

about the environment, for instance creating a precise map of the environment and localize within the environment.

We propose changes to the Simultaneous Localization and Mapping augmented state vector in order to take repeated measurements of the map point. We show that these repeated measurements with certain manoeuvres (motion) around or by the map point are crucial for constraining the Inertial Navigation position divergence (bounded estimation error) while manoeuvring in vicinity of the map point. This eliminates some of the uncertainty of the map point estimates i.e. it reduces the covariance of the map points estimates. This concept brings different parameterization (feature initialisation) of the map points in Simultaneous Localization and Mapping and we refer to it as concept of aiding Inertial Navigation by Simultaneous Localization and Mapping.

We show that making such an integrated navigation system requires coordination with the guidance and control measurements and the vehicle task itself for performing the required vehicle manoeuvres (motion) and achieving better navigation accuracy. This fact brings new challenges to the practical design of these modern jam proof Global Positioning System free autonomous navigation systems.

Further to the concept of aiding Inertial Navigation by Simultaneous Localization and Mapping we have investigated how a bearing only sensor such as single camera can be used for aiding Inertial Navigation. The results of the concept of Inertial Navigation aided by Simultaneous Localization and Mapping were used. New parameterization of the map point in Bearing Only Simultaneous Localization and Mapping is proposed. Because of the number of significant problems that appear when implementing the Extended Kalman Filter in Inertial Navigation aided by Bearing Only Simultaneous Localization and Mapping other algorithms such as Iterated Extended Kalman Filter, Unscented Kalman Filter and Particle Filters were implemented. From the results obtained, the conclusion can be drawn that the nonlinear filters should be the choice of estimators for this application.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today Unmanned Aerial Vehicles (UAV's) have become an indispensable ingredient for many applications where human operation is considered unnecessary, too dangerous or impossible [Shim et al., 2005]. These applications include planetary explorations, environmental and climate research and monitoring, traffic monitoring, inspection of man-made structures such as power lines and pipelines, urban planning, pollution studies and many others. Today's intelligence gathering, surveillance and reconnaissance missions are not possible to be carried out without UAV's.

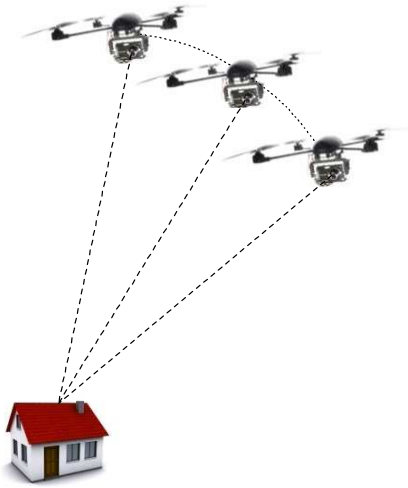

Figure 1.1: Quadrotor UAV equipped with single video camera navigating over a house

All these applications and missions require UAV's to operate in a partially known or unknown environments. There we usually have limited knowledge of the environment or we don't have any knowledge, see figure (1.1).
Typically in these applications and missions the need for safety is arising. Safety here simply implies that the UAV's are not damaged or destroyed during the mission [Sivakumar and Sengupta, 2004]. Part from factors such as vehicle constraints, environment factors (strong winds, icing, thunderstorms) and enemy fire, reliable navigation parameters play crucial role in the safety of the UAV's. Supplying wrong or not precise navigation parameters (position, velocity and attitude) to the Guidance and Control System very often turn to be catastrophic for the UAV's. Simple approach of integration of several complementary navigation systems will provide more accuracy than that of individual systems. Many of today's operational UAV navigation systems rely on inertial sensors as a primary measurement source. From the early 1960's when the Inertial Navigation (IN) was introduced has been used for many navigation tasks and on many types of vehicles: land, sea, air and low orbit. The majority of the Inertial Navigation Systems (INS) were extremely expensive because, in part, of the cost of high-quality, well characterized sensors and typically the need for a stabilized sensor platform. This high cost limited such systems primarily to military, scientific, and commercial aircraft applications. In addition, the use of stabilized platforms resulted in this class of INS having size and power requirements too large for many applications [Farrell and Barth, 1999].
High-quality, individually selected and characterized sensors are required in certain navigation applications to meet the relatively high-accuracy requirements over long-duration mission without external positioning aiding. Advances in the MEMS technology and material processing have made it possible to produce small, low-cost inertial sensors. Although the low-cost sensors cannot be expected to meet the accuracy and precision specifications for all navigation applications, they have opened the doors and bringed the IN technology to the wide university and education communities for research and development. Today, almost every university runs a laboratory or has a research group for autonomous systems, where they use various inertial sensors in their research.
There are two primary INS implementation approaches. The first approach uses a

stabilized platform mechanized as the vehicle moves to maintain sensor alignment with a predetermined reference frame. The second approach uses a strap-down platform rigidly attached to the vehicle reference (body) frame. The stabilized-platform has two main advantages over that of strapdown systems [Farrell and Barth, 1999]:

- The inertial sensors are subjected only to small angular rates. In a high-accuracy system without external aiding, this is important for three reasons:(1) sensors nonlinearity may be exited by high dynamic loads; (2) lower sensor bandwidth results in an increased signal-to-noise ratio; (3) lower sensor range allows increased sensor sensitivity.
- The computational load of a stabilized-platform system is smaller than that of a strapdown system.

But this stabilized-platform has several detractors which make it undesirable:

- Friction in the bearings exists and motors are not perfect (i.e. dead zones, etc.).
- The process of keeping the platform aligned with the reference frame consumes power which is not practical for an embedded system.
- There are high cost involved due to the need for high quality motors, bearings and other mechanical parts.
- Regular maintenance requires trained and certified personnel which can be difficult for an autonomous navigation systems.

The main benefits of the strap-down approach are the decrease in navigation system size, power, and cost because of the elimination of the stabilized platform and its actuators [Farrell and Barth, 1999]. This system is a major hardware simplification of the old gimballed systems. The accelerometers and gyros are mounted in body coordinates and are not mechanically moved. Instead, a software solution is used to keep track of the orientation of the Inertial Measurement Unit (IMU) (and vehicle) and rotate the measurements from the vehicle (body)

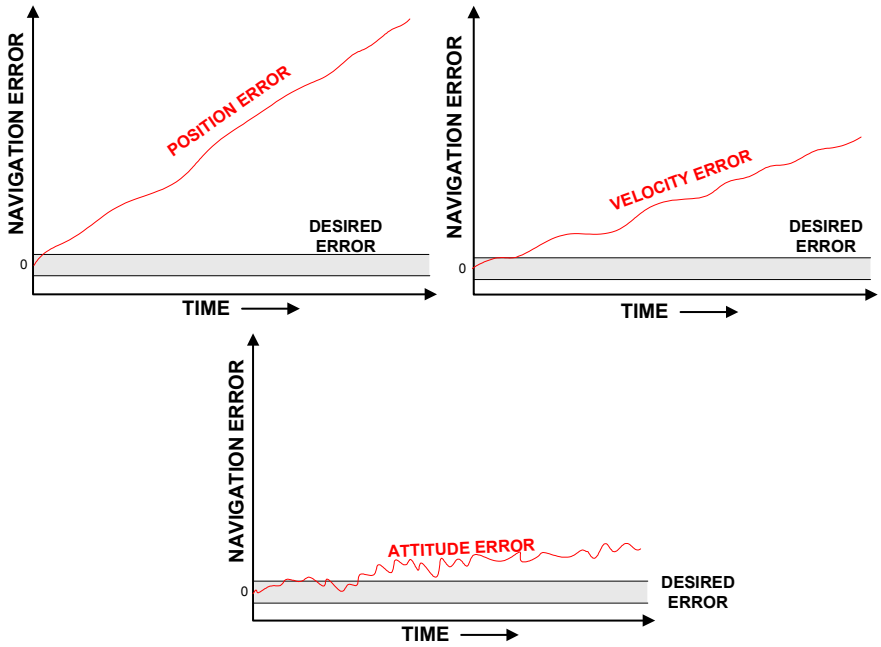

Figure 1.2: Inertial Navigation diverging slowly from the real measurements with time

frame to the reference frame. Inertial Navigation (IN) alone however suffers many drawbacks that complicate its usage as a standalone navigation system. Instrumentation, computational, alignment and environmental errors cause the IN to diverge slowly with time, as shown on figure (1.2). The inertial divergence is often compensated for by employing some additional source of navigation information, external to IN. Before the 1990's near landfall, the IN was updated by a VHF omnidirectional range (VOR) navigation system fix after which the aircraft would navigate to its destination on the VOR airways. On long flights, in order to correct the inevitable drift of the inertial navigators, long range (LORAN) navigation system was used. In the 1970's because of the scanty coverage of the LORAN chains outside the developed areas, it was supplemented by OMEGA navigation system [Kayton, 2003]. From the 1990's to the present day Global Positioning System (GPS) has been the dominant navigation aid for IN, see figure (1.3). GPS and IN have complementary characteristics and GPS aided IN has been used successfully for surveying, mining, dredging, automotive, railroad, and aircraft applications for commercial, military and scientific customers [Farrell

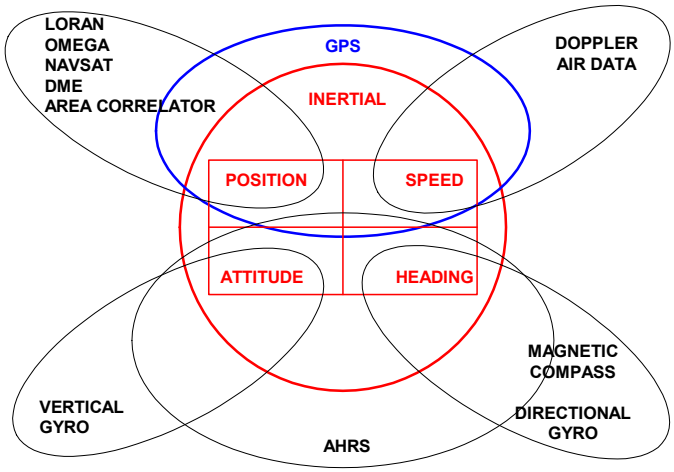

Figure 1.3: Additional sources of navigation parameters to Inertial Navigation [Ching-Fang, 1991]. (Noninteractive aiding information's)

and Barth, 1999]. However, in a number of scenarios, GPS measurements may be completely unavailable or they simply may not be precise (or reliable) enough to be used to adequately update the IN. Tunnels, canyons, forests and urban areas are typical examples where this can occur and, in addition, in a military environment, the GPS signal may be deliberately jammed. Driven primarily by these reasons alternative methods of aiding IN have seen great attention. Worth mentioning here as a concept is IN aided using aircraft dynamics, an approach that is described in [Koifman and Bar-Itzhack, 1999], where the authors show that under some specific conditions the aircraft dynamic model can be used to aid the IN. Aiding IN with vision sensors has been the favored solution over the past several years. The idea is to develop a viable solution (i.e. sensor fusion algorithm) that will provide a source of reliable aiding information to IN from pure vision means. The algorithm should not assume any knowledge of the environment and should not require information from separate range sensors. This would create a passive jam proof, GPS free, autonomous navigation system. Traces of the idea of aiding IN with vision sensors can be found in the late 1970's. In [Bar-Itzhack, 1978] the author addresses the problem of IN error divergence with time and he proposes that the vehicle carry a sighting device (SD) like radar or optical devices in order to estimate the IN errors.

In [Sinopoli et al., 2001], [Watanabe et al., 2005] vision information is used for detection and investigation of objects of interest as well as the vehicle itself in order to plan a path from its current position to the terminal point. Video camera information in [Call et al., 2006], [Watanabe et al., 2005] is used as primary sensor for obstacle detection and avoidance. In [Brzezinska et al., 1998] IN, GPS and video camera information are integrated for creating an airborne mobile mapping system. Some preliminary results of navigation aided image processing and image processing oriented sensor management architecture for UAV surveillance are presented in [Nygards et al., 2004]. A control and image processing system that enables a UAV to track structures such as oil-gas pipes, roads, bridges, canals etc. is described in [Rathiman et al., 2006] in which the system was tested with a road and an aqueduct. Biologically inspired navigation methods for example the "bee navigation" method, based on optical flow field are studied in [Franceschini, 2004], [Lerner et al., 2004]. In these methods, ego-motion is derived from the optical-flow field. In ego-motion integration approach, the motion of the camera with respect to itself is estimated [Lerner et al., 2004]. Once the ego-motion is obtained, one can integrate this motion to derive the camera's path.
Vision based navigation algorithm for a vertical take-off and landing (VTOL) UAV where the navigation Extended Kalman filter (EKF) filter uses vision data from ground feature tracking based on Lucas-Kanade algorithm in order to compensate for the GPS failures is developed in [Koch et al., 2006]. Similarly, in [Webb and Prazenica, 2007] the authors show that it is possible to estimate an aircraft states from a set of tracked fixed feature points in inertial frame by using single video camera mounted on a micro air vehicle (MAV).
All these applications indicate a widespread use of vision sensors in various UAV tasks but not for directly aiding IN per se. The previously mentioned IN aiding methods (VOR, LORAN, OMEGA and GPS), fall into a group of methods using so-called non interactive aiding information, figure (1.3). The non interaction being with the surrounding (environment) of the vehicle since the measurements are usually given in a form of position fixes, ground speed, radials and are computed external to the IN i.e. are not computed relative the vehicle.

Vision sensors fall in the group of interactive aiding information since they provide relative measurements *between* the vehicle and its environment. Because

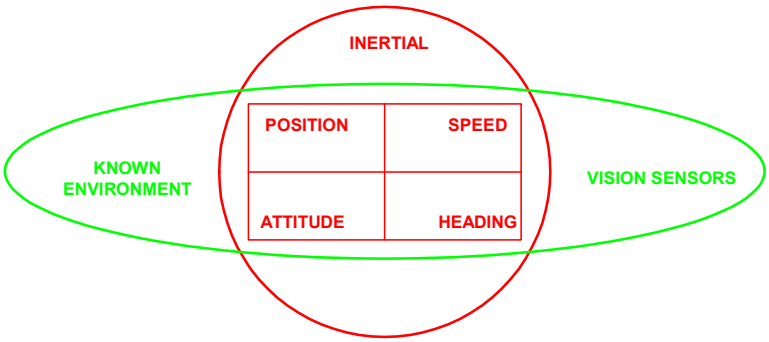

Figure 1.4: Additional sources of navigation parameters to Inertial Navigation (vision sensors as interactive relative aiding information's)

of the nature of the measurements being relative, they interconnect the IN and the environment. If we have a known environment then, using the relative measurements, we can improve IN performance (localization) and vice versa. Also, if we have perfect IN parameters then we can create a map of the environment (mapping), figure (1.4). In reality both the environment and IN are uncertain and therefore this is a "chicken and egg" problem and the solution appears to be to run the localization and the mapping simultaneously as the vehicle moves through the environment. This is the Simultaneous Localization and Mapping (SLAM) problem. SLAM permits navigation in an initially unknown environment using only onboard sensing (relative) measurements, and therefore can be used in situations where GPS is unavailable. In the case when the vehicle process model is represented by IN equations, and when the relative measurements between the vehicle and the map points are measured by vision sensors, this can be considered as special case of vision aided IN. This case is called Airborne SLAM [Kim, 2004], [Langelaan, 2006], [Watkins, 2007], [Ivey and Johnson, 2006], [Jung and Lacroix, 2003], where as the onboard sensors are taking the relative measurements of the map points and the vehicle itself the SLAM estimator augments the map point positions to the map and begins to estimate the vehicle and map states together with successive observations.

In this thesis we address IN position divergence. The information for updating the position comes from combination of vision and motion. When using such a combination many of the difficulties of the vision sensors (relative depth, geometry

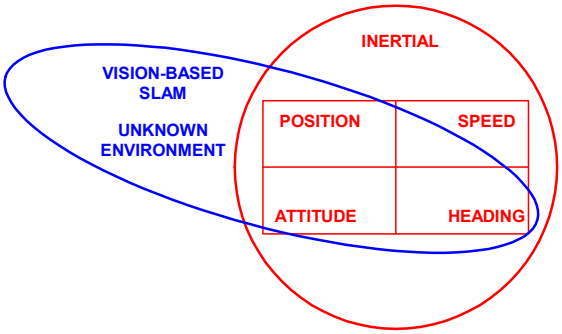

Figure 1.5: Inertial Navigation aided by Simultaneous Localization and Mapping using Vision Sensors

and size of objects, image blur and etc.) can be circumvented. Motion grants the vision sensors with many cues that can help better to acquire information about the environment, for instance creating a precise map of the environment and localize within the environment. We propose changes to the SLAM augmented state vector in order to take repeated measurements of the map point. We show that these repeated measurements with certain manoeuvres (motion) around or by the map point are crucial for constraining the IN position divergence (bounded estimation error) while manoeuvring in vicinity of the map point. This eliminates some of the uncertainty of the map point estimates i.e. it reduces the covariance of the map points estimates. This concept brings different parameterization (feature initialisation) of the map points in SLAM and we refer to it as concept of aiding IN by SLAM. We show that making such an integrated navigation system requires coordination with the guidance and control measurements and the vehicle task itself for performing the required vehicle manoeuvres (motion) and achieving better navigation accuracy. This fact brings new challenges to the practical design of these modern jam proof GPS free autonomous navigation systems. Further to the concept of aiding IN by SLAM we have investigated how a bearing only sensor such as single camera can be used for aiding IN. The results of the concept of IN aided by SLAM were used. New parameterization of the map point in BOSLAM is proposed. Because of the number of significant problems that appear when implementing the Extended Kalman Filter (EKF) in IN aided by BOSLAM (Jacobians, highly nonlinear system, second and higher order errors that are not

negligible) other algorithms such as Iterated Extended Kalman Filter (IEKF), Unscented Kalman Filter (UKF) and Particle Filters (PF) were implemented in IN aided by BOSLAM, which perform better then EKF in this application.

1.1 Related Work

All the up to date advances in the area of aided IN can be divided in several sections, each one representing the approaches and the latest research findings.

1.1.1 GPS Aided IN

GPS aided IN today can be considered as well understood and mature integration solution. As we have mentioned before GPS aided IN have been used successfully for surveying, mining, dredging, automotive, railroad, and aircraft applications for commercial, military and scientific customers.
A large number of articles and papers that discuss about GPS aided IN are available today. Specialized books that treat IN, GPS and their integration in a unified framework, are also available [Farrell and Barth, 1999], [Grewal et al., 2001], [Biezad, 1999]. Most of the Kalman filtering textbooks use GPS aided IN as an example of typical Kalman filter implementation [Brown and Hwang, 1992], [Grewal and A.P.Andrews, 2001], [H.W.Sorenson, 1985], [Leondes, 1982]. This fact without no doubts emphasizes the popularity of this integration and solution. As will be described later in section 2.5 of the thesis, both GPS and IN measurements can be combined using either direct or indirect filter configurations.
Current research activities in this area are focused on implementation of different filter techniques then the classical Kalman Filter. The apparent goal is to "improve" the Kalman Filter results. GPS aided IN with implementation of Unscented Kalman Filter can be found in [Zhang et al., 2005], [Crassidis, 2006]. Expectation-maximization (EM) method for GPS aided IN for land vehicle navigation is proposed in [Huang and Leung, 2004]. The problem of GPS/IN integration by using a Rao-Blackwellized Particle filter is addressed in [Giremus et al., 2004], [Vernaza and Lee, 2006].

1.1.2 Vision Based Navigation

Visual information is the must nowadays in terms of navigation and guidance measurements for autonomous vehicles. This perception technique is famous because of its long-range, high resolution and most important because of its passive property (it does not emit energy, which makes it possible to incorporate other heat-sensitive sensors, such as infrared) [Zhenhe et al., 2007]. The latest advances in computer vision theory has paved the way and encouraged many researchers to implement vision in various applications from autonomous navigation and guidance of robots to medical diagnostics and food quality evaluation. On the hardware side, both the video camera and computer technology have made significant progress so that high resolution images are processed at real-time at high rates or transferred to the end users in part of a second. In [Sinopoli et al., 2001] and [Watanabe et al., 2004] vision information was used for detection and investigation of objects of interest and the vehicle itself in order to plan a path from its current position to the terminal point. Video camera information in [Call et al., 2006] and [Watanabe et al., 2005] is used as primary sensor for obstacle detection and avoidance. In [Brzezinska et al., 1998] INS, GPS and Video camera information's are integrated for creating an airborne mobile mapping system. Some preliminary results of navigation aided image processing and image processing oriented sensor management architecture for UAV IR/EO surveillance are presented in [Nygards et al., 2004]. The control and image processing system which enables an UAV to track structures like oil-gas pipes, roads, bridges, canals and etc. is described in [Rathiman et al., 2006]. There the system was tested with a road and an aqueduct. Biologically inspired navigation methods for example the "bee navigation" method, based on optical flow field are studied in [Franceschini, 2004], [Lerner et al., 2004]. In this methods from the optical-flow field ego-motion can be derived. In ego motion integration approach the motion of the camera with respect to itself is estimated [Lerner et al., 2004]. Once the ego-motion was obtained, one can integrate this motion to derive the camera's path. One of the factors that make this approach attractive is that no specific features need to be detected and only the correspondence between the two consecutive images should be found in order to derive the optical-flow field [Lerner et al., 2004]. The

weakness of ego-motion integration comes from the fact that small errors are accumulated during the integration process. Hence, the estimated camera's path is drifted and the pose estimation accuracy decrease with time. Vision based navigation algorithm for a VTOL-UAV where the navigation EKF filter uses vision data from ground feature tracking based on Lucas-Kanade algorithm in order to compensate for the GPS failures is developed in [Koch et al., 2006]. Similarly, in [Webb and Prazenica, 2007] the authors showed that it is possible to estimate the aircraft states from a set of tracked fixed feature points in inertial frame by using video camera mounted on a micro air vehicle (MAV).

1.1.3 Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping (SLAM) is the process of simultaneously estimating the state of an autonomous vehicle and the map points in the environment. It permits vehicle navigation in an initially unknown environment using only onboard sensing (relative) measurements, and therefore can be used in situations where GPS is unavailable. In the case when instead the ground vehicle model the driver of the process model is IN, and when the relative measurements between the vehicle and the landmarks are measured by vision sensors this can be considered as special case of vision aided IN. This case is used in the so called Airborne SLAM [Kim, 2004], [Langelaan, 2006], [Watkins, 2007], [Ivey and Johnson, 2006], [Jung and Lacroix, 2003] where as the onboard sensors are taking the relative measurements of the map points and the vehicle itself the SLAM estimator augments the map point positions to the map and begins to estimate the vehicle and map states together with successive observations. The fundamental advantage of SLAM algorithms is that they account for the statistical correlations that exist between the vehicle position and the landmarks positions. On the other hand, the main limitation of SLAM is its high computational complexity and thus performing SLAM in environments with thousands of landmarks still remains a great challenge.

1.2 Contributions of the Thesis

- Many of the authors in SLAM use trigonometric functions in their observation models. The use of trigonometric functions has, apart from the fact that they may have singularities at certain points, the disadvantage of making computer processing slow. As such, it is recommended to avoid trigonometric functions. Since vectors have power to simplify geometrical problems we have used vectors to explain the vision sensors geometry. Each measurement from the vision sensors is represented by a line. Instead of using angles like bearing or azimuth to the map points we use unit direction vectors to the map point. This approach presents an important trigonometric substitution and is more efficient in terms of computational speed and accuracy.

- In the theory of SLAM the unknown map points are added to the map when they are first observed from the vehicle. One map point estimate is added to the map for each map point. This initialization of the map points does not follow the accumulation of the vehicle error gradually. It maps the environment using the vehicle error positions and when closes the loop i.e. revisits the stored map points, it updates the augmented state vector (the vehicle state and the map). Within this parameterizations of the map points, taking repeated observations of the map point cannot further reduce the covariances of the map points. These facts ask for other alternative form of the augmented state vector in order to implement SLAM as sensor fusion algorithm for aiding IN and constraining the IN position divergence while manoeuvring in vicinity of the map point.

- When manoeuvring in vicinity of the map point, note in the SLAM theory is made to the SLAM augmented state vector in order to take repeated measurements of the map point. These repeated measurements with certain movements around or by the map point provide the valuable information for constraining the IN position divergence.

- The concept of aiding IN by SLAM clearly shows that making such an integrated navigation system requires coordination with the guidance and

control measurements and the vehicle task itself for performing the needing vehicle maneuvers (movements) and achieving better navigation accuracy. This facts brings new challenges to the practical design of these modern jam proof GPS free autonomous navigation systems.

- Further to the concept of aiding IN by SLAM, we focused our research on BOSLAM. We have investigated how a bearing only sensor such as single camera can be used for aiding IN. We used the results from SLAM aided IN. Since the range to the map points is unknown in BOSLAM, criterion needs to be used to choose certain number of "most promising" map point estimates in the augmented state vector. New parameterization of the map point in BOSLAM is proposed. Because of the number of significant problems that appear when implementing EKF in IN aided by BOSLAM (Jacobians, highly nonlinear system, second and higher order errors that are not negligible) other algorithms such as IEKF and UKF, PF are implemented, which perform better then EKF.

1.3 Reader's Guide

The remainder of this thesis is organized as follows:

Chapter 2: Inertial Navigation starts with the basic definition of Inertial Navigation, then caries out defining the coordinate reference frames, the shape of the Earth and variation of the Earth's gravity field used in deriving the navigation equations. These equations first are derived with respect to fixed (inertial) frame, then with respect to rotating frame. Detailed navigation equations are also included. This chapter ends with the part where the sources of error in Inertial Navigation and aided IN as alternative approach for dealing with the errors are described.

Chapter 3: Vision Sensors are nowadays considered essential for UAV navigation, firstly because they are cost effective, small, compact and reliable passive sensors and secondly because they are capable of providing a rich source of in-

formation about the vehicle environment. The single camera and stereo camera setups and its vector applications are explained in this chapter. The limitations of the both setups are carefully described.

Chapter 4: Sensor Fusion introduces us to the concept of sensor fusion. Here we present the most famous sensor fusion algorithms such as the classic linear Kalman Filter, the Extended Kalman Filter, Iterated Extended Kalman Filter, Unscented Kalman Filter and the Particle Filters.

Chapter 5: Simultaneous Localization and Mapping is the chapter where the main characteristics of the SLAM problem are elaborated. The current trends in SLAM are presented. Limitations of the SLAM solutions are included in this chapter as well.

Chapter 6: Inertial Navigation Aided by Simultaneous Localization and Mapping chapter describes the concept of aiding IN by SLAM. We describe the changes to the SLAM augmented state vector that are needed in order to take repeated measurements of the map point. We show that repeated measurements with certain movements around or by the map point are crucial for constraining the IN position divergence while manoeuvring in vicinity of the map point.

Chapter 7: Inertial Navigation Aided by Bearing-Only Simultaneous Localization and Mapping chapter follows the results from the previous chapter. Here we augment the state vector with certain number of estimates for each map point. We take repeated measurements of the map points with certain manoeuvres and constrain the vehicle position divergence while manoeuvring in vicinity of the map point. New parameterization of the map point in BOSLAM is proposed in this chapter. The performance of four nonlinear filters (EKF, IEKF, UKF and PF) has been investigated. From the results we conclude that nonlinear filters should be the choice of estimators for the problem of IN aided by BOSLAM.

Chapter 8: Conclusions summarizes the results of the research. Future work activities are also presented here.

Chapter 2

Inertial Navigation

Inertial Navigation (IN) is the process of calculating position by integration of velocity and computing velocity by integration of the total acceleration where the total acceleration is calculated as the sum of gravitational acceleration, plus the acceleration produced by applied non gravitational forces (known as specific force acceleration) [Savage, 1998a], [Savage, 1998b]. For the several past decades, IN has been used for many navigation tasks. The majority of these systems were extremely expensive and this high cost limited the usage of such systems primarily to military and commercial aircraft application. High quality and characterized sensors in the Inertial Measurement Unit (IMU) are required in certain navigation applications to meet the relatively high accuracy requirements over a long-duration mission. Today increasing powerful, smaller, and less expensive computational equipments allows strap-down navigation algorithms to be implemented accurately on inexpensive small packages.

2.1 Coordinate Frames and Transformations

Fundamental to the process of Inertial Navigation is the precise definition of a number of Cartesian coordinate (reference) frames. Each frame is an orthogonal, right-handed, coordinate frame or axis set. For navigation over the Earth, it is necessary to define axis sets which allow the inertial measurements to be related to the cardinal directions of the Earth, that is, frames which have a physical

significance when attempting to navigate in the vicinity of the Earth.

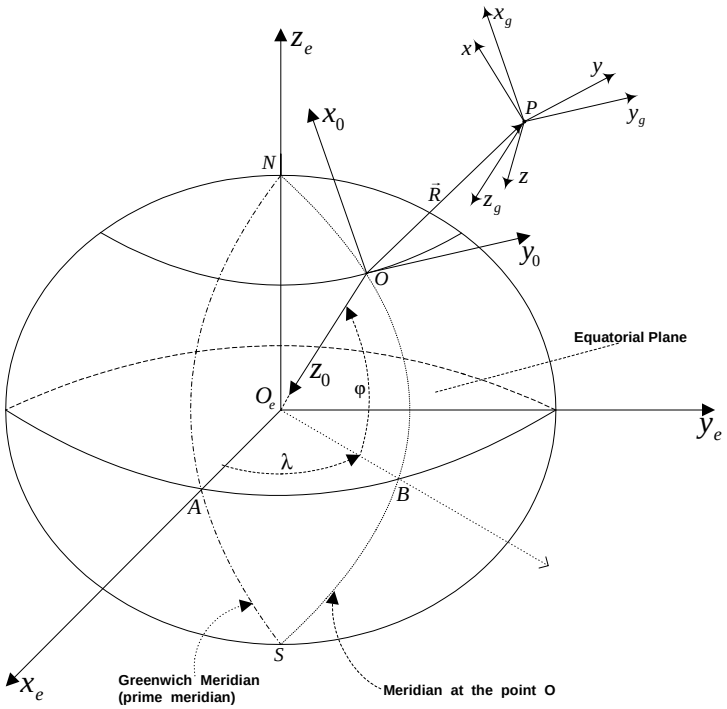

Figure 2.1: Coordinate frames. Earth Centered Earth Fixed Frame $E(O_e; x_e; y_e; z_e)$, Navigation frame $N(O; x_0; y_0; z_0)$, and Body frame $B(P; x; y; z)$

2.1.1 The shape of the Earth

Before we define the reference frames and determine position on the Earth using inertial measurements, it is necessary to make some assumptions regarding the shape of the Earth. The spherical model of the Earth is not sufficiently representative for very accurate navigation. Owing to the slight flattening of the Earth at the poles, it is customary to model the Earth as a reference ellipsoid which approximates more closely to the true geometry.

One such reference ellipsoid model is the WGS 84 model, defined by the World Geodetic System Committee in 1984 [Farrell and Barth, 1999], [Titterton, 1997]. The WGS 84 ellipsoid parameters as defined in Table 2.1 are used throughout the discussion.

Length of the semi-major axis	$a = 6378137\ m.$
Length of the semi-minor axis	$b = 6356752.3142\ m.$
Flattering of the ellipsoid	$f = (a - b)/a = 1/298.2572\ 23563$
Major eccentricity of the ellipsoid	$e = [\ f(2 - f\ )]^{1/2} = 0.0818191908\ 426$
Earth's rate	$\omega_{ie} = 7.292115 \times 10^{-5}\ rad\ /\ s$

Table 2.1: WGS 84 ellipsoid parameters

2.1.2 Variation of the Earth's gravity field

Accelerometers provide measurements of the difference between the acceleration with respect to inertial space and the gravitational attraction acting at the location of the navigation system. In order to extract the precise estimates of true acceleration needed for accurate navigation in the vicinity of the Earth, it is necessary to model accurately the Earth's gravity field. The gravity field is defined as the acceleration field arising from the combined effects of the earth's gravitational field and the inward centripetal acceleration due to the earth's rotation. If we assume that the earth is conformed to the reference ellipsoid model, the gravity vector would be normal to the reference ellipsoid and its magnitude could be precisely calculated. Since the actual earth's surface deviates from the reference ellipsoid both the magnitude and the direction of the gravity vector vary with position on the Earth's surface and altitude above it. Such deviations in the magnitude and direction of the gravity vector are known as gravity anomalies. Mathematical representations of the Earth's gravitational field, the centripetal acceleration and the gravity anomalies are discussed in more depth in [Britting, 1971], [Farrell and Barth, 1999]. For many applications it is sufficient to assume the variation of gravity field is with altitude and is given by the following relation [Farrell and Barth, 1999]:

$$\mathrm{g(h)} \approx \mathrm{g_0}(1 - \frac{2h}{a}) \tag{2.1}$$

where g_0 is the equatorial value of gravity which for the WGS 84 model is $g_0 = 9.7804900\ m/s^2$.

2.1.3 Earth Centered Earth Fixed Frame (ECEF)

The Earth Centered Earth Fixed Frame $E(O_e; x_e; y_e; z_e)$ as shown on figure 2.1 has its origin fixed to the center of the Earth and its axis $O_e x_e$ and $O_e y_e$ are fixed to the Earth and rotate together, with a rate ω_{ie} around the $O_e z_e$ axis. The $O_e x_e$ axis is extended through the intersection of the prime (Greenwich) meridian and the equator (point A on figure 2.1). The position of any point on the Earth for example the point O, can be described by the ECEF spherical coordinates: $\varphi - latitude$ and $\lambda - longitude$, or by the ECEF rectangular coordinates: x_e, y_e, z_e, as shown on figure 2.1.

2.1.4 Navigation frame

The Navigation frame $N(O; x_0; y_0; z_0)$, has its origin on the surface point on the Earth (point of interest), for example point O, see figure 2.1. The z_0 axis is pointing toward the interior of the Earth perpendicular to the reference ellipsoid. The x_0 axis points to true north while y_0 axis points east. For short term navigation where the navigation period is short as required for many tactical UAV applications and missions the rotation of the Earth can be neglected and the Navigation frame can be considered as inertial frame. Very often there is a need to transform measurements from the ECEF frame to Navigation frame and opposite. The transformation matrix of vectors from the ECEF frame to Navigation frame is given as [Farrell and Barth, 1999]:

$$C_e^n = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\sin\lambda & -\sin\varphi \end{bmatrix} \tag{2.2}$$

The transformation of the coordinates of a point from the Navigation frame to the ECEF frame is:

$$[x_e, y_e, z_e]^T = [x_e(0), y_e(0), z_e(0)]^T + C_n^e [x_0, y_0, z_0]^T \tag{2.3}$$

where, $C_n^e = (C_e^n)^T$ and $[x_e(0), y_e(0), z_e(0)]^T$ are the ECEF coordinates of the origin of the Navigation frame.

2.1.5 Body frame

The Body frame $B(P; x; y; z)$ is rigidly attached to the vehicle usually at the center of gravity. The x axis is defined in the forward direction. The z axis is perpendicular to the x axis and is pointing to the bottom of the vehicle. The y axis completes the right handed orthogonal coordinate system. The transformation matrix for transforming Navigation frame coordinates into vehicle Body frame coordinates is defined by the series of three plane rotations involving the three Euler angles (ϕ-roll, θ-pitch and ψ-yaw), typically used in aerospace applications.

$$C_n^b = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & \mathrm{s}\psi\mathrm{c}\theta & -\mathrm{s}\theta \\ -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{s}\phi \\ \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{c}\phi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\psi\mathrm{s}\theta\mathrm{c}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \tag{2.4}$$

The transformation matrix for transforming Body frame coordinates into Navigation frame coordinates is $C_b^n = (C_n^b)^T$. When using the matrix (2.4) and when the pitch angle is $\theta = \pm 90^\circ$ we have singularity in the matrix. Often for defining the transformation matrix we can use the quaternion method for parameterizing the rotation angles. Quaternion parametrizations are singularity free and are more computationally efficient then the Euler angles [Farrell and Barth, 1999].

$$C_b^n = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \tag{2.5}$$

The quaternion approach is often the preferred implementation approach as the linearity of the quternions, the lack of trigonometric functions, and the small numbers of parameters that allow efficient implementation [Farrell and Barth, 1999]. The only shortcoming of the quaternions is that they don't have simple

geometrical representation and because of this can't be measured directly. But this is not a problem because there is a connection between the relation (2.4) and (2.5), more about this can be found in [Titterton, 1997].

2.2 Navigation Equations

2.2.1 Navigation with respect to a fixed frame

In a situation where we need to navigate with respect to inertial or fixed non-accelerating and non-rotating set of axis, the measured components of specific force and estimates of the gravitational field are summed to determine the components with respect to that space fixed reference frame. These quantities can be integrated once in order to determine the velocity and once more in order to determine the position in that frame. The mathematical expression of this process can be done in the following manner. Let r represent the position vector of some point P with respect to O, the origin of the reference frame, as shown on figure 2.2.

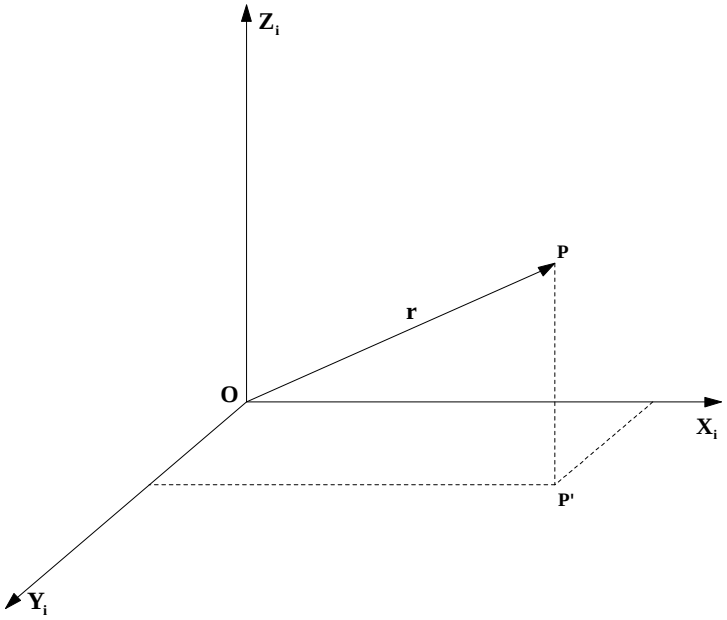

Figure 2.2: Position vector with respect to reference frame

The acceleration of P with respect to this space fixed axis set denoted by the subscript $_i$, (read inertial) is defined by:

$$a_i = \frac{d^2}{dt^2} r \,|_i \tag{2.6}$$

A triad of perfect accelerometers will provide a measure of the specific force acting at the point P where:

$$f = \frac{d^2}{dt^2} r \,|_i - g \tag{2.7}$$

in which g is the mass attraction gravitation vector. Rearranging equation (2.7) yields the following equation:

$$\frac{d^2}{dt^2} r \,|_i = f + g \tag{2.8}$$

This is called the navigation equation since, with suitable integration it yields the navigation quantities of velocity and position. The first integral gives the velocity of point P with respect to the i-frame:

$$v_i = \frac{d}{dt} r \,|_i \tag{2.9}$$

whilst a second integration gives its position.

2.2.2 Navigation with respect to a rotating frame

In practice, we often need to derive the estimates of a vehicle's velocity and position with respect to a rotating reference frame, as when navigating in the vicinity of the Earth. In this situation, additional apparent forces will be acting which are functions of the reference frame motion. This results in a revised form of the navigation equation which may be integrated to determine the ground speed of the vehicle v_e, directly. Alternatively, v_e may be computed from the inertial

velocity v_i, using the theorem of Coriolis, as follows,

$$v_e = \frac{d}{dt} r \mid_e = v_i - \Omega_{ie} \times r \tag{2.10}$$

where $\Omega_{ie} = \begin{bmatrix} 0 & 0 & \omega_{ie} \end{bmatrix}^T$ is the turn rate of the Earth frame with respect to the i-frame and $\times$ denotes a vector cross product.

2.3 Detailed Navigation Equations

The navigation equation (2.8) may be solved in any one of the reference frames defined in section 2.1. If the Earth frame is chosen, then the solution of the navigation equation, most commonly the velocity is given in the local Navigation frame and the position is given with ECEF rectangular or spherical coordinates. One such mechanization is the one derived by Britting in [Britting, 1971]. This mechanization is suitable for navigation of UAV's over large distances around the Earth and it also allows very easy integration of the information's from the GPS receiver. As derived in [Britting, 1971], the IN mechanization equations are the computer implementation of the IN equations, where $\hat{x}$ denotes the computed value of the variable x. The IN state vector x, which can be written as, $\dot{x} = f(x, u, t) + w$ contains the following states:

$$x = [\ \varphi \quad \lambda \quad h \quad V_N \quad V_E \quad V_D \quad q_0 \quad q_1 \quad q_2 \quad q_3]^T$$

As mentioned before the position is given with ECEF spherical coordinates where, φ is the latitude, is λ longitude, both in radians, h is the altitude in meters, V_N, V_E, V_D are the components of the velocity vector $v^n = [V_N, V_E, V_D]$ given in Navigation frame, q_0, q_1, q_2, q_3 and are Euler-Rodriguez quaternions, representing the attitude. Inputs u to the IN are the measurements from the IMU such as accelerations $\begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T$ and angular rates $[\ p \quad q \quad r]^T$ measured in body frame and w represents white noise process with known covariance. From the measured accelerations we can calculate the components of the velocity by the

following equations:

$$\dot{\hat{V}}_N = -(\dot{\hat{\lambda}} + 2\omega_{ie})\sin(\hat{\varphi})\hat{V}_E + \dot{\hat{\varphi}}\hat{V}_D + \hat{f}_N$$

$$\dot{\hat{V}}_E = (\dot{\hat{\lambda}} + 2\omega_{ie})\sin(\hat{\varphi})\hat{V}_N + (\dot{\hat{\lambda}} + 2\omega_{ie})\cos(\hat{\varphi})\hat{V}_D + \hat{f}_E$$

$$\dot{\hat{V}}_D = -\dot{\hat{\varphi}}\hat{V}_N - (\dot{\hat{\lambda}} + 2\omega_{ie})\cos(\hat{\varphi})\hat{V}_E + \hat{f}_D + \hat{g}$$

$$\begin{bmatrix} f_N \\ f_E \\ f_D \end{bmatrix} = C_b^n \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

where, ω_{ie} is the inertial rate of rotation of the earth in rad/s, g is the local gravitational acceleration in m/s^2, $\begin{bmatrix} f_N & f_E & f_D \end{bmatrix}^T$ denotes the specific force vector in Navigation frame, and C_b^n is the transformation matrix (2.5) given in quaternion form which transforms the measurements from the Body frame to the Navigation frame.

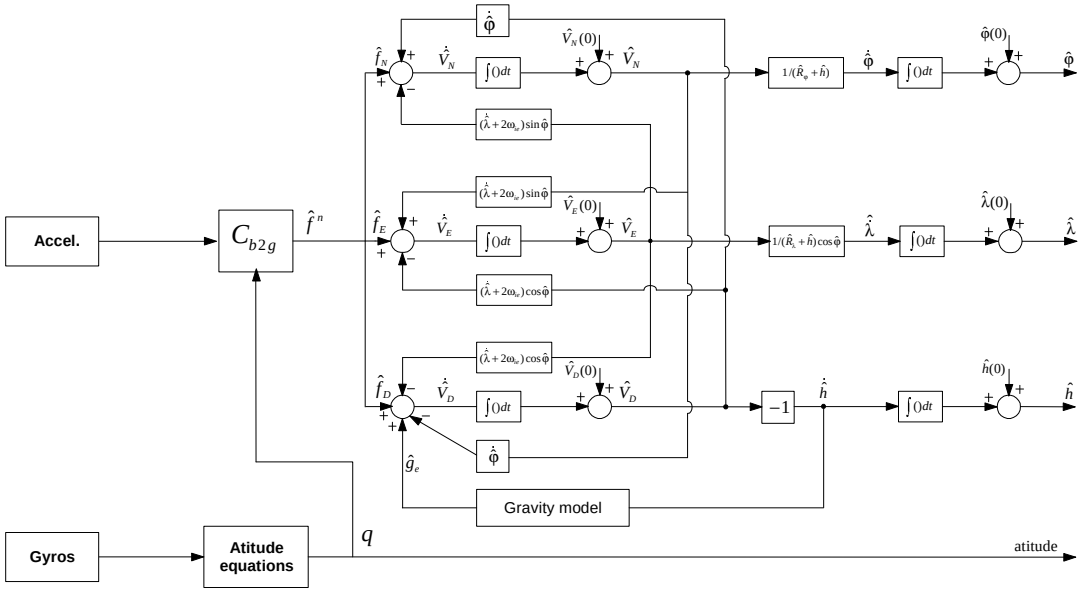

Figure 2.3: IN mechanization diagram

Thus the computed velocity in Navigation frame, is given by

$$\hat{v}^n = \int \dot{\hat{v}}^n dt + \hat{v}^n(0).$$

As the latitude φ, longitude λ and the altitude h are related to the velocity components, these quantities can be calculated as follows.

$$\dot{\hat{\varphi}} = \frac{\hat{V}_N}{(\hat{R}_\varphi + \hat{h})}$$

$$\dot{\hat{\lambda}} = \frac{\hat{V}_E}{(\hat{R}_\lambda + \hat{h})\cos(\hat{\varphi})}$$

$$\dot{\hat{h}} = -\hat{V}_D$$

where, R_φ represents the meridian radius in meters, R_λ is the normal radius also in meters. The latitude, longitude and altitude are obtained by integrating the above equations with the suitable initial conditions:

$$\hat{\varphi} = \int \dot{\hat{\varphi}} dt + \hat{\varphi}(0)$$

$$\hat{\lambda} = \int \dot{\hat{\lambda}} dt + \hat{\lambda}(0)$$

$$\hat{h} = \int \dot{\hat{h}} dt + \hat{h}(0)$$

The attitude equations given in quaternions can be used which integrate the angular rates vector $[\ p \quad q \quad r]^T$ and are given by the following relation:

$$\dot{\hat{q}}_0 = -\frac{1}{2}(\hat{q}_1 p + \hat{q}_2 q + \hat{q}_3 r)$$

$$\dot{\hat{q}}_1 = \frac{1}{2}(\hat{q}_0 p + \hat{q}_2 r - \hat{q}_3 q)$$

$$\dot{\hat{q}}_2 = \frac{1}{2}(\hat{q}_0 q + \hat{q}_3 p - \hat{q}_1 r)$$

$$\dot{\hat{q}}_3 = \frac{1}{2}(\hat{q}_0 r + \hat{q}_1 q - \hat{q}_2 p)$$

But as we have mentioned before in section 2.1, for very short term navigation as required for many tactical UAV applications and missions further simplifications to the system mechanization may be permitted.

For instance, when the navigation period is short the effects of the rotation of the Earth on the attitude computation process can sometimes be ignored. Coriolis corrections are no longer essential in the velocity equations to give sufficiently accurate navigation [Titterton, 1997]. In this case we can neglect the Earth curvature and the Navigation frame can be considered as inertial frame.

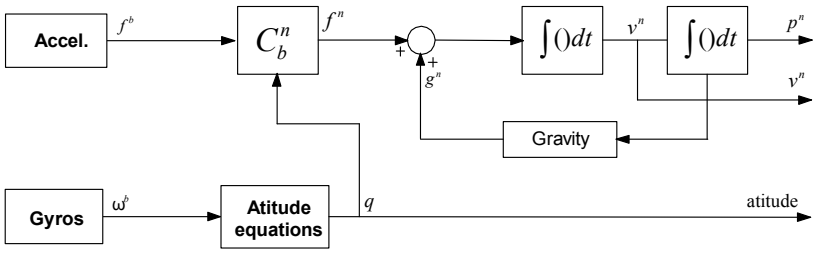

Figure 2.4: IN mechanization diagram - Navigation frame

In this case the navigation equation can have the following form (including the attitude equations):

$$\begin{aligned}\dot{\underline{p}}^n &= \underline{v}^n \\ \dot{\underline{v}}^n &= C_b^n \underline{f}^b + \underline{g}^n \\ \dot{\underline{q}}_b^n &= \tfrac{1}{2}\Omega \underline{q}_b^n\end{aligned}$$

These equations modeled discretely take the following form:

$$\begin{bmatrix} p_k^n \\ v_k^n \\ q_k \end{bmatrix} = \begin{bmatrix} v_{k-1}^n \Delta t + p_{k-1}^n \\ [C_b^n f_k^b + g^n]\Delta t + v_{k-1}^n \\ \frac{1}{2}\Omega q_{k-1}\Delta t + q_{k-1} \end{bmatrix}$$

2.4 Sources of Error in Inertial Navigation System

IN presents many drawbacks, which complicate the usage as alone navigation system. Error in the calculated navigation system state can arise from four main sources:

1. **Instrumentation Errors**: The sensed variables may not equal the physical quantities because of imperfections in the sensors (e.g., bias, scale factor, nonlinearity, and random noise).

2. **Computational Errors**: The navigation equations are typically implemented by a digital computer. Quantization, overflow, and numeric (e.g., integration) errors can occur.

3. **Alignment Errors**: The sensors and their platform cannot be aligned perfectly with their assumed directions.

4. **Environment Errors**: The environment cannot be modeled exactly and affects compensation of the measurements.

All these errors make the IN system diverging slowly from the real measurements with time. The rate at which navigation errors grow over long periods of time is governed predominantly by the accuracy of the initial alignment, imperfections in the inertial sensors used by the system, and the dynamics of the trajectory followed by the vehicle. To illustrate Table 2.2 is showing the quadratic effect of the accelerometer bias on the position [Walchko, 2002].

$$error = \frac{1}{2} bias \cdot t^2$$

Hence, sensor biases can be included as part of the state vector of the IN state

Bias m/s^2	Error (m) t=100 sec	Error(m) t=30 min
0.1	500	162000
0.01	5	16200
0.001	0.5	1620
0.0001	0.05	162

Table 2.2: Quadratic effect of the accelerometer bias on the position

vector and estimated along with the position, velocity and attitude. As shown in [Franklin et al., 2000], [Hong and L. M. Chun, 2005] in order to estimate the biases (disturbances), we must provide a model of how they behave.

If we assume the bias is constant, the model is quite simple:

$$\dot{\alpha}^a = 0$$
$$\dot{\alpha}^g = 0$$

where α^a is accelerometer bias, α^g is qyro bias. But, we could extend the idea to other kinds of behavior, for example as shown in [Farrell et al., 2000] we can model the biases as random walk processes. The random walk process better known as Wiener or Brownian-motion process is implemented by integrating the output of white noise source. Then for the accelerometer and gyro bias we have:

$$\dot{\alpha}^a = \omega^a(t)$$
$$\dot{\alpha}^g = \omega^g(t)$$

where the covariance's for the driving white noise processes ω^a and ω^g can be obtained by analysis of the instrument biases over extended period of time. In actuality, α^a and α^g represent a composite of accelerometers and gyro errors, more in [Farrell et al., 2000]. We need to be aware that the bias model needs to have different values for each accelerometers and gyros.

2.5 Aided Inertial Navigation System

To make up for the inertial measurement drifts, an alternative approach, consisting in employing some additional source of navigation information, external to the inertial system to improve the accuracy of the inertial system, is needed. For example GPS Aided IN today is well understood and widely implemented on many aerial and ground vehicles. The advantages and disadvantages of the GPS and IN systems make them complementary, and the best estimates of the position, velocity and attitude can be obtained by combining both GPS and IN measurements using one of the following Aided Inertial Navigation Filter Configuration: Within an integrated navigation system, the filter can be configured either as a *direct* or *indirect* form depending on the types of sensors and the complexities of the system [Kim, 2004].

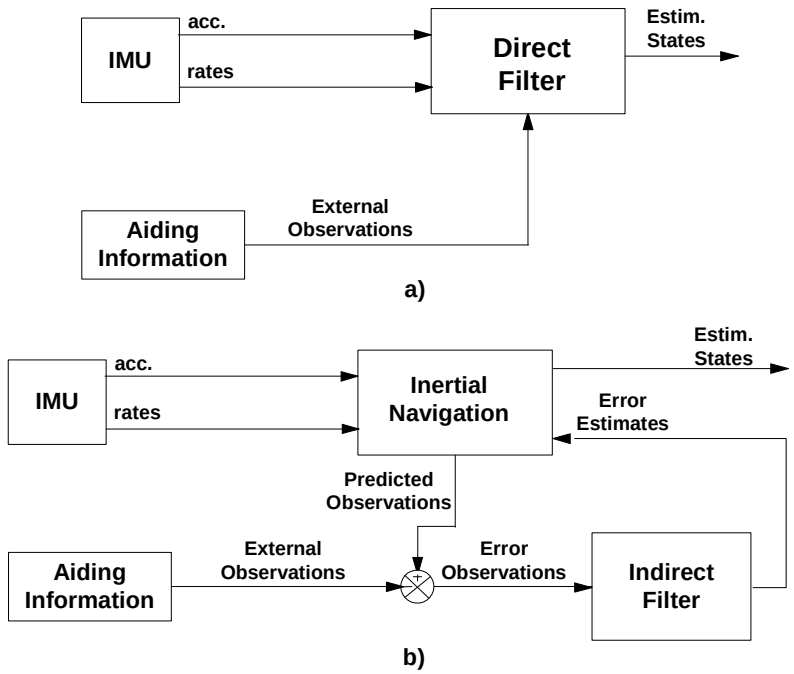

Figure 2.5: (a) The direct filter configuration and (b) the indirect filter

In a direct configuration, the filter directly estimates the states of interest. It typically constitutes a main functional block within the system performing both the Inertial Navigation equations and the observation fusion. In the indirect formulation, the filter estimates the error quantities of the states, and applies this error to the external Inertial Navigation equations loop for correction, hence it estimate the state indirectly. By dealing with the error quantities, the filter can now be decoupled from the main loop and can operate in a complementary fashion.

Chapter 3

Vision Sensors

Vision sensors are nowadays considered essential for UAV navigation, firstly because they are cost effective, small, compact and reliable passive sensors and secondly because they are capable of providing a rich source of information about the vehicle environment.

3.1 Single Camera

The single camera as projective sensor is an ideal hardware setup for aerial vehicles as it has small weight and can be easily mounted on the vehicle. A major drawback of a single camera is the absence of depth information i.e. relative range to the map points, see figure (3.1). By performing motion i.e. taking measurements of the map points from two or more camera positions, depth information can be estimated. This technique correlates the depth information with the camera motion and raises the issue of great uncertainty in the depth information. It is possible that the growth of uncertainty in the camera motion is quite rapid and this large uncertainty is reflected in the depth estimates that also become uncertain. Since vectors have power to simplify geometrical problems we have used vectors to explain the vision sensors geometry. With reference to figure (3.1), when the vehicle moves, the camera observes the map point M with position vector $\underline{m}$, repeatedly as it translates through the camera image. At each image, the camera is capturing a projection ray from the map point M to the perspec-

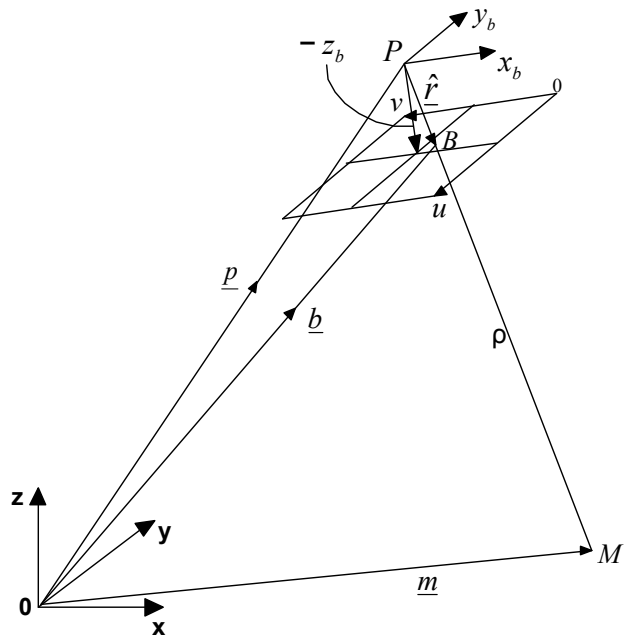

Figure 3.1: Single camera (vectors application)

tive center P of the camera with position vector $\underline{p}$. Each projection ray from the camera is represented by a line $\rho\underline{\hat{r}}$. Instead of using angles such as bearing or azimuth to the map points we use unit direction vectors $\underline{\hat{r}}$ to the map points. This approach is more efficient in terms of computational speed and accuracy, allows us to make important trigonometric function substitutions in the vision sensors geometry, and addresses the problem of singularities in the trigonometric functions at certain points. It is assumed that the single camera is mounted in such a way that the camera frame $(P; x_b, y_b, z_b)$ is at (or very near to) the center of gravity of the vehicle and is aligned with the vehicle body frame $(V; x_b, y_b, z_b)$. The origin of the camera frame is at the perspective center P of the camera i.e. the center of gravity of the vehicle. The x_b axis is defined in forward direction and is perpendicular to the horizontal component u of the camera image. The z_b axis is perpendicular to x_b axis and is equal to the focal length of the camera. The negative sign appears when z_b axis is pointing to the center of the camera image. The y_b axis completes the right-handed orthogonal coordinate system and is perpendicular to the vertical component v of the camera image, see figure (3.1). In the camera (body) frame we measure the camera vector r_b. The unit vector

in the direction r_b notated as $\underline{\hat{r}_b}$ may be evaluated as:

$$\underline{\hat{r}_b} = \frac{\underline{r_b}}{|\underline{r_b}|} \tag{3.1}$$

Unit vector $\underline{\hat{r}_b}$ can be transformed in the navigation frame $(0; x, y, z)$ with the transformation matrix that transforms body frame coordinates into navigation frame coordinates C_b^n, equation (2.5):

$$\underline{\hat{r}} = C_b^n \underline{\hat{r}_b} \tag{3.2}$$

As we can see from equations (3.1) and (3.2) the use of unit direction vectors as measurements to the map points involves a non linear transformation from the fundamental camera vector. This transformation will introduce correlations between the errors on the components of the unit direction vector. Both the transformation matrix in equation (3.2) and the normalizing term in equation (3.1) are with very small quantities and the errors on successive measurements can be omitted and treated as approximately uncorrelated. This allows the measurement covariance matrix to be considered as diagonal matrix. This assumption in most practical cases allows fast processing of the measurements and is saving a significant amount of processing time [Wade and Grewal, 1988].

3.2 Stereo Camera

For the stereo camera setup, a second camera is mounted with known offset D from the first. We assume no angular offset from the vehicle body frame. The origin of the second camera frame is at the perspective center P' of the second camera with position vector p', see figure (3.2). With a stereo camera, we can estimate the map point positions and the range to the map points. These two facts are the major advantages of the stereo camera over the single camera. The range vector $\underline{\rho}$ being the difference between the map point vector $\underline{m}$ and the position vector $\underline{p}$ may be written as

$$\underline{\rho} = \underline{m} - \underline{p} \tag{3.3}$$

From the definition of vector product, the minimum distance d from a point to

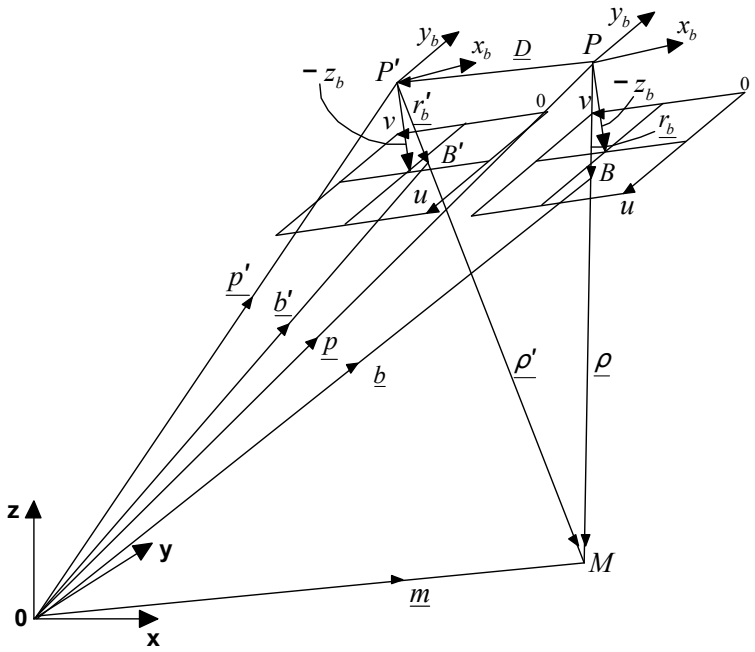

Figure 3.2: Stereo camera (vectors application)

a line is (Appendix A):

$$d = \left|(\underline{m} - \underline{p}) \times \hat{r}\right| \tag{3.4}$$

Using the stereo camera we have measurements of the map point M from the positions P and P' each represented by a line $(\underline{p} + \rho\underline{\hat{r}})$ and $(\underline{p'} + \rho'\underline{\hat{r}'})$. For a single measurement, let us choose an error weighting function of the magnitude of the minimum distance d squared:

$$e = \left|(\underline{m} - \underline{p}) \times \underline{\hat{r}}\right|^2 \tag{3.5}$$

Expanding equation (3.5)

$$e = \underline{m}^T R\underline{m} - 2\underline{m}^T R\underline{p} + \underline{p}^T R\underline{p} \tag{3.6}$$

where $R = [(\underline{\hat{r}}^T\underline{\hat{r}})I - 0.5(\underline{\hat{r}}\underline{\hat{r}}^T + \underline{\hat{r}}\underline{\hat{r}}^T)]$. However, since $\underline{\hat{r}} = [\ a \quad b \quad c\ ]^T$ is unit length:

$$R = [I - \underline{\hat{r}}\underline{\hat{r}}^T] = \begin{bmatrix} (1-a^2) & -ab & -ac \\ -ab & (1-b^2) & -bc \\ -ac & -bc & (1-c^2) \end{bmatrix} \tag{3.7}$$

For multiple measurements the mean error is given by:

$$E = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{3.8}$$

Since $\underline{m}$ is constant:

$$E = \underline{m}^T \frac{1}{n}\sum_{i=1}^{n}(R)_i\underline{m} - 2\underline{m}^T\frac{1}{n}\sum_{i=1}^{n}(R\underline{p})_i + \frac{1}{n}\sum_{i=1}^{n}(\underline{p}^T R\underline{p})_i$$

$$E = \underline{m}^T A\underline{m} - 2\underline{m}^T\underline{b} + \underline{c}$$

Differentiating with respect to $\underline{m}$

$$\frac{dE}{d\underline{m}} = 2A\underline{m} - 2\underline{b} \tag{3.9}$$

where $A = \frac{1}{n}\sum_{i=1}^{n}(R)_i$, $\underline{b} = \frac{1}{n}\sum_{i=1}^{n}(R\underline{p})_i$, $\underline{c} = \frac{1}{n}\sum_{i=1}^{n}(\underline{p}^T R\underline{p})_i$. Equating equation (3.9) and solving for $\underline{m}$ estimates the map point position vector

$$\underline{m} = A^{-1}\underline{b} \tag{3.10}$$

We use this method for estimation of the range vector ρ. Using the measured vectors $\underline{r_b}$, $\underline{r'_b}$ we calculate the two unit direction vectors:

$$\underline{\hat{r}_b} = \frac{\underline{r_b}}{\underline{|r_b|}}$$

and

$$\underline{\hat{r}'_b} = \frac{\underline{r'_b}}{\underline{\left|r'_b\right|}}.$$

With the two unit direction vectors $\hat{\underline{r}}_b = [\ a_1 \quad b_1 \quad c_1\ ]^T$ and $\hat{\underline{r}}'_b = [\ a_2 \quad b_2 \quad c_2\ ]^T$ using equation (3.7), we calculate R_1 and R_2 matrices. The matrix A is then calculated:

$$A = \frac{1}{2}(R_1 + R_2)$$

If we let the position vector to be $\underline{p} = 0$ and accordingly $\underline{p'} = \underline{p} + \underline{D}$ i.e. $\underline{p'} = \underline{D}$ we can calculate the body frame components $[\ \rho_{bx} \quad \rho_{by} \quad \rho_{bz}]^T$ of the relative range ρ_b to the map point. From the vectors $\underline{b}_1 = R_1\underline{p}$ and $\underline{b}_2 = R_2\underline{p'}$ we calculate the vector $\underline{b}$

$$\underline{b} = \frac{1}{2}(\underline{b}_1 + \underline{b}_2)$$

Applying equation (3.10) gives the body frame components of the relative range $\underline{\rho_b}$

$$\underline{\rho_b} = A^{-1}\underline{b} \tag{3.11}$$

The navigation frame components $[\ \rho_x \quad \rho_y \quad \rho_z]^T$ of the relative range vector $\underline{\rho}$ can be calculated as:

$$\underline{\rho} = C_b^n \underline{\rho_b} \tag{3.12}$$

The range $\underline{\rho}$ in the navigation frame is

$$|\rho| = C_b^n \underline{\rho_b} \tag{3.13}$$

As with the single camera for the stereo camera the use of range components as measurements to the map points involves a non linear transformation from the fundamental camera vectors which will introduce correlations between the errors on the range components. As with the single camera we assume that these quantities can be omitted and consider the measurement covariance matrix as diagonal matrix. The stereo camera has drawbacks however, the main one being the limited estimability range [J.Sola. et al., 2008]. The stereo camera can provide reasonably good map point position estimates to a limited range. These estimates depend on the camera offset $\underline{D}$ (figure (3.2). The solution seems to be to increase the offset. This will increase the estimability range but will add mechanical complexity to the sensor. Large camera offsets are typically used for

aerial vehicles. The feature matching in these stereo camera setups becomes a challenging problem. In this thesis, we assume that all the image processing and feature matching are solved and that they are provided as pixel values of the map points in the camera images and that the errors in the camera images act as zero mean white noise with known covariance.

Chapter 4

Sensor Fusion

Sensor fusion can be regarded as a method for integrating information from various sensors. The idea of integration is to take advantage of the complementary strengths of the sensors and to obtain the best estimate for a dynamic system's states. Sensor fusion algorithms are particularly useful in low-cost UAV applications, where acceptable performance and reliability is desired, given a limited set of inexpensive sensors [Niculescu, 2001].
The sensor fusion system can provide:

- filtered high-rate navigation and control data for increased performance, estimation of the flight parameters which are not measured directly (i.e. attitude angles, angle-of-attack, sideslip),
- detection of significant changes in aircraft dynamics (i.e. icing, airframe damage),
- ability to replace failed sensor outputs with estimates.

One of the aims of this thesis was to study the applicability of various filtering algorithms which can be applied in sensor fusion like the classic linear Kalman Filter (KF), the natural extension for systems with nonlinear dynamics the Extended Kalman Filter (EKF), Iterated Extended Kalman Filter (IEKF) Unscented Kalman Filter (UKF) and Particle Filters (PF).

4.1 The Problem of Nonlinear Filtering

Nonlinear filtering is a problem of estimating a state of a stochastic dynamic system from noisy measurements. In this thesis the state space approach is used for modeling the dynamic systems with discrete time formulation of the problem. We are adopting the Bayesian approach as general framework for state estimation with assumption that the dynamic system model and the measurement model are available in probabilistic form. Within the Bayesian approach we attempt to construct the posterior probability density function (pdf) of the state based on all the available statistical information including the available measurements. If the system and/or the measurements models are nonlinear then the posterior pdf is non-Gaussian [Ristic et al., 2004]. This posterior pdf can be an answer of what the solution i.e. the estimate of the state should be and with respect to a ceratin criterion the best (optimal) solution can be obtained.
Consider the following discrete-time system model which describes the evolution of the state:

$$x_k = f_k(x_{k-1}, w_k) \tag{4.1}$$

where f_k is a nonlinear function of the state x_{k-1} and w_k is a zero mean process noise sequence. The noisy measurements z_k are related to the state estimate x_k via the following measurements equation:

$$z_k = h_k(x_k, v_k) \tag{4.2}$$

where h_k is a known nonlinear function and v_k is an measurement noise sequence. The noise sequences w_k and v_k are assumed to be white with known pdf and mutually independent. It is assumed that an initial pdf $p(x_0|z_0)$ known as prior is available. From Bayesian perspective the required posterior pdf $p(x_k|z_{1:k})$ can be obtained, recursively, in two stages: prediction and update [Ristic et al., 2004].
Prediction stage gives the prior pdf of the state at time k

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{4.3}$$

where it is supposed that the required pdf $p(x_{k-1}|z_{1:k-1})$ at time $k-1$ is available. The probabilistic model of the evolution of the state $p(x_k|x_{k-1})$ is defined by equation (4.1) and the known statistics of the process noise w_k.
Update stage involves update of the prior pdf using the Bayes' rule

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k} \tag{4.4}$$

where the likelihood function $p(z_k|x_k)$ is defined by equation (4.2) and the known statistics of the measurement noise v_k. This recursive propagation of the posterior pdf is only a conceptual solution and in general cannot be determined analytically. Solutions exist in restrictive set of cases. For a linear-Gaussian case the Kalman filter is being the most popular. Others solutions that approximate the optimal solution such as the Extended Kalman Filters, Unscented Kalman Filter and the Particle filters are possible and will be presented later in this chapter.

4.2 Kalman Filter

The Kalman filter assumes that the posterior density is Gaussian and is completely characterized by its mean vector and covariance matrix. This fact makes the linear filtering problems particularly tractable [Jazwinski, 1970]. Equations (4.1) and (4.2) for the linear dynamic system can be rewritten as:

$$x_k = F_k x_{k-1} + w_k \tag{4.5}$$

$$z_k = H_k x_k + v_k \tag{4.6}$$

where F_k is a linear state transition matrix, H_k is the measurement (observation) matrix which relates the state vector x_k with the measurements z_k. The zero mean white Gaussian sequences w_k and v_k are mutually independent and have covariances Q_k and R_k respectively.
The Kalman filter equations are derived using equations (4.3) and (4.4) with the involved densities being Gaussian:

$$p(x_{k-1}|z_{1:k-1}) = N(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \tag{4.7}$$

$$p(x_k|z_{1:k-1}) = N(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \tag{4.8}$$

$$p(x_k|z_{1:k}) = N(x_k; \hat{x}_{k|k}, P_{k|k}) \tag{4.9}$$

where $N(x; m, P)$ represents a Gaussian density with argument x, mean m and covariance P. The appropriate means and covariances of the Kalman Filter are computed as follows [Ristic et al., 2004], [Jazwinski, 1970]:

Prediction

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \tag{4.10}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{4.11}$$

Update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k \hat{x}_{k|k-1}) \tag{4.12}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{4.13}$$

where the Kalman gain is calculated by

$$K_k = P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R_k]^{-1} \tag{4.14}$$

4.3 Extended Kalman Filter

To estimate the state of a non linear system, a modified form of the Kalman Filter, the Extended Kalman Filter (EKF) is used. EKF approximates (linearizes) the dynamic system model (4.1) and the measurement model (4.2) by the first terms in their Taylor expansion. These equations for the nonlinear dynamic system with additive noise can be rewritten as:

$$x_k = f_k(x_{k-1}) + w_k \tag{4.15}$$

$$z_k = h_k(x_k) + v_k \tag{4.16}$$

Within EKF the posterior density $p(x_k|z_{1:k})$ is approximated as Gaussian and the relationships (4.7) to (4.9) are assumed to apply:

$$p(x_{k-1}|z_{1:k-1}) \approx N(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \tag{4.17}$$

$$p(x_k|z_{1:k-1}) \approx N(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \tag{4.18}$$

$$p(x_k|z_{1:k}) \approx N(x_k; \hat{x}_{k|k}, P_{k|k}) \tag{4.19}$$

The appropriate means and covariances of the EKF are computed as follows [Ristic et al., 2004], [Jazwinski, 1970]:

Prediction

$$\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1|k-1}) \tag{4.20}$$

$$P_{k|k-1} = \nabla f_k^x P_{k-1|k-1} \nabla f_k^{xT} + Q_k \tag{4.21}$$

Update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k[z_k - h_k(\hat{x}_{k|k-1})] \tag{4.22}$$

$$P_{k|k} = P_{k|k-1} - K_k \nabla h_k^x P_{k|k-1} \tag{4.23}$$

where the Kalman gain is calculated by

$$K_k = P_{k|k-1} \nabla h_k^{xT} [\nabla h_k^x P_{k|k-1} \nabla h_k^{xT} + R_k]^{-1} \tag{4.24}$$

The Jacobian ∇f_k^x of the nonlinear function f_k is with respect to the state x_k and is evaluated at $\hat{x}_{k-1|k-1}$. Similar the Jacobian ∇h_k^x of the nonlinear function h_k is with respect to the state x_k and is evaluated at $\hat{x}_{k|k-1}$. They are the local linearizations of the nonlinear functions and serve as satisfactory description of the nonlinearities.

4.4 Iterated Extended Kalman Filter

Iterated Extended Kalman Filter (IEKF) is largely based on EKF. IEKF presents an algorithm which is extremely effective in accounting for measurement nonlinearities [Jazwinski, 1970]. It is known as local iteration algorithm. By local iteration, we mean iteration at a point t_k, or on an interval $[t_{k-1}, t_k]$. The purpose of the iterations is to improve the reference trajectory, and thus the estimate, in the presence of significant nonlinearities. Since the iteration is local, the recur-

sive filter structure is retained; new estimates are computed as new observations become available. The IEKF uses the EKF equations (4.22) replaced by:

$$\eta_{i+1} = \hat{x}_{k|k-1} + K_k[z_k - h_k(\eta_i) - \nabla h_k^{\eta_i}\{\hat{x}_{k|k-1} - \eta_i\}] \qquad i = 1, \ldots, l \tag{4.25}$$

$$\hat{x}_{k|k} = n_l \tag{4.26}$$

The iteration starts with $\eta_1 = \hat{x}_{k|k-1}$, and terminates when there is no significant difference between consecutive iterates. Note that the Kalman gain K_k, equation (4.24) is revaluated on each iteration, as are the measurement function h_k and the matrix $\nabla h_k^{\eta_i}$ of the nonlinear function h_k with respect to n_i. The last iterate, say n_l, is taken for the estimate $\hat{x}_{k|k}$.

4.5 Unscented Kalman Filter

Numbers of significant problems appear when implementing EKF. The first is the need to analytically evaluate the Jacobian matrices of the dynamic system model and the measurement model. The Jacobian is not guaranteed to exist, or might not have a finite value. Further, there can be considerable implementation difficulties when the system is composed of many states and is highly non-linear. Finally, the assumption that the second and higher-order errors might not be negligible and linearization can introduce significant errors [Julier, 1997]. The Unscented Kalman Filter (UKF) addresses the approximation issues of the EKF. It does not approximate the nonlinear functions instead it approximates the posterior density $p(x_k|z_{1:k})$ as a Gaussian density but now specified using a minimal set of carefully chosen sample points called sigma points. These sample points completely capture the true mean and covariance of the Gaussian density, and when propagated through the true nonlinear system, capture the posterior mean and covariance accurately to the second order (Taylor series expansion) for any nonlinearity [Haykin, 2001].

4.5.1 The Unscented Transformation

The unscented transformation is a method that calculates the statistics of a random variable which undergoes nonlinear transformation [Julier, 1996]. Given $n-$dimensional random variable x_{k-1} with mean $\hat{x}_{k-1|k-1}$ and covariance $P_{k-1|k-1}$ is propagated through a nonlinear function $\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1})$. The calculation of the statistics of the random variable $\hat{x}_{k|k-1}$ is made by $2n+1$ weighted samples (sigma points) that are selected by the following algorithm:

$$\begin{aligned} \chi^0_{k-1|k-1} &= \hat{x}_{k-1|k-1} \\ \chi^i_{k-1|k-1} &= \hat{x}_{k-1|k-1} + (\sqrt{(n+\lambda)P_{k-1|k-1}})_i \\ \chi^{i+n}_{k-1|k-1} &= \hat{x}_{k-1|k-1} - (\sqrt{(n+\lambda)P_{k-1|k-1}})_i \end{aligned} \tag{4.27}$$

and the associated weights

$$W^m_0 = \lambda/(n+\lambda), W^c_0 = \lambda/(n+\lambda) + (1-\alpha^2+\beta)$$

$$W^m_0 = W^c_0 = 1/\{2(n+\lambda)\}, \quad i = 1, \ldots, 2n$$

where the parameter λ is a scaling parameter defined as $\lambda = \alpha^2(n+\kappa) - n$. The constant α determines the spread of the sigma points around $\hat{x}_{k-1|k-1}$, and is usually set to small positive value (e.g. $1 \leq \alpha \leq 10^{-4}$). The constant κ is a secondary scaling parameter, which is usually set to $3-n$, and β is used to incorporate prior knowledge of the distribution (for Gaussian distributions, $\beta = 2$ is optimal). $(\sqrt{(n+\lambda)P_{k-1|k-1}})_i$ is the i-th column of the matrix square root [Haykin, 2001]. Given the set of samples generated by equation (4.27) each sigma point is instantiated through the process model to yield a set of transformed samples:

$$\chi^i_{k|k-1} = f[\chi^i_{k-1|k-1}, u_k] \qquad i = 1, \ldots, 2n \tag{4.28}$$

The mean and covariance are computed as

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2n} W^m_i \chi^i_{k|k-1} \tag{4.29}$$

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^c \left\{\chi_{k|k-1}^i - \hat{x}_{k|k-1}\right\} \left\{\chi_{k|k-1}^i - \hat{x}_{k|k-1}\right\}^T \tag{4.30}$$

and the cross-covariance as

$$P_{k|k-1}^c = \sum_{i=0}^{2n} W_i^c \left\{\chi_{k-1|k-1}^i - \hat{x}_{k|k-1}\right\} \left\{\chi_{k|k-1}^i - \hat{x}_{k|k-1}\right\}^T \tag{4.31}$$

The mean and covariance are directly calculated using standard vector and matrix operations. This means that the algorithm can be applied to almost any choice of process model, and the "implementation overhead" is low because it is not necessary to evaluate Jacobians or any other derivatives. The method has a further advantage: it yields more accurate predictions than those determined through linearization [Julier, 1997]. The unscented transform described above can be written conveniently in matrix form as follows:

$$\chi_{k-1|k-1} = \left[\begin{array}{ccc} \hat{x}_{k-1|k-1} & \cdots & \hat{x}_{k-1|k-1} \end{array}\right] + \sqrt{c}\left[\begin{array}{ccc} 0 & \sqrt{P_{k-1|k-1}} & -\sqrt{P_{k-1|k-1}} \end{array}\right]$$

$$\hat{\chi}_{k|k-1} = f[\chi_{k-1|k-1}, u_k]$$

$$\hat{x}_{k|k-1} = \hat{\chi}_{k|k-1} w_m$$

$$P_{k|k-1} = \hat{\chi}_{k|k-1} W \hat{\chi}_{k|k-1}^T$$

$$P_{k|k-1}^c = \chi_{k-1|k-1} W \hat{\chi}_{k|k-1}^T$$

where $\chi_{k-1|k-1}$ is the matrix of sigma points, function $f(\cdot)$ is applied to each column of the argument matrix separately, $c = \alpha^2(n + \kappa)$, and vector w_m and matrix W are defined as follows [Sarkka, 2006]:

$$w_m = \left[\begin{array}{ccc} W_o^m & \cdots & W_{2n}^m \end{array}\right]^T$$

$$W = \left(I - \left[\begin{array}{ccc} w_m & \cdots & w_m \end{array}\right]\right) \times \mathrm{diag}\left(W_0^c \cdots W_{2n}^c\right) \times \left(I - \left[\begin{array}{ccc} w_m & \cdots & w_m \end{array}\right]\right)^T.$$

4.5.2 Unscented Kalman Filter

Using the matrix form of the unscented transform described above the prediction and update steps of the Unscented Kalman filter (UKF) where the noises are additive can be computed as follows:

Prediction:

$$\chi_{k-1|k-1} = [\ \hat{x}_{k-1|k-1} \quad \cdots \quad \hat{x}_{k-1|k-1}\ ] + \sqrt{c}[\ 0 \quad \sqrt{P_{k-1|k-1}} \quad -\sqrt{P_{k-1|k-1}}]$$

$$\hat{\chi}^*_{k|k-1} = f[\chi_{k-1|k-1}, u_k]$$

$$\hat{x}_{k|k-1} = \hat{\chi}^*_{k|k-1} w_m$$

$$P_{k|k-1} = \hat{\chi}^*_{k|k-1} W \hat{\chi^*_{k|k-1}}^T + Q_k$$

Update:

$$\chi_{k|k-1} = [\ \hat{x}_{k|k-1} \quad \cdots \quad \hat{x}_{k|k-1}\ ] + \sqrt{c}[\ 0 \quad \sqrt{P_{k|k-1}} \quad -\sqrt{P_{k|k-1}}]$$

$$\mathrm{Z}_{k|k-1} = h[\chi_{k|k-1}]$$

$$\hat{z}_{k|k-1} = \mathrm{Z}_{k|k-1} w_m$$

$$P^{zz}_{k|k-1} = \mathrm{Z}_{k|k-1} W \mathrm{Z}^T_{k|k-1} + R_k$$

$$P^{xz}_{k|k-1} = \chi_{k|k-1} W \mathrm{Z}^T_{k|k-1}.$$

Then we compute the filter gain K_k and we update the mean and covariance as follows:

$$K_k = P^{xz}_{k|k-1} {P^{zz}_{k|k-1}}^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\{z_k - \hat{z}_{k|k-1}\}$$

$$P_{k|k} = P_{k|k-1} - K_k P^{zz}_{k|k-1} K_k^T$$

4.6 Particle Filters

The EKF, IEKF and UKF nonlinear filters rely on Gaussian approximations. These techniques show limitations and inconsistencies in the solutions in applications with highly nonlinear models. The more non-Gaussian noise is with the inputs and the measurements and/or the more non-linear the models are, the more significant the problems become.
The current state-of-the-art filtering (estimation) theory looks for reliable and accurate recursive estimation techniques for non-linear and non-Gaussian applications. Very popular are the sequential Monte Carlo methods, known as Particle filters (PF). The central idea of the PF is to represent the required probability density function as a set of random samples (particles), rather then as a function over the state space [Gordon et al., 1993]. As the number of particles becomes very large they provide equivalent representation of the required probability density function. Directly from the particles, estimates of the moments such as the mean and covariance can be obtained.

4.6.1 Introduction to Monte Carlo Techniques

In many scientific problems it is of essential importance to compute the integral

$$I = \int_D g(x)dx, \tag{4.32}$$

where D is often a region in a high dimensional space and $g(x)$ is target function of interest. If we can draw independent and identically distributed (i.i.d.) random samples $x^{(1)}, \ldots, x^{(n)}$ uniformly from D (by a computer), an approximation to I can be obtained as [Liu, 2001]:

$$\hat{I}_n = \frac{1}{n}\{g(x^{(1)}) + \ldots + g(x^{(n)})\} \tag{4.33}$$

The average of many independent random variables with common mean and finite variances tends to stabilize at their common mean that is,

$$\lim_{n\to\infty} \hat{I}_n = I \quad \text{with} \quad \text{probability} \quad \text{of} \quad 1,$$

which is stated and guaranteed by the law of large numbers. Assessment of the convergence rate can be made by the central limit theorem (CLT) [Liu, 2001]:

$$\sqrt{n}(\hat{I}_n - I) \rightarrow N(0, \sigma^2) \quad \text{in} \quad \text{distribution},$$

where $\sigma^2 = \text{var}\{g(x)\}$. Hence the error term of this Monte Carlo approximation is $O(n^{-1/2})$, regardless of the dimensionality of x.

4.6.2 Importance Sampling

The Monte Carlo techniques suffer from wasting a lot of effort in evaluating random samples located in regions where the function value is almost zero. The idea of importance sampling suggests that we should focus on the regions of "importance" so as to save computational resources [Liu, 2001]. This idea of biasing towards "importance" regions is particularly useful for Monte Carlo computation with high dimensional models.

Suppose that $p(x) \propto \pi(x)$ is a probability density function from which it is difficult to draw samples but for which $\pi(x)$ can be evaluated [Doucet et al., 2000], [Arulampalam et al., 2002]. Let $x^{(i)} \approx g(x)$, $i = 1, \ldots, N$ be samples that are drawn from a trial (proposal) density $g(\cdot)$ called importance density function that is similar to $\pi(x)$. Then a weighted approximation to the discrete density $p(x)$ is given by

$$p(x) \approx \sum_{i=1}^{N_s} w^{(i)} \delta(x - x^{(i)}) \tag{4.34}$$

where the normalized importance weight of the $i - th$ sample is

$$w^{(i)} \propto \frac{\pi(x^{(i)})}{g(x^{(i)})} \tag{4.35}$$

If the samples $x_{0:k}^{(i)}$ were generated from the importance density $g(x_{0:k}|z_{1:k})$ then the weights defined with (4.35) become

$$w^{(i)} \propto \frac{p(x_{0:k}^{(i)}|z_{1:k})}{g(x_{0:k}^{(i)}|z_{1:k})} \tag{4.36}$$

4.6.3 Sequential Importance Sampling

The sequential importance sampling uses the strategy of building up the importance density sequentially. In this case suppose that at time $k-1$ we have samples constituting an approximation to $p(x_{0:k-1}|z_{1:k-1})$. We want with the measurements z_k at time k to approximate $p(x_{0:k}|z_{1:k})$ with a set of new samples. If we decompose the importance density such as

$$g(x_{0:k}|z_{1:k}) = g(x_k|x_{0:k-1}, z_{1:k})g(x_{0:k-1}|z_{1:k-1}) \tag{4.37}$$

then we can obtain samples $x_{0:k}^{(i)} \approx g(x_{0:k}|z_{1:k})$ by augmenting each of the existing samples $x_{0:k-1}^{(i)} \approx g(x_{0:k-1}|z_{1:k-1})$ with a new state $x_k^{(i)} \approx g(x_k|x_{0:k-1}, z_{1:k})$. Thus the weight update equation can then be evaluated as [Arulampalam et al., 2002]:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{g(x_k^{(i)}|x_{k-1}^{(i)}, z_k)} \tag{4.38}$$

where the condition $g(x_k|x_{0:k-1}, z_{1:k}) = g(x_k|x_{k-1}, z_k)$ is satisfied and the importance density becomes only depended on x_{k-1} and z_k. Equivalently sequential importance sampling algorithm approximates the density $p(x_k|z_{1:k})$ as

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{4.39}$$

with normalized importance weights defined by:

$$\bar{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}} \tag{4.40}$$

It can be shown that as $N \to \infty$ the approximation given by (4.39) approaches to the true posterior density $p(x_k|z_{1:k})$ [Ristic et al., 2004], [Arulampalam et al., 2002].

The estimated mean value $\hat{x}_k$ and the covariance P_k can be computed using the current state x_k and the weight $\bar{w}_k^{(i)}$ as

$$\hat{x}_k = E\{x_k\} = \sum_{i=1}^{N} \bar{w}_k^{(i)} x_k^{(i)} \tag{4.41}$$

$$\begin{aligned} P_k &= E\left\{[x_k - E\{x_k\}][x_k - E\{x_k\}]^T\right\} \\ &\approx \sum_{i=1}^{N} \bar{w}_k^{(i)} (x_k^{(i)} - \hat{x}_k)(x_k^{(i)} - \hat{x}_k)^T \end{aligned} \tag{4.42}$$

Within the sequential importance sampling algorithm the best possible choice for the importance density function should be the posterior density $p(x_k|z_{1:k})$. However for importance functions of the form (4.37) the variance of the importance weights can only increase over time [Doucet et al., 2000]. This variance increase is known as degeneracy phenomenon or degeneracy problem. It means that after certain number of recursions easily we encounter a situation where almost all but one of the importance weights is very close to zero. This brings large computational effort for updating the trajectories whose contribution to the approximation $p(x_k|z_{1:k})$ is almost zero. The degeneracy problem is unavoidable and can be regarded as problem which was preventing the practical implementations of the PF for many years.

4.6.4 Resampling

The idea behind the resampling method is by measuring the degeneracy of the sequential importance sampling algorithm to remove the samples with very small weights and multiply samples with large weights. This method works on reducing the effects of the degeneracy. A suitable measure of degeneracy of the sequential importance sampling algorithm is the effective sample size N_{eff}, which can be estimated by [Arulampalam et al., 2002]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^{(i)})^2} \tag{4.43}$$

Usually the resampling method in the sequential importance sampling algorithm will not be performed on every time step, but only when needed i.e. when N_{eff} falls bellow some threshold N_T. The resampling step involves generating a new set of samples $\{x_k^{(i)^*}\}_{i=1}^N$ by resampling N times from the approximate discrete representation given by

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{4.44}$$

so that $\Pr(x_k^{(i)^*} = x_k^j) = w_k^j$. The resulting sample is an i.i.d. sample from the discrete density (4.44) and hence the new weights are uniform [Arulampalam et al., 2002]. One efficient algorithm for the resampling, which is easy to implement and is with $O(N)$ complexity is the systematic resampling scheme [Ristic et al., 2004].

4.6.5 Generic Particle Filter

The sections from 4.5.3 to 4.5.5 are presenting the building blocks of the Generic Particle Filter (GPF). Its pseudocode is given on Table 4.1.

$$[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^N] = PF[\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N, z_k]$$

1. Filtering via sequential importance sampling algorithm (section 4.5.4)
   $$[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^N] = SIS[\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N, z_k]$$
2. Calculate $\hat{N}_{eff}$ using equation (4.42)
3. *IF* $\hat{N}_{eff} < N_T$
   Resample (section 4.5.5)
   $$[\{x_k^{(i)}, w_k^{(i)}, -\}_{i=1}^N] = RESAMPLE[\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^N]$$
4. *END IF*

Table 4.1: Generic Particle Filter

The sequential importance sampling algorithm is the basis for most of the PF that are proposed in the literature. Many special cases of PF may be treated

as different in a sense that different important density function is chosen and/or there is a modification in the methodology for reducing the effects of degeneracy. Presenting all the versions of the PF is out of the scope of this thesis. For more detailed coverage of the PF please refer to [Ristic et al., 2004], [Doucet et al., 2001], [Arulampalam et al., 2002], [Daum, 2005].

4.6.6 Rao-Blackwellized Particle Filter

The Rao-Blackwellization method recommends that we carry out analytical computations as much as possible. For number of applications with high dimensional state vectors the implementation of PF becomes problematic. This is a case because for high dimensional state vectors we need bigger number of samples to cover the state space efficiently. Rao-Blackwellized Particle Filter (RBPF) addresses this issue and splits the state vector into two parts, one part x_k^p which is estimated using the PF and other part x_k^x which is estimated in a closed form for example with the Kalman-like filtering algorithms. RBPF uses the following factorization of the posterior distribution of the state vector,

$$p(x_{1:k}^p, x_k^k | z_{1:k}) = p(x_k^k | x_{1:k}^p, z_{1:k}) p(x_{1:k}^p | z_{1:k}) \tag{4.45}$$

which follows from the Bayes' rule. The measurements $z_{1:k} = \{z_1, \ldots, z_k\}$ are up to time k. When the term $p(x_k^k | x_{1:k}^p, z_{1:k})$ of equation (4.45) is linear Gaussian, the KF will be used for the estimation, whereas for the nonlinear non-Gaussian part $p(x_{1:k}^p | z_{1:k})$ PF will be used. The interpretation is that the KF is associated with each of the samples in the PF. This gives mixed state space representation with x_k^p represented with samples and x_k^x represented with a KF for each sample [Hendeby et al., 2010].

4.6.7 Local Linearization Particle Filter

One of the most critical design issues of the PF is the choice of importance density $g(x_k | x_{k-1}^{(i)}, z_k)$. We are looking for importance functions which are minimizing the variance of the importance weights. Ideally it should be an optimal importance density function conditioned upon $x_{k-1}^{(i)}$ and the z_k. In general this is possible in

some special cases and class of models, for example when x_k is a member of a finite set or in a case of linear observation model where the optimal importance density is Gaussian [Ristic et al., 2004]. The suboptimal methods that approximate the optimal importance density are commonly used. The most popular suboptimal choice is the transition prior,

$$g(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)}) \tag{4.46}$$

For an additive zero mean Gaussian process noise model the transition prior is:

$$p(x_k|x_{k-1}^{(i)}) = N(x_k; f(x_{k-1}^{(i)}), Q_{k-1}), \tag{4.47}$$

Substitution of (4.46) into (4.38) then yields

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|x_k^{(i)}) \tag{4.48}$$

Note that when the optimal importance function is used, the importance weights can be computed before the particles are propagated to time k. Equation (4.48) states that this is not possible with the transition prior [Ristic et al., 2004].
Other popular suboptimal method for approximation of the optimal importance density is the local linearization method [van der Merwe et al., 2000]. It incorporates the most current observation with the optimal Gaussian approximation of the state based on the EKF or the UKF. The idea is to use for each particle with index$-i$ a separate EKF(i) or UKF(i) to generate a Gaussian importance density, that is

$$g(x_k^{(i)}|x_{k-1}^{(i)}, z_k) = N(x_k^{(i)}; \hat{x}_k^{(i)}, \hat{P}_k^{(i)}), \tag{4.49}$$

where $\hat{x}_k^{(i)}$ and $\hat{P}_k^{(i)}$ are estimates of the mean and covariance computed by EKF(i) or UKF(i) at time k using measurements z_k. This Particle Filter is known as Local Linearization Particle Filter (LLPF). Its pseudocode is given on Table 4.2. Use of the UKF for the local linearization makes the Unscented Particle Filter (UPF). This method preforms the resampling at every time step and therefore the

importance weights are not passed from one iteration to the next (this is similar to the Sampling Importance Resampling Filter or Bootstrap Filter [Gordon et al., 1993]). The use of UKF for the local linearization instead of the EKF is reported to improve the performance [Ristic et al., 2004], [van der Merwe et al., 2000].

$$[\{x_k^{(j)}, P_k^{(j)}\}_{j=1}^N] = LLPF[\{x_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^N, z_k]$$

1. FOR $i = 1:N$
   Run UKF (section 4.4.2)
   $[\hat{x}_k^{(i)}, \hat{P}_k^{(i)}] = UKF[x_{k-1}^{(i)}, P_{k-1}^{(i)}, z_k]$
   Draw a sample from the importance density given by equation (4.48)
   $x_k^{(i)} \approx N(x_k^{(i)}, \hat{x}_k^{(i)}, \hat{P}_k^{(i)})$
   Calculate $\tilde{w}_k^{(i)} = \dfrac{p(z_k \mid x_k^{(i)}) p(x_k^{(i)} \mid x_{k-1}^{(i)})}{g(x_k^{(i)} \mid x_{k-1}^{(i)}, z_k)}$, where $g(x_k^{(i)} \mid x_{k-1}^{(i)}, z_k)$ is given by (4.48).
   END FOR

2. Calculate total weight $t = SUM[\{\tilde{w}_k^{(i)}\}_{i=1}^N]$

3. FOR $i = 1:N$
   Normalize the weghts $w_k^{(i)} = t^{-1}\tilde{w}_k^{(i)}$
   END FOR

4. Resample (section 4.5.5)
   $$[\{x_k^{(j)}, -, i^{(j)}\}_{j=1}^N] = RESAMPLE[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^N]$$

5. FOR $j = 1:N$
   Assign covariance $P_k^{(j)} = \hat{P}_k^{(i)(j)}$
   END FOR

Table 4.2: Local Linearization Particle Filter (Unscented Particle Filter)

Chapter 5

Simultaneous Localization and Mapping (SLAM)

The Simultaneous Localization and Mapping (SLAM) problem asks if it is possible for an autonomous vehicle to start in an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute the vehicle position. The SLAM problem also can be found abbreviated as Concurrent Mapping and Localization (CML) problem in the literature. It presents one of the most fundamental problems in robotics today [Thrun et al., 2005].

5.1 Formulation of the SLAM Problem

In SLAM the vehicle is starting at an unknown location and is moving through an environment with population of landmarks[1]. Assume that the absolute positions of the landmarks are not available. The vehicle is equipped with a sensor which gives relative measurements between any individual landmark and the vehicle itself, as shown on figure (5.1).

[1] the terms landmark and map point will be used synonymously in this thesis

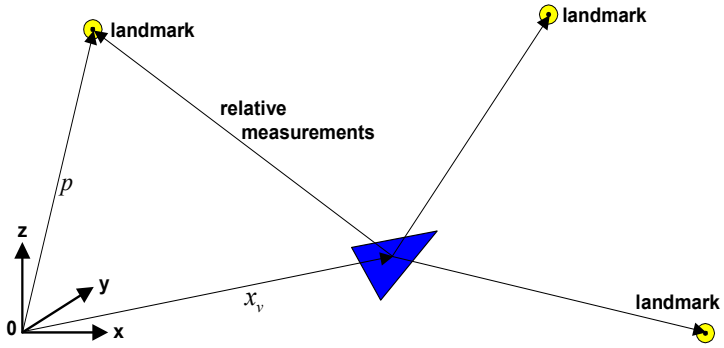

Figure 5.1: The vehicle taking relative measurements to the landmarks

At time instant k, the following quantities are defined:

- x_k^v - vehicle state vector
- u_k - the control input
- m_i - vector describing the position of the i-th landmark whose true position is assumed time invariant
- z_{ik} - observation taken from the vehicle of the position of the i-th landmark at time k.
- $m_k = [\ m_1 \quad m_2 \quad \cdots \quad m_n\ ]^T$ - map vector of the landmarks in the environment systems.

From a probabilistic perspective the SLAM problem involves estimating the posterior probability density function (pdf) of the vehicle state x_k^v along with the map m, given the measurements $z_{1:k}$ and control inputs $u_{1:k}$ up to time k:

$$p(x_k^v, m|z_{1:k}, u_{1:k}) \tag{5.1}$$

In general a recursive solution of the SLAM problem is desirable. Suppose that the required pdf $p(x_{k-1}^v, m|z_{1:k-1}, u_{1:k-1})$ at time $k-1$ is available. From Bayesian perspective the posterior pdf $p(x_k^v, m|z_{1:k}, u_{1:k})$ can be obtained in two stages: prediction and update.

Prediction

$$p(x_k^v, m|z_{1:k-1}, u_{1:k}) = \int p(x_k^v|x_{k-1}^v, u_k)p(x_{k-1}^v, m|z_{1:k-1}, u_{1:k-1})dx_{k-1} \tag{5.2}$$

Update

$$p(x_k^v, m|z_{1:k}, u_{1:k}) = \frac{p(z_k|x_k^v, m)p(x_k^v, m|z_{1:k-1}, u_{1:k})}{\int p(z_k|x_k^v, m)p(x_k^v, m|z_{1:k-1}, u_{1:k})dx_k} \tag{5.3}$$

Equation (5.2) and (5.3) provide the conceptual solution of the SLAM problem. This recursive solution is a function of the probabilistic model of the evolution of the state (vehicle model) $p(x_k^v|x_{k-1}^v, u_k)$ and the likelihood function (observation model) $p(z_k|x_k^v, m)$. The most common solution which represents the vehicle model and the observation model in a state-space form with additive Gaussian noise is the Extended Kalman Filter (EKF). Other alternative solution uses the Rao-Blackwellized Particle Filter (RBPF) and is known as FastSLAM algorithm. This solution describes the vehicle model with a set of samples (trajectory particles). After updating these samples it builds the corresponding map in closed form i.e. analytically. Newer solutions with much potential have been proposed including the use of the information state form [Durrant-White and Bailey, 2006].

5.2 Kalman Filter Solutions to the SLAM Problem

Kalman Filter solutions to the SLAM problem describe the probabilistic model of the evolution of the state (vehicle model) $p(x_k^v|x_{k-1}^v, u_k)$ as

$$x_k^v = f(x_{k-1}^v, u_k) + w_k \tag{5.4}$$

where $f(\cdot)$ describes the motion of the vehicle through the environment and w_k is the process noise being zero mean white Gaussian sequence with covariance Q_k. The observation model $p(z_k|x_k^v, m)$ is described in the form

$$z_k = h_k(x_k^v) + v_k \tag{5.5}$$

where $h(\cdot)$ relates the measurements z_k with the vehicle state x_k^v and v_k is the measurement noise being zero mean white Gaussian sequence with covariance R_k. Equations (5.4) and (5.5) involve nonlinear terms and require use of EKF that approximates (linearize) the nonlinear functions. The EKF-SLAM solution is very well know and inherits many of the same benefits and problems as the EKF solutions in navigation and tracking applications [Durrant-White and Bailey, 2006].

5.2.1 Vehicle and Augmented State Vector

As the environment is explored new landmarks are observed and are added along the vehicle state thus creating single augmented state vector. This form of the state vector allows the Kalman filter to maintain in its covariance matrix a measure of all the correlations between the errors in the vehicle and the map. The augmented state vector and the covariance matrix take the following form:

$$x_k = \begin{bmatrix} x_k^v \\ m_1 \\ \vdots \\ m_n \end{bmatrix} \tag{5.6}$$

$$P_k = \begin{bmatrix} P^{vv} & P^{vm1} & \dots & P^{vmn} \\ P^{m1v} & P^{m1m1} & \dots & P^{m1mn} \\ P^{m2v} & P^{m2m1} & \dots & P^{m2mn} \\ \vdots & \vdots & \ddots & \vdots \\ P^{mnv} & P^{mnm1} & \dots & P^{mnmn} \end{bmatrix} \tag{5.7}$$

where P^{vv} is the covariance of the vehicle estimate and P^{ii} is the covariance of the i-th landmark. The elements $P^{vi} = P^{iv^T}$ measure the correlation between the error in the vehicle and the landmarks and $P^{ij} = P^{ji^T}$ measure the correlation between the errors in two map point estimates x_k^{mi} and x_k^{mj}.

5.2.2 The Estimation Process

After the augmentation of the vehicle state with the map and creating the augmented state vector, EKF update equations can be applied to compute the mean and covariance

Prediction

$$\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1|k-1}) \tag{5.8}$$

$$P_{k|k-1} = \nabla f_k^x P_{k-1|k-1} \nabla f_k^{xT} + Q_k \tag{5.9}$$

where ∇f_k^x is the Jacobian of the nonlinear function $f(\cdot)$ is with respect to the state x_k and is evaluated at $\hat{x}_{k-1|k-1}$.

Update

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k[z_k - h_k(\hat{x}_{k|k-1})] \tag{5.10}$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{5.11}$$

where the innovation covariance S_k and the Kalman gain K_k are calculated by

$$S_k = \nabla h_k^x P_{k|k-1} \nabla h_k^{xT} + R_k \tag{5.12}$$

$$K_k = P_{k|k-1} \nabla h_k^{xT} S_k^{-1} \tag{5.13}$$

Similar ∇h_k^x is the Jacobian of the nonlinear function $h(\cdot)$ is with respect to the state x_k and is evaluated at $\hat{x}_{k|k-1}$.

5.3 Structure of the SLAM Problem

This section presents results underlying the structure of the SLAM problem. They are addressing a linear model of the evolution of the state (vehicle model), equation (5.4) and linear observation model, equation (5.5), see [Dissanayake et al., 2001], [Csorba, 1997] for more details.

The correlations between the landmark estimates increase monotonically as more relative measurements are made. The complete covariance matrix P_k in block form is shown with equation (5.7). The determinant of the state covariance matrix presents a measure of the volume of the uncertainty ellipsoid (contour

ellipsoid, see Appendix B) associated with the state estimate. The algorithm is initialized using a positive semidefinite (psd) state covariance matrix $P_{k-1|k-1}$. The matrices Q_k and R_k are both psd, and consequently the matrices $P_{k|k-1}$, S_k, $K_k S_k K_k^T$ and $P_{k|k}$ are all psd. The update equations ensure that total uncertainty of the state estimate does not increase during an update:

$$\det P_{k|k} = \det[P_{k|k-1} - K_k S_k K_k^T] \leq \det P_{k|k-1} \tag{5.14}$$

The general properties of psd matrices ensure that this holds for any submatrix of the map covariance matrix. The proof of this property for more general probabilistic case still remains an open problem [Durrant-White and Bailey, 2006]. The correlations are critical part of the SLAM problem and none of these correlations can be assumed zero. As relative measurements are taken during the update process, correlations ensure that all landmarks are updated and the knowledge of the position of the landmarks improves. Important note is that the improvement in the knowledge in the map does not mean that the determinants of the landmarks covariance matrices will tend to zero. The limiting covariance of the map can never be below a certain limit which is a function of the initial covariance of the vehicle estimate P_0^{vv} and the process and measurement noise matrices Q_k and R_k, respectively.

5.4 Limitations of the Kalman Filter Solutions to the SLAM Problem

Almost any practical SLAM application will involve nonlinear vehicle and/or observation models. These models can only be approximately applied within the Kalman filter equations using linearization of some kind. By far, the most common is the use of the Extended Kalman filter (EKF) to solve the SLAM problem. Nonlinearities can be a significant problem here and can lead to inconsistency in the solutions, as shown in [Julier and Uhlmann, 2007]. Unfortunately, the errors incurred by the linearization are introduced into both the vehicle and landmark estimates, but because the errors are unknown, the effect on the cross covariance

cannot be determined [Julier and Uhlmann, 2001]. Smith, Self and Cheeseman in their original work [Smith et al., 1990], address the errors due to the nonlinearities and they argue that these errors can be greatly reduced by iteration using the Iterated Extended Kalman Filter equations. In contrast Julier and Uhlmann in [Julier and Uhlmann, 2001] argue that by replacing the Kalman filter by more sophisticated methods (such as Unscented Kalman filter or the Iterated Extended Kalman Filter) does not significantly affect the results. The reason they say, is that any errors will ultimately undermine the integrity of the filter.
The computational issue raising from the real-time implementations of the Kalman filter based SLAM is still present. This is due the observation update step, which requires that all landmarks and the joint covariance matrix be updated every time an observation is made. Naively, this means that the computation grows quadratically with the number of landmarks. The standard formulation of the EKF-SLAM solution is especially fragile to incorrect data association of observations to landmarks. The loop-closure problem, when a vehicle returns to reobserve landmarks after a large traverse, is especially difficult. The association problem is compounded in environments where landmarks are not simple points and indeed look different from different viewpoints [Durrant-White and Bailey, 2006], [Bailey and Durrant-White, 2006].

5.5 Rao-Blackwellized Particle Filter Solution to the SLAM Problem

The implementation of the Rao-Blackwellized Particle Filter (RBPF) as solution to the SLAM problem is known as the Fast-SLAM algorithm, first introduced in [Montemerlo et al., 2002]. This algorithm was first to directly represent the nonlinear process model and the non-Gaussian vehicle states distribution and has made the fundamental conceptual shift in the design of probabilistic SLAM [Durrant-White and Bailey, 2006].

The RBPF as solution to the SLAM problem estimates the posterior given by equation (5.1). With the key factorization

$$p(x_k^v, m|z_{1:k}, u_{1:k}) = p(m|x_k^v, z_{1:k})p(x_k^v|z_{1:k}, u_{1:k}) \tag{5.15}$$

we derive the recursive algorithm which in each iteration updates the samples (trajectory particles) $x_k^{v(i)}$ and then builds the corresponding map m^i in closed form i.e. analytically.
The RBPF iteration steps are as follows:

1. First, new samples $x_k^{v(i)}$ are drawn from the proposal (importance) density function $g(\cdot)$ from the previous samples $x_{k-1}^{v(i)}$

2. Then we assign the importance weights to each of the samples using equation (4.38)

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_{k-1}^{m(i)}, x_k^{v(i)})p(x_k^{v(i)}|x_{k-1}^{v(i)}, u_k)}{g(x_k^v|x_{k-1}^{v(i)}, z_k, u_k)} \tag{5.16}$$

3. Perform the resampling method given in section 4.6.4.

4. For each sample the corresponding landmark estimate $p(m^i|x_{1:k}^{v(i)}, z_{1:k})$ is computed based on the trajectory of the samples $x_{1:k}^{v(i)}$ and the measurements $z_{1:k}$.

The robustness and efficiency of the RBPF strongly depends on the proposal (importance) density $g(\cdot)$. If the density differs too much from the true posterior then there is a high risk that the filter might be divergent.

Chapter 6

Inertial Navigation Aided by Simultaneous Localization and Mapping

Integrated navigation system design requires selection of set of a sensors and computation power that provides reliable and accurate navigation parameters (position, velocity and attitude) with high update rates and bandwidth in small and cost effective manner. Inertial and vision sensors with their complementary characteristics have the potential to meet these requirements. In this chapter we present a sensor fusion algorithm that provides aiding information to IN from vision sensor. Using vision sensors and with no a priori knowledge of the environment while maneuvering in vicinity of the map point it is shown that it is possible to constrain the Inertial Navigation position divergence (to bound the estimation error) and to eliminate some of the uncertainty of the map point estimates. This concept brings different parametrization (feature initialization) of the map points in SLAM and we refer to it as concept of aiding IN by SLAM.

6.1 Simultaneous Localization and Mapping as Sensor Fusion Algorithm

As mentioned in chapter 4, sensor fusion can be regarded as a method for integrating information from various sensors. The idea of integration is to take advantage of the complementary strengths of the sensors.
SLAM uses relative measurements (range and bearing) from the vehicle with respect to the environment to build a map of the environment whilst simultaneously using the generated map to compute vehicle position, see figure (6.1). There, the true trajectory of the vehicle is represented by solid line, the estimated trajectory by a dashed line.

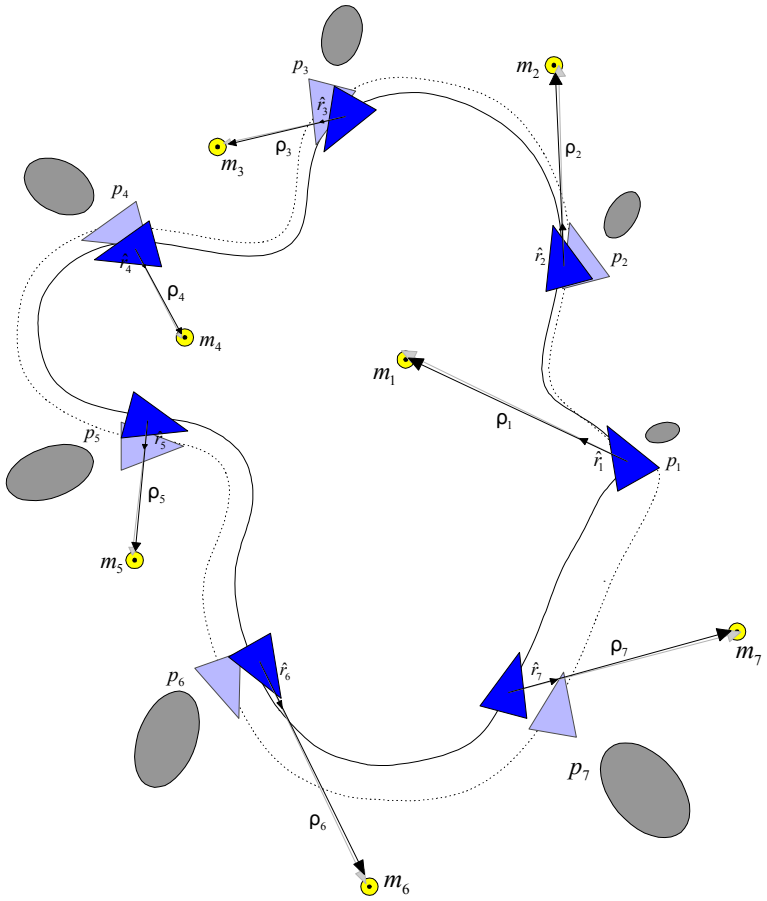

Figure 6.1: Simultaneous Localization and Mapping

In SLAM, both the estimated trajectory of the vehicle and the estimated position of all the map points $m_1, m_2, \ldots, m_7$ are generated online without the need for a priori knowledge of environment [Durrant-White and Bailey, 2006], [Bailey and Durrant-White, 2006]. With reference to figure (6.1), as the vehicle moves and explores the environment, new map points are observed from vehicle positions and added to the map. The approaches dealing with SLAM in [Smith et al., 1990], [Dissanayake et al., 2001] and [Csorba, 1997] augment the vehicle states with these new map positions thus creating single augmented state vector. These processes provide a general representation for the spatial relationships of the map points and the vehicle where all the spatial variables are tied together in one vector. The problem arises that the estimated vehicle positions (light blue triangles) will differ from the true vehicle positions (blue triangles) due to accumulated uncertainty in the vehicle position and hence will add errors to the map positions and will correlate the map point estimates with one another. As the vehicle progresses through the environment these errors become increasingly correlated, and never will become less correlated [Dissanayake et al., 2001].
When the vehicle revisits the stored map points, for example the map point m_1, the accumulated uncertainty in the vehicle position (grey ellipses on figure (6.1)) can be estimated and the map point m_1 uncertainty can be reduced. The map points are all correlated therefore are updated and the overall map accuracy is improved. The revisiting process makes it possible to build a precise map, where the error in each map point reaches a lower bound determined only by the initial uncertainty of the vehicle position. Within this form of the augmented state vector, taking repeated measurements of the map point cannot further reduce the uncertainty of the map points [Dissanayake et al., 2001].
Certain movements of the vehicle around or nearby the map point provide valuable information's of the spatial relationship of the map point and the vehicle, see figure (6.2). SLAM theory is interested in the bigger picture of the spatial relationships of how to build a map of the environment and simultaneously to use the map to compute the vehicle position and does not use this information's. Our interest is towards the smaller picture. We are looking for solution that will constrain the position divergence and will reduce the uncertainty of the map points while maneuvering in vicinity of the map point.

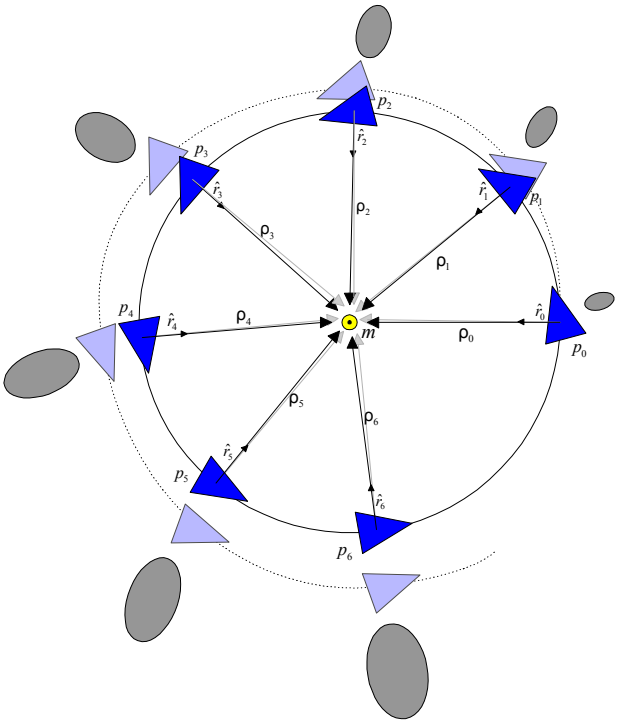

Figure 6.2: Repeated relative measurements of a map point (circular movement)

As the vehicle moves and circles, the map point m is observed from the vehicle positions $p_1, p_2, \ldots, p_6$. The problem arises that the estimated vehicle positions (light blue triangles) will differ from the true vehicle positions (blue triangles) due to accumulated uncertainty in the vehicle position (grey ellipses on figure (6.2)). This uncertainty will be transferred to the map point. The measurement errors cause the relative measurements to be not perfect, see the gray relative measurements on figure (6.2). Because of the accumulated uncertainty in the vehicle position and the measurement errors we cannot compute the map point position exactly. Instead from each vehicle position $p_k = [\ x \quad y \quad z\ ]^T$ an map point position estimate $x_k^{mi} = [\ x^i \quad y^i \quad z^i\ ]^T$ can be calculated. Figure (6.2) shows six vehicle positions $p_1, p_2, \ldots, p_6$, which will bring six map point estimates $x^{m1}, x^{m2}, \ldots, x^{m6}$ calculated by

$$x_k^{mi} = p_k + \rho_k \hat{r}_k \tag{6.1}$$

where the unit direction vector $\hat{r}_k$ and the range ρ_k are the relative measurements. This equation can be written in component form, within a more a general model with the vehicle position part of the vehicle state x_k^v, the relative measurements z_k and the measurement errors v_k which we assume that are zero mean with known covariance.

$$x_k^{mi} = g^i[x_k^v, z_k, v_k] = \begin{bmatrix} x + \rho\hat{r}_x \\ y + \rho\hat{r}_y \\ z + \rho\hat{r}_z \end{bmatrix} \tag{6.2}$$

This requires the spatial variables i.e. the map point estimates and the vehicle states be treated as an intrinsic part of the spatial representation. The general representation used in SLAM where vehicle states are augmented with the map point estimates and where the associated covariance matrix represents the uncertainty of each of the map point estimates and their inter-dependencies is appropriate. This representation will bring different parameterization (feature initialization) of the map point and we refer to it as concept of aiding IN by SLAM. The approach proposed is to augment the state vector not only with one map point estimate but with certain number of estimates for each map point. In theory, the number of augmented map point estimates for each map point can go to infinity. Heuristically through simulation we realized that we need more then three map point estimates. The idea of augmenting the state vector with number of map point estimates should perhaps have been noted sooner in SLAM. In [Kwok and Dissanayake, 2004] the authors augment the state vector with multiple hypotheses i.e. map point estimates for each map point. They do this in order to solve the map point initialization problem in Bearing-Only Simultaneous Localization and Mapping (BOSLAM). As successive observations of the map point are made, they remove all but one hypothesis from the map and use that one in the update.

6.1.1 Inertial Navigation and Augmented State Vector

When SLAM is performed on aerial vehicles as in [Kim, 2004], [Watkins, 2007], [Langelaan, 2006], [Ivey and Johnson, 2006], [Jung and Lacroix, 2003] the vehicle state x_k^v can be represented by Inertial Navigation (IN) mechanization equations

which give position $p_k^n = [\ x \quad y \quad z\ ]^T$, velocity $v_k^n = [\ V_x \quad V_y \quad V_z\ ]^T$ and attitude quaternion $q_k = [\ q_0 \quad q_1 \quad q_2 \quad q_3]^T$ of the vehicle in navigation frame:

$$\begin{bmatrix} p_k^n \\ v_k^n \\ q_k \end{bmatrix} = \begin{bmatrix} v_{k-1}^n \Delta t + p_{k-1}^n \\ [C_b^n f_k^b + g^n]\Delta t + v_{k-1}^n \\ \frac{1}{2}\Omega q_{k-1}\Delta t + q_{k-1} \end{bmatrix} \tag{6.3}$$

or

$$x_k^v = f_v(x_{k-1}^v, u_k) + w_k^v \tag{6.4}$$

where $f_k^b = [\ f_x^b \quad f_y^b \quad f_z^b\ ]^T$ are the Inertial Measurement Unit (IMU) acceleration measurements given in body frame. These measurements account the measurements errors such as the additive accelerometer noise $w_k^a = [\ w^{ax} \quad w^{ay} \quad w^{az}\ ]^T$ and the accelerometer bias $\alpha_k^a = [\ \alpha^{ax} \quad \alpha^{ay} \quad \alpha^{az}]^T$. The gravity model in navigation frame g^n can be assumed constant. The transformation matrix C_b^n is that transforms body frame coordinates into navigation frame coordinates given in quaternions. The matrix Ω is the angular rate matrix as a function of IMU angular rates (gyro) measurements $\omega_k^b = [\ p \quad q \quad r\ ]^T$:

$$\Omega = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \tag{6.5}$$

The gyro measurements account the measurements errors such as the additive gyro noise $w_k^g = [\ w^{gx} \quad w^{gy} \quad w^{gz}\ ]^T$ and the gyro bias $\alpha_k^g = [\ \alpha^{gx} \quad \alpha^{gy} \quad \alpha^{gz}]^T$ as well. The acceleration measurements errors cause the IN velocity and position to diverge with time, see equation (6.1). The gyro measurements errors cause the attitude quaternion $[\ q_0 \quad q_1 \quad q_2 \quad q_3]^T$ to diverge as well. The process noise matrix w_k^v models these error processes as zero-mean white Gaussian with known covariance Q_k^v:

$$Q_k^v = diag\ [\sigma_x^2 \quad \sigma_y^2 \quad \sigma_z^2 \quad \sigma_{V_x}^2 \quad \sigma_{V_y}^2 \quad \sigma_{V_z}^2 \quad \sigma_{q_0}^2 \quad \sigma_{q_1}^2 \quad \sigma_{q_2}^2 \quad \sigma_{q_3}^2] \tag{6.6}$$

As proposed in section 6.1, the approach is to augment the vehicle state with number of map point estimates $x_k^{mi} = [\ x^i \quad y^i \quad z^i]^T \qquad i = 1, 2, \ldots, n$ of the map point:

$$x_k^m = [\ x_k^{m1T} \quad x_k^{m2T} \quad \ldots \quad x_k^{mnT}\ ]^T \tag{6.7}$$

By assumption the unknown map point is stationary so no process noise acts upon the map point estimates:

$$x_k^{mi} = x_{k-1}^{mi} \tag{6.8}$$

The augmented state vector containing the vehicle state and the map point estimates is denoted as:

$$x_k = [\ x_k^{vT} \quad x_k^{m1T} \quad \ldots \quad x_k^{mnT}\ ]^T \tag{6.9}$$

The augmented process model is of the form

$$x_k = f(x_{k-1}, u_k) + w_k = \begin{pmatrix} f_v(x_{k-1}^v, u_k) + w_k^v \\ x_{k-1}^{m1} \\ \vdots \\ x_{k-1}^{mn} \end{pmatrix} \tag{6.10}$$

where w_k is the augmented process noise matrix with covariance matrix Q_k:

$$Q_k = \begin{bmatrix} Q_k^v & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}.$$

The corresponding covariance P_k of the mean x_k of the augmented state vector (6.9) is:

$$P_k = \begin{bmatrix} P_k^{vv} & P_k^{vm1} & \cdots & P_k^{vmn} \\ P_k^{m1v} & P_k^{m1m1} & \cdots & P_k^{m1mn} \\ P_k^{m2v} & P_k^{m2m1} & \cdots & P_k^{m2mn} \\ \vdots & \vdots & \ddots & \vdots \\ P_k^{mnv} & P_k^{mnm1} & \cdots & P_k^{mnmn} \end{bmatrix} \tag{6.11}$$

where P_k^{vv} is the covariance of the vehicle state estimate and P_k^{ii} is the covariance of the i-th map point estimate. The elements $P_k^{vi} = P_k^{iv^T}$ measure the correlation between the error in the vehicle and the map point estimates and $P_k^{ij} = P_k^{ji^T}$ measure the correlation between the errors in two map point estimates x_k^{mi} and x_k^{mj}. The method for augmenting the map point estimates, equation (6.9) and building the covariance matrix, equation (6.11) is as follows. The new map point estimate is added to the augmented state vector and a new row and column are added to the covariance matrix to describe the uncertainty in the map point estimate and the interdependencies of this estimate with the other map point estimates [Smith et al., 1990]. The expanded system[1] is:

$$x_k^+ = f_{aug}[x_k, z_k] = \begin{bmatrix} x_k \\ g^i[x_k, z_k] \end{bmatrix} \tag{6.12}$$

$$P_k^+ = \nabla f_{aug}(\cdot) P_k \nabla f_{aug}^T(\cdot) = \begin{bmatrix} I & 0 \\ \nabla g_k^x & \nabla g_k^z \end{bmatrix} \begin{bmatrix} P_k & 0 \\ 0 & R_k \end{bmatrix} \begin{bmatrix} I & 0 \\ \nabla g_k^x & \nabla g_k^z \end{bmatrix}^T =$$

$$= \begin{bmatrix} P_k & P_k \nabla g_k^{x^T} \\ \nabla g_k^x P_k & \nabla g_k^x P_k \nabla g_k^{x^T} + \nabla g_k^z R_k \nabla g_k^{z^T} \end{bmatrix} \tag{6.13}$$

where ∇g_k^x and ∇g_k^z are Jacobians of function $g(\cdot)$ (6.2) with respect to the augmented state x_k and the observation z_k respectively and are given in Appendix C. Equation (6.12) and (6.13) are repeated until we augment the state vector with "sufficient" number of map point estimates. Through simulation we realized that we need more then three map point estimates.

6.1.2 Observation Models

After the augmentation of the state vector the vehicle is maneuvering, circling around the map point. These maneuvers provide the relative measurements (observations) that can be used to constrain the IN position divergence and reduce the covariance of the map point estimates. The vision sensor (stereo camera) can

[1] the plus (+) represents the augmentation of the state vector with the new map point estimate

provide observations such as the range components in body frame, see chapter 3 for details:

$$z_k = [\ \rho_{bx} \quad \rho_{by} \quad \rho_{bz}]^T \tag{6.14}$$

where

$$\begin{aligned}
\rho_{bx} &= (q_0^2 + q_1^2 - q_2^2 - q_3^2)(x^i - x) + 2(q_1q_2 + q_0q_3)(y^i - y) + 2(q_1q_3 - q_0q_2)(z^i - z) \\
\rho_{by} &= 2(q_1q_2 - q_0q_3)(x^i - x) + (q_0^2 - q_1^2 + q_2^2 - q_3^2)(y^i - y) + 2(q_2q_3 + q_0q_1)(z^i - z) \\
\rho_{bz} &= 2(q_1q_3 + q_0q_2)(x^i - x) + 2(q_2q_3 - q_0q_1)(y^i - y) + (q_0^2 - q_1^2 - q_2^2 + q_3^2)(z^i - z)
\end{aligned}$$

The observation model for the vehicle observations (6.14) of the map point can be written:

$$z_k^i = h^i[x_k^v, x_k^{mi}, v_k] = C_n^b \begin{bmatrix} x^i - x \\ y^i - y \\ z^i - z \end{bmatrix} + v_k \tag{6.15}$$

where $C_n^b = (C_b^n)^T$ is the transformation matrix which transforms navigation frame coordinates into body frame coordinates and v_k is the observation noise with covariance R_k:

$$R_k = diag[\ \sigma_{\rho_{bx}}^2 \quad \sigma_{\rho_{by}}^2 \quad \sigma_{\rho_{bz}}^2\ ] \tag{6.16}$$

If we consider each camera as an independent sensor in the stereo camera setup, then the vision sensor can provide two unit direction vectors in body frame, see chapter 3 for details:

$$z_k = \begin{bmatrix} \hat{r}_{bx} & \hat{r}_{by} & \hat{r}_{bz} & \hat{r}'_{bx} & \hat{r}'_{by} & \hat{r}'_{bz} \end{bmatrix}^T \tag{6.17}$$

where for the first camera we have:

$$\begin{aligned}
\hat{r}_{bx} &= \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_1q_2+q_0q_3)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_1q_3-q_0q_2)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\
\hat{r}_{by} &= \frac{2(q_1q_2-q_0q_3)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_2q_3+q_0q_1)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\
\hat{r}_{bz} &= \frac{2(q_1q_3+q_0q_2)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_2q_3-q_0q_1)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}}
\end{aligned}$$

For the second camera using its position $p' = p + D$ and its components $x' = x + D_x$, $y' = y + D_y$ and $z' = z + D_z$ where the camera offset D is a constant

vector we have:

$$\begin{aligned}
\hat{r}'_{bx} &= \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(x^i-x')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{2(q_1q_2+q_0q_3)(y^i-y')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{2(q_1q_3-q_0q_2)(z^i-z')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} \\
\hat{r}'_{by} &= \frac{2(q_1q_2-q_0q_3)(x^i-x')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(y^i-y')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{2(q_2q_3+q_0q_1)(z^i-z')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} \\
\hat{r}'_{bz} &= \frac{2(q_1q_3+q_0q_2)(x^i-x')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{2(q_2q_3-q_0q_1)(y^i-y')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(z^i-z')}{\sqrt{(x^i-x')^2+(y^i-y')^2+(z^i-z')^2}}
\end{aligned}$$

The observation model for the vehicle observations (6.17) of the map point can be written:

$$z_k^i = h^i[x_k^v, x_k^{mi}, v_k] = C_n^b \begin{bmatrix} x^i - x \\ y^i - y \\ z^i - z \\ x^i - x' \\ y^i - y' \\ z^i - z' \end{bmatrix} + v_k \tag{6.18}$$

where v_k the observation noise is with covariance R_k:

$$R_k = diag[\ \sigma^2_{\hat{r}_{bx}} \quad \sigma^2_{\hat{r}_{by}} \quad \sigma^2_{\hat{r}_{bz}} \quad \sigma^2_{\hat{r}'_{bx}} \quad \sigma^2_{\hat{r}'_{by}} \quad \sigma^2_{\hat{r}'_{bz}}\ ] \tag{6.19}$$

If the vehicle is equipped with single camera then the vision sensor can provide a unit direction vector in body frame, see chapter 3 for details:

$$z_k = \begin{bmatrix} \hat{r}_{bx} & \hat{r}_{by} & \hat{r}_{bz} \end{bmatrix}^T \tag{6.20}$$

where

$$\begin{aligned}
\hat{r}_{bx} &= \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_1q_2+q_0q_3)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_1q_3-q_0q_2)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\
\hat{r}_{by} &= \frac{2(q_1q_2-q_0q_3)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_2q_3+q_0q_1)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\
\hat{r}_{bz} &= \frac{2(q_1q_3+q_0q_2)(x^i-x)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{2(q_2q_3-q_0q_1)(y^i-y)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(z^i-z)}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}}
\end{aligned}$$

The observation model for the vehicle observations (6.20) of the map point can be written:

$$z_k^i = h^i[x_k^v, x_k^{mi}, v_k] = C_n^b \begin{bmatrix} \frac{x^i - x}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\ \frac{y^i - y}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\ \frac{z^i - z}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \end{bmatrix} + v_k \tag{6.21}$$

where v_k the observation noise is with covariance R_k:

$$R_k = diag[\ \sigma^2_{\hat{r}_{bx}} \quad \sigma^2_{\hat{r}_{by}} \quad \sigma^2_{\hat{r}_{bz}}\ ] \tag{6.22}$$

Assuming no attitude errors will avoid the errors (additive gyro noise and gyro bias) in the angular rates (gyro) measurements and will bring a transformation matrix C_b^n without errors in equation (6.15). This assumption is very useful in section 6.2 in the derivation of the convergence properties of the proposed concept of aiding IN by SLAM. The vision sensor (stereo camera) with the assumption of no attitude errors can provide range components in navigation frame:

$$z_k = [\ \rho_x \quad \rho_y \quad \rho_z]^T \tag{6.23}$$

The observation model for the vehicle observations of the map point can be written:

$$z_k^i = h^i[x_k^v, x_k^{mi}, v_k] = \begin{bmatrix} x^i - x \\ y^i - y \\ z^i - z \end{bmatrix} + v_k \tag{6.24}$$

where v_k the observation noise is with covariance R_k:

$$R_k = diag[\ \sigma^2_{\rho_x} \quad \sigma^2_{\rho_y} \quad \sigma^2_{\rho_z}\ ] \tag{6.25}$$

Similar for the single camera, if we avoid the errors in the gyro measurements and in the transformation matrix C_b^n in equation (6.21), the vision sensor can provide unit direction vector in navigation frame:

$$z_k = \begin{bmatrix} \hat{r}_x & \hat{r}_y & \hat{r}_z \end{bmatrix}^T \tag{6.26}$$

The observation model for the vehicle observations of the map point can be written:

$$z_k^i = h^i[x_k^v, x_k^{mi}, v_k] = \begin{bmatrix} \frac{x^i - x}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\ \frac{y^i - y}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \\ \frac{z^i - z}{\sqrt{(x^i-x)^2+(y^i-y)^2+(z^i-z)^2}} \end{bmatrix} + v_k \tag{6.27}$$

where v_k the observation noise is with covariance R_k:

$$R_k = diag[\ \sigma^2_{\hat{r}_x} \quad \sigma^2_{\hat{r}_y} \quad \sigma^2_{\hat{r}_z}\ ] \tag{6.28}$$

6.1.3 The Estimation Process

After the augmentation, we start updating the augmented state vector x_k by using each of the map points estimates at a time. As with the approaches dealing with SLAM [Smith et al., 1990], [Dissanayake et al., 2001] and [Csorba, 1997] the EKF can be implemented for the update. The state covariance is propagated using the Jacobians of the augmented process model:

$$P_{k|k-1} = \nabla f_k^x P_{k-1|k-1} \nabla f_k^{xT} + Q_k \tag{6.29}$$

where ∇f_k^x is the Jacobian of the augmented process model (6.10) with respect to the state x_k evaluated at $\hat{x}_{k|k-1}$ and is given in Appendix D. When an observation z_k occurs we update the augmented state vector and its covariance:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k v_k \tag{6.30}$$

$$P_{k|k} = P_{k|k-1} - K_k \nabla h_k^x P_{k|k-1} \tag{6.31}$$

where v_k is the innovation vector and K_k is the Kalman gain. These are computed by

$$v_k = z_k - \hat{z}_{k|k-1} \tag{6.32}$$

$$\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1}) \tag{6.33}$$

$$K_k = P_{k|k-1} \nabla h_k^{xT} [\nabla h_k^x P_{k|k-1} \nabla h_k^{xT} + R_k]^{-1} \tag{6.34}$$

where ∇h_k^x is the Jacobian of the nonlinear observation function $h(\cdot)$ (6.15) or (6.24) with respect to the state x_k evaluated at $\hat{x}_{k|k-1}$ i.e. the vehicle state $\hat{x}^v_{k|k-1}$ and the map point estimate used in the update $\hat{x}^{mi}_{k|k-1}$. The Jacobian ∇h_k^x is defined in Appendix E and Appendix F, respectively.

6.2 On the Convergence of Inertial Navigation Aided by Simultaneous Localization and Mapping

The covariance matrix P_k gives a measure of the uncertainty of the state estimate $\hat{x}_k$. As described in Appendix B, the contour ellipsoid provides graphical representation of the uncertainty in different state space directions. With time the state estimate $\hat{x}_k$ becomes more uncertain and the ellipsoid grows, whereas the measurements tend to reduce the uncertainty and to shrink the ellipsoid. The convergence of the estimate means that the ellipsoid is shrinking in all directions. For simplicity in attaining analytical results let's assume that the vehicle state x^v_k and the map point estimates x^{mi}_k are one dimensional vectors. Further lest assume that we have two map point estimates. The augmented state vector containing the vehicle state and the two map point estimates can be written:

$$\hat{x}_k = \begin{bmatrix} x^v & x^{m1} & x^{m2} \end{bmatrix}^T \tag{6.35}$$

These assumptions make the convergence analysis easily tractable. The results that follow intuitively apply for the rest of the dimensions of the vehicle state and the map point estimates. They also apply for greater number of map point estimates augmented in the state vector. The covariance matrix being a symmetric matrix which describes the uncertainty of $\hat{x}_k$ before the measurements is:

$$P_k = \begin{bmatrix} \sigma^2_{11} & \sigma^2_{12} & \sigma^2_{13} \\ \sigma^2_{12} & \sigma^2_{22} & \sigma^2_{23} \\ \sigma^2_{13} & \sigma^2_{23} & \sigma^2_{33} \end{bmatrix} \tag{6.36}$$

where σ_{11}^2 is the initial mean square error in knowledge of the vehicle state x^v, σ_{22}^2 is the initial mean square error in knowledge of the first map point estimate x^{m1}, σ_{12}^2 measures the corresponding cross correlation. The initial mean square error in knowledge of the second map point estimate x^{m2} is σ_{33}^2. The cross correlation between the second map point estimate and the vehicle state is given by σ_{13}^2 and the cross correlation between the two map point estimates is given by σ_{23}^2.
We wish to find the improvements in knowledge of the vehicle state and the two map point estimates through a processing a single noisy relative measurement between the vehicle and the first map point estimate following the observation model (6.24). The measurement matrix H_k is:

$$H_k = [\ -1 \quad 1 \quad 0]^T \tag{6.37}$$

with measurement noise covariance $R_k = [\sigma_R^2]$. Using these matrix quantities the Kalman gain K_k is calculated, see section 4.2 equation (4.14):

$$K_k = \begin{bmatrix} \frac{-\sigma_{11}^2+\sigma_{12}^2}{\sigma_{11}^2-2\sigma_{12}^2+\sigma_{22}^2+\sigma_R^2} \\ \frac{-\sigma_{12}^2+\sigma_{22}^2}{\sigma_{11}^2-2\sigma_{12}^2+\sigma_{22}^2+\sigma_R^2} \\ \frac{-\sigma_{13}^2+\sigma_{23}^2}{\sigma_{11}^2-2\sigma_{12}^2+\sigma_{22}^2+\sigma_R^2} \end{bmatrix} \tag{6.38}$$

After several manipulations and simplifications the updated covariance matrix P_k is calculated, see section 4.2 equation (4.13). Here P_k is shown only with its diagonal elements, since they contain the mean square errors:

$$P_k = \begin{bmatrix} \frac{\sigma_{11}^2(\sigma_{22}^2(1-\rho^2)+\sigma_R^2)}{\sigma_{11}^2-2\rho\sigma_{11}\sigma_{22}+\sigma_{22}^2+\sigma_R^2} & & \\ & \frac{\sigma_{22}^2(\sigma_{11}^2(1-\rho^2)+\sigma_R^2)}{\sigma_{11}^2-2\rho\sigma_{11}\sigma_{22}+\sigma_{22}^2+\sigma_R^2} & \\ & & \sigma_{33}^2(1-\frac{\rho_1^2\sigma_{11}^2-2\rho_1\rho_2\sigma_{11}\sigma_{22}+\rho_2^2\sigma_{22}^2}{\sigma_{11}^2-2\rho\sigma_{11}\sigma_{22}+\sigma_{22}^2+\sigma_R^2}) \end{bmatrix} \tag{6.39}$$

where the correlation coefficients are defined by $\rho = \frac{\sigma_{12}^2}{\sigma_{11}\sigma_{22}}$, $\rho_1 = \frac{\sigma_{13}^2}{\sigma_{11}\sigma_{33}}$ and $\rho_2 = \frac{\sigma_{23}^2}{\sigma_{22}\sigma_{33}}$. This is done for clarity and better analysis of the following few limiting cases.

When the correlation coefficients $\rho = 0$, $\rho_1 = 0$ and $\rho_2 = 0$, a case when there are no correlations between the estimates or a case when the correlations are ignored, the final uncertainty in the estimates is given by:

$$P_k = \begin{bmatrix} \sigma_{11}^2 \left(\frac{\sigma_{22}^2 + \sigma_R^2}{\sigma_{11}^2 + \sigma_{22}^2 + \sigma_R^2} \right) & & \\ & \sigma_{22}^2 \left(\frac{\sigma_{11}^2 + \sigma_R^2}{\sigma_{11}^2 + \sigma_{22}^2 + \sigma_R^2} \right) & \\ & & \sigma_{33}^2 \end{bmatrix} \tag{6.40}$$

As we can see from equation (6.40) for the first two diagonal elements of the updated covariance matrix P_k, the vehicle state and the first map point estimate are updated. Their uncertainty is reduced since the denominators are bigger then the numerators in the fractions in the brackets. This reduction depends on the initial mean square error in knowledge of the vehicle state σ_{11}^2 and the initial mean square error in knowledge of the first map point estimate σ_{22}^2 and the measurement noise covariance σ_R^2.

As we can see from equation (6.40) for the third diagonal element of the updated covariance matrix P_k, the final uncertainty of the second map point estimate is equal to the initial uncertainty. When the correlations are zero or are ignored nothing can be learned from the relative measurements about the second map point estimate. In a case when we have a perfect positive correlations i.e. when $\rho = 1$, $\rho_1 = 1$ and $\rho_2 = 1$, the final uncertainty in the estimates is given by

$$P_k = \begin{bmatrix} \sigma_{11}^2 \left(\frac{1}{1 + \frac{(\sigma_{11} - \sigma_{22})^2}{\sigma_R^2}} \right) & & \\ & \sigma_{22}^2 \left(\frac{1}{1 + \frac{(\sigma_{11} - \sigma_{22})^2}{\sigma_R^2}} \right) & \\ & & \sigma_{33}^2 \left(\frac{1}{1 + \frac{(\sigma_{11} - \sigma_{22})^2}{\sigma_R^2}} \right) \end{bmatrix} \tag{6.41}$$

As we can see from equation (6.41), the vehicle state and the two map point estimates are updated. The amount of the gained information i.e. the reduction in the uncertainty of the diagonal elements of the updated covariance matrix P_k depends upon the ratio of the square of the binomial formed with the standard deviations of the vehicle state σ_{11} and the first map point estimate σ_{22} and the

measurement noise covariance σ_R^2. The covariance matrix shows are the state estimates converging or not, not whether they are converging to the correct value. The consistency property addresses the convergence of the estimates to the correct values. When estimating a state of a dynamic system, the models (the dynamic equation, the measurement equation and the random variables entering into these equations) contain certain approximations. In such estimation processes what one has is the current estimate of the state $\hat{x}_{k|k}$ and its associated covariance matrix $P_{k|k}$. Using these two quantities the consistency asks the state estimator (filter) to give approximate first and second order moments of the state as:

$$E[x_k - \hat{x}_{k|k}] \stackrel{\Delta}{=} E[\tilde{x}_{k|k}] = 0 \tag{6.42}$$

$$E[[x_k - \hat{x}_{k|k}][x_k - \hat{x}_{k|k}]^T] \stackrel{\Delta}{=} E[\tilde{x}_{k|k}\tilde{x}_{k|k}^T] - P_{k|k} \tag{6.43}$$

This property is based on finite number of samples (measurements) which requires that the estimation errors are consistent with their theoretical statistical properties i.e. are unbiased and have covariance matrix as calculated by the filter. In contradiction the consistency is an asymptotic (infinite size sample) property. Then the consistency criteria of the filter require that the state errors and the innovations are zero mean and to have magnitude that commensurate with the state covariance. Several consistency tests that use either the estimation errors (Normalized Estimation Error Squared NEES) or the innovations (Normalized Innovation Squared NIS) are available in the literature to verify the consistency [Bar-Shalom et al., 2001].

6.3 On the Observability Analysis of Inertial Navigation Aided by Simultaneous Localization and Mapping using Fisher Information Matrix

6.3.1 Introduction

Discussion on the observability of a system provides insights and understanding of the fundamental limits of the estimation processes. Since observability analysis can give the best achievable performance even before the system is built, it can be considered as tool for computer analysis of many complicated estimation processes. Observability is a property of a specific state space representation for a system, rather than of the system itself. Certain state space models are more suitable for estimation purposes than others, even though both might accurately portray the input - output characteristics of a system [Maybeck, 1982]. If little is to be gained from the estimation process, then we should consider remodeling the system. This might involve taking additional or alternate measurements, or redesigning the dynamics of the system [Jazwinski, 1970]. In a deterministic sense one system is completely observable if its initial state can be fully and uniquely recovered from its output, observed over a finite time interval, and the knowledge of the input [Bar-Shalom et al., 2001]. When the system is linear time invariant, observability rank test is performed on the equations. Similar for linear time variant systems observability Gramian matrix is used. For nonlinear systems observability analysis most common tools are based on the Lie Algebra, which suffers of being long and not easy to apply [Hermann and Krener, 1977]. Here the observability is related to indistinguishability of the states with respect to the control inputs unlike the linear systems.

Alternative to these methods the information matrix which is a statistical generalization of the observability matrix can be used for observability analysis. This matrix amounts to the information limit being sufficient or insufficient for the estimation process. It allows an interpretation of the estimation in terms of information theoretic concepts. The information matrix commonly referred to us

the Fisher information matrix and its inverse provide the filtering Cramer-Rao lower bound (CRLB). The CRLB is a lower bound on the mean square error that can be achieved by any unbiased estimator and is a useful system design tool today and may be used for comparison of given filtering algorithms performances [Ristic et al., 2004].

Simultaneous Localization and Mapping (SLAM) uses relative measurements (range and bearing) from the vehicle with respect to the environment to build a map of the environment whilst simultaneously using the generated map to compute vehicle position. In SLAM, both the trajectory of the vehicle and the position of all the map points are estimated online without the need for a priori knowledge of environment [Durrant-White and Bailey, 2006], [Bailey and Durrant-White, 2006]. When talking about observability of the SLAM we need to recognize the characteristic state vector i.e. augmented state vector consisting of the vehicle state (described by the process model which evolves with time) and map state (stationary state which is not changing with time). In the SLAM literature several approaches are present. In range bearing SLAM case for aerial vehicle where the SLAM equations are defined as piece-wise linear system which simplifies the observability analysis of the system is presented in [Bryson and Sukkarieh, 2008a]. There the authors have investigated how the states are affected by the maneuvers/control actions taken by the vehicle and reveal the need for vehicle motions for maximizing the observability of the states over multiple time segments. Similarly using the tools from the control theory, observability analysis for BOSLAM case for planar vehicle is performed in [Vidal-Calleja et al., 2007]. Authors clearly emphasize the need to avoid the case of zero velocity and show that the performance of the estimator in BOSLAM is strongly related to the trajectories described by the vehicle. Since SLAM is highly nonlinear and coupled system the nonlinear observability analysis of SLAM is performed in [Lee et al., 2006]. Alternatively to the approaches presented before these authors address the effects of the control inputs on the observability of SLAM. This observability study shows that typical planar SLAM is observable when two a priori known map point's observations are available. The nonlinear observability analysis of SLAM is extended in [Perera et al., 2009]. The authors give proofs of the state observability of SLAM to any number of map points and compare

the linear and nonlinear observability analysis. The observability analysis of the SLAM using the FIM (for linear case) is evaluated in [Andrade-Cetto and Sanfeliu, 2005]. The authors first employ the observability analysis of SLAM on a two-dimensional linearized planar vehicle model. Using the FIM for observability analysis of the SLAM is extended in [Z.Wang and Dissanayake, 2008]. This work introduced novel technique for observability analysis of SLAM. They reformulate the augmented state vector to include all the vehicle states (poses) from where the measurements are taken. This converted the SLAM to a problem of estimating a set of unknown, constant random variables. This technique also allows capturing the information contained in the control inputs between two successive vehicles poses. Two very important conclusions are given in this work. First one is that the range bearing relative SLAM is observable. Relative SLAM being the case when the reference frame is attached to the vehicle initial pose and equivalent to the assumption of perfect knowledge of the vehicle initial pose. The second one refers to the condition for observability of relative BOSLAM and it says that necessary and sufficient condition for relative BOSLAM to be observable is that each feature (map point) present in the environment must be observed at more than one robot pose; and for each feature, the feature location and all the poses from which this feature is observed are not collinear. Further they argue that when the collinearity occur one or both the coordinates of the relevant feature (map point) can not be uniquely determined, while the observability of the states related to other features is not affected.

As described in section 6.1 changes to the SLAM augmented state vector are needed in order to take repeated measurements of the map point and implement SLAM for aiding IN. The proposed approach is to augment the state vector not only with one map point estimate but with certain "sufficient" number of estimates for each map point. This approach further proposes to perform certain movements around the map point which are crucial for constraining the IN position divergence and reducing the covariance of the map points. This in turn provides the map point and the vehicle positions not to be collinear, following the above condition for observability. In this thesis we use the Fisher information matrix for observability analysis of the concept of IN aided by SLAM. Here we will bring more insights and deeper understanding of the limits of the concept.

6.3.2 Estimation Observability and Information

Apparent to the estimation and control theory is that there is a close connection between observability and estimation. Observability being a "yes" or "no" answer cannot be given always, as in general nonlinear estimation processes. However a degree of observability can be given in terms of observed information about the state. Stochastic observability studies the information theoretical point of view of observability. In other words it tries to find quantity of information about states which is contained in the observation process [Mohler and Hwang, 1988].
How well the state is known is measured by the estimation error covariance matrix P_k. Since P_k depends on its initial condition P_0 it doesn't reflect the uncertainty in the estimates by virtue of the filtering the data alone. Setting $P_0^{-1} = 0$, which means that no weight is attached to the prior estimate, then in order to determine the state x_k, the information matrix

$$J_k = \sum_{i=1}^{k} F_k^T(i) H^T(i) R^{-1}(i) H(i) F_k(i) \tag{6.44}$$

must be positive definite [Jazwinski, 1970]. If J_k is singular, then certain linear combinations of the elements of x_k cannot be determined i.e. there is no information about them in the measurement data $\{z_1, \ldots, z_k\}$. From (6.44) we notice that the information matrix depends on F_k and H_k, the linear system model (being noise free) and not on the data themselves. The information matrix satisfies the following difference equation [Maybeck, 1982], [Jazwinski, 1970]:

$$J_k = F_{k|k-1}^T J_{k-1} F_{k|k-1} + H_k^T R_k^{-1} H_k \tag{6.45}$$

and is related to the covariance matrix by

$$J_k = P_k^{-1} - F_k^T P_0^{-1} F_k \tag{6.46}$$

If there were no a priory information about the state, i.e. if $P_0^{-1} = 0$, then the information matrix is the inverse of the estimation error covariance matrix. The larger the eigenvalues of J_k, the smaller the eigenvalues of P_k and the more precise the estimate is. If any eigenvalues of J_k are zero, there are directions in state

space along which the measurements gives us no information [Maybeck, 1982]. The information matrix is commonly referred to us as Fisher information matrix (FIM) and its inverse provide the filtering Cramer-Rao lower bound (CRLB). Consider a nonlinear estimation problem defined by:

$$x_k = f_k(x_{k-1}) + w_k$$
$$z_k = h_k(x_k) + v_k$$

For $\hat{x}_k$ an unbiased estimator of the state vector x_k, based on the measurement data $\{z_1, \ldots, z_k\}$, with prior knowledge of initial density $p(x_0)$, the covariance matrix P_k has a lower bound (CRLB) expressed as follows:

$$P_k = E\left\{[\hat{x}_k - x_k][\hat{x}_k - x_k]^T\right\} \geq J_k^{-1} \tag{6.47}$$

The inequality in (6.47) means that the difference $P_k - J_k^{-1}$ is a positive semidefinite matrix. Matrix J_k being the FIM and its inverse the CRLB. Intuitively more information we have the lower the CRLB is. Essential property of the FIM is that it is additive for independent measurements. For consistent estimator, for N independent and identically distributed samples (measurements) the FIM is N times the individual Fisher information [Cover and Thomas, 2006]. Therefore as more measurements are taken we need to have increasing amount of information. For many practical applications it is desired to calculate the FIM recursively. An elegant method is derived in [Thicavsky et al., 1998] also given in [Ristic et al., 2004] as follows:

$$J_k = D_{k-1}^{22} - D_{k-1}^{21}[J_{k-1} + D_{k-1}^{11}]^{-1} D_{k-1}^{12} \tag{6.48}$$

where

$$D_{k-1}^{11} = -E\left\{\nabla x_{k-1}[\nabla x_{k-1} \log p(x_k|x_{k-1})]^T\right\} \tag{6.49}$$

$$D_{k-1}^{21} = -E\left\{\nabla x_{k-1}[\nabla x_k \log p(x_k|x_{k-1})]^T\right\} \tag{6.50}$$

$$D_{k-1}^{12} = -E\left\{\nabla x_k[\nabla x_{k-1} \log p(x_k|x_{k-1})]^T\right\} = [D_{k-1}^{21}]^T \tag{6.51}$$

$$D_{k-1}^{22} = -E\left\{\nabla x_k[\nabla x_k \log p(x_k|x_{k-1})]^T\right\} - E\left\{\nabla x_k[\nabla x_k \log p(z_k|x_k)]^T\right\} \tag{6.52}$$

The expectation $E\{\cdot\}$ in (6.49), (6.50) and (6.51) is with respect to x_{k-1} and x_k, whereas in (6.52 is with respect to x_{k-1}, x_k and z_k. In absence of process noise i.e. $Q_k = 0$, the evolution of the state vector is purely deterministic. Hence the expectation operator can be dropped out [Ristic et al., 2004]. The recursion of (6.48) then can be written as:

$$J_k = Q_k^{-1} + \tilde{H}_k^T R_k^{-1} \tilde{H}_k - Q_k^{-1} \tilde{F}_{k-1} [J_{k-1} + \tilde{F}_{k-1}^T Q_k^{-1} \tilde{F}_{k-1}]^{-1} \tilde{F}_{k-1}^T Q_k^{-1} \tag{6.53}$$

Using the matrix inversion lemma this simplifies to:

$$J_k = [Q_k + \tilde{F}_{k-1} J_{k-1}^{-1} \tilde{F}_{k-1}^T]^{-1} + \tilde{H}_k^T R_k^{-1} H_k \tag{6.54}$$

Due to the absence of process noise it further simplifies to:

$$J_k = [\tilde{F}_{k-1}^{-1}]^T J_{k-1} \tilde{F}_{k-1}^{-1} + \tilde{H}_k^T R_k^{-1} \tilde{H}_k \tag{6.55}$$

Compare (6.55) to the covariance computation in Extended Kalman Filter (EKF) in section 4.3. If we replace J_k by P_k^{-1} and apply the matrix inversion lemma, these two become identical in their form. The only difference is that the EKF equation features the Jacobians $\hat{F}_{k-1}$ and $\hat{H}_k$, while (6.55) is based on Jacobians $\tilde{F}_{k-1}$ and $\tilde{H}_k$. The difference between a Jacobian with hat $\hat{}$ and with a tilde sign $\tilde{}$ is that the latter is evaluated at the true value of the state vector (which obviously is no available to the EKF) [Ristic et al., 2004]. The conclusion is that the CRLB recursion for nonlinear estimation, in the absence of process noise is identical to the covariance matrix propagation of the EKF, where the Jacobians are evaluated at the true state vector x_k, as first reported in [Taylor, 1978].

6.3.3 Observability Analysis of Inertial Navigation Aided by Simultaneous Localization and Mapping

6.3.3.1 Nonmaneuvering Case (stationary vehicle)

For the nonmaneuvering case of the observability analysis of IN aided by SLAM, the vehicle is stationary. With the vision sensor we are taking repeated measurements (6.15) of the map point. The control input to the IN mechanization

equations (6.3) is $u_k = [\ 0 \quad 0 \quad f_z^b \quad 0 \quad 0 \quad 0\ ]^T$. The f_z^b acceleration measurement appears in the control input due to the Earth gravity field. Ideally this acceleration measurement after the transformation to navigation frame is to be compensated with g^n, the gravity model in navigation frame, see equation (6.3). Considering a case with single map point estimate in the augmented state vector $x_k = [\ x_k^{vT} \quad x_k^{m1^T}\ ]^T$, the process model (6.10) gives the following state transition matrix F_k:

$$F_k = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.56}$$

In the case of stationary vehicle no process noise acts upon the augmented state vector x_k. This allows making the assumption of zero process noise. Then FIM can be computed recursively using equation (6.55), repeated here for convenience:

$$J_k = [\tilde{F}_{k-1}^{-1}]^T J_{k-1} \tilde{F}_{k-1}^{-1} + \tilde{H}_k^T R_k^{-1} \tilde{H}_k \tag{6.57}$$

where $\tilde{F}_k$ is given by equation (6.56). The $\tilde{H}_k$ the Jacobian of the nonlinear function $h(\cdot)$ (6.15) with respect to the augmented state vector x_k is defined in Appendix E. The recursion (6.57) is initialized by

$$J_0 = P_0^{-1},$$

where P_0 is the initial covariance matrix of the state estimate. The measurement noise R_k is modeled as zero-mean white Gaussian sequence with known covariance (6.16).

6.3.3.2 Maneuvering Case (coordinated turn)

For the maneuvering case of the observability analysis of IN aided by SLAM, the vehicle performs a circular movement around the map point commonly referred as coordinated turn (CT). This vehicle maneuver is executed under constant speed and constant turn rate along a circular path at a constant altitude. When the vehicle with CT motion has a constant speed, it satisfies a kinematic constraint: $\underline{V} \cdot \underline{A} = 0$, where $\underline{V}$ is the vehicle velocity vector and $\underline{A}$ being the acceleration vector. Although the CT model prescribes constant speed and constant turn rate it is an idealization which is not met in practice. The Inertial Navigation mechanization equations (6.3) in component from when the vehicle performs CT become:

$$\begin{aligned}
\begin{pmatrix} x \\ y \\ z \end{pmatrix}_k^n &= \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_{k-1}^n \Delta t + \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{k-1}^n \\
\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_k^n &= \left(C_b^n \begin{pmatrix} f_x^b \\ f_y^b \\ f_z^b \end{pmatrix}_k^b + \begin{pmatrix} 0 \\ 0 \\ g^n \end{pmatrix}^n \right) \Delta t + \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}_{k-1}^n \\
\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_k &= \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -r \\ 0 & 0 & r & 0 \\ 0 & -r & 0 & 0 \\ r & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_{k-1} \Delta t + \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}_{k-1}
\end{aligned} \tag{6.58}$$

and the control input $u_k = [\ 0 \quad 0 \quad f_z^b \quad 0 \quad 0 \quad r\ ]^T$. The yaw angular rate r is constant. As in the nonmaneuvering case, the f_z^b acceleration measurement appears in the control input due to the Earth gravity field. Ideally this acceleration measurement after the transformation to navigation frame is to be compensated with g^n, the gravity model in navigation frame. In this case we assume zero process noise. White noise in accelerometers and gyro measurements can cause a

long-term accumulation of errors, known as random walk [Webster, 1999]. In our simulation we operate the IMU for a short period of 100 seconds. This allows us to make the assumption of no white noise in the accelerometers and gyro measurements. This deterministic maneuvering trajectory is not achievable in practice and will give us an FIM which is conservative and overly optimistic. Nevertheless this conservative FIM as we will show in the simulation results is very useful in assessing the comparative error performance of the IN aided by SLAM concept. Again considering a case with single map point estimate in the augmented state vector $x_k = [\ x_k^{vT} \quad x_k^{m1\,T}\ ]^T$ and IN mechanization equations (6.3) when the vehicle performs CT, the process model (6.10) gives the following state transition matrix F_k:

$$F_k = \begin{bmatrix}
1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -r\Delta t/2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & r\Delta t/2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -r\Delta t/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
r\Delta t/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \tag{6.59}$$

As in the nonmaneuvering case the FIM is computed recursively using equation (6.55), with same initialization conditions, but with a difference that $\tilde{F}_k$ is given by equation (6.59).

6.3.4 Observability Analysis of Inertial Navigation Aided by Simultaneous Localization and Mapping Assuming no Attitude Errors

As we mentioned in the section 6.3.1, the observability is a property of a specific state space representation for a system, rather than of the system itself. If little is to be gained from the estimation process, then we should consider remodeling the system. This might involve taking additional or alternate measurements, or redesigning the dynamics of the system [Jazwinski, 1970].

6.3.4.1 Nonmaneuvering Case (stationary vehicle)

As in section 6.3.3.1 for the nonmaneuvering case of the observability analysis the vehicle is stationary. The initialization conditions and the state transition matrix F_k are same as in 6.3.3.1. The difference are the repeated measurements given by equation (6.24) and $\tilde{H}_k$ the Jacobian of the nonlinear function $h(\cdot)$ (6.24) with respect to the augmented state vector x_k, defined in Appendix F.

6.3.4.2 Maneuvering Case (coordinated turn)

As in section 6.3.3.2, for the maneuvering case of the observability analysis the vehicle performs the coordinated turn (CT) maneuver. The initialization conditions and the state transition matrix F_k are same as in 6.3.3.2. The difference are the repeated measurements given by equation (6.24) and $\tilde{H}_k$ the Jacobian of the nonlinear function $h(\cdot)$ (6.24) with respect to the augmented state vector x_k, defined in Appendix F.

6.3.5 Simulation Scenario and Results

A simulation scenario where the vehicle is stationary (first case) and when it performs the CT maneuver around a map point (second case) are used for observability analysis of IN aided by SLAM without and with the assumption of no attitude errors, see figure (6.3) and figure (6.4).

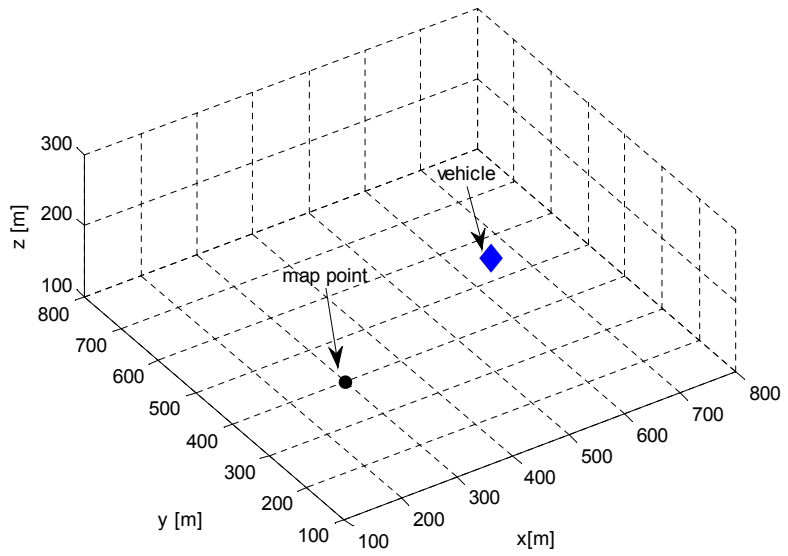

Figure 6.3: Simulation scenario where the vehicle is stationary (first case)

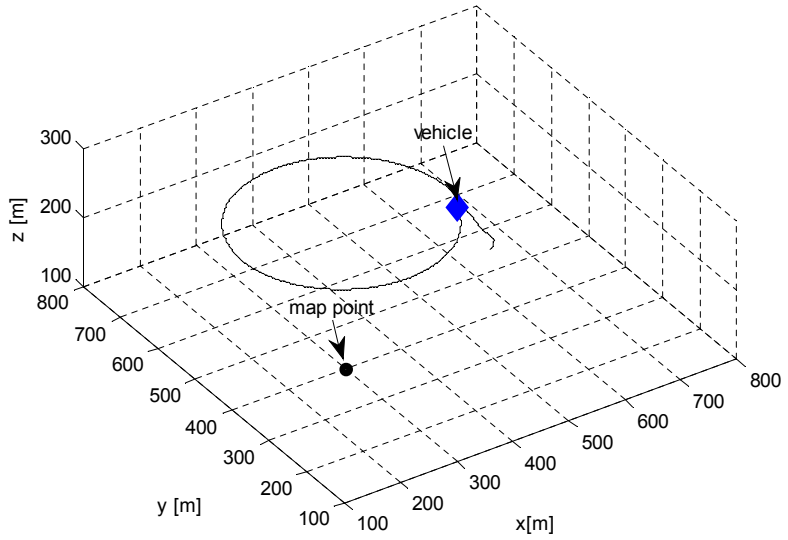

Figure 6.4: Simulation scenario where the vehicle performs CT maneuver around a map point (second case)

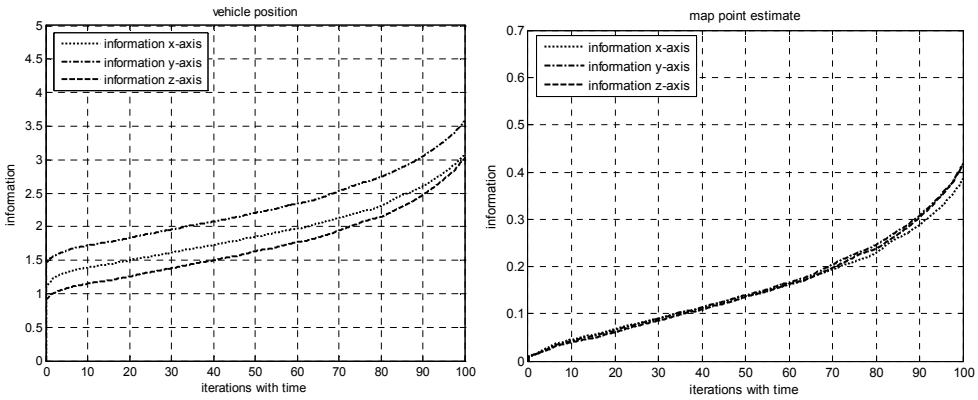

Figure 6.5: Fisher Information Matrix for the vehicle position and the map point estimate (stationary vehicle) without the assumption of no attitude errors

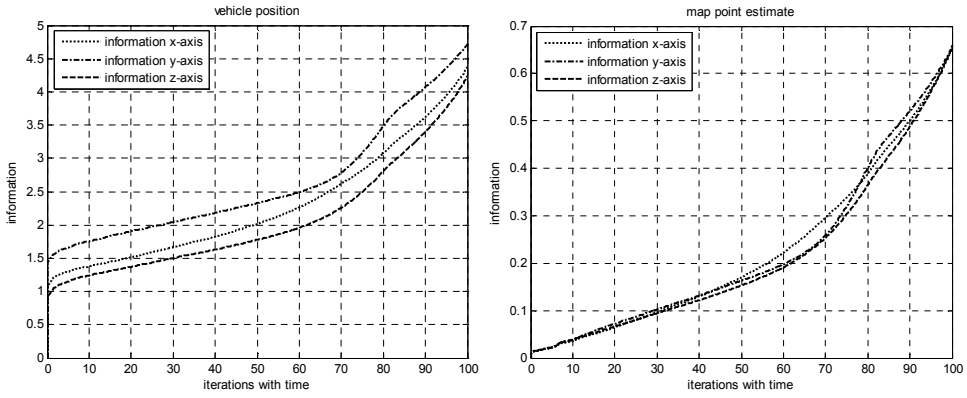

Figure 6.6: Fisher Information Matrix for the vehicle position and the map point estimate (vehicle performs CT maneuver) without the assumption of no attitude errors

Figure (6.5) and (6.6) present the simulation results for the observability analysis of IN aided by SLAM without the assumption of no attitude errors. Figure (6.5) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle is stationary. Figure (6.6) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle performs CT maneuver. If we compare the results we can see that the maneuver is increasing the amount of information content. Figure (6.7) and (6.8) present the simulation results for the observability analysis of IN aided by SLAM with the assumption of no attitude errors.

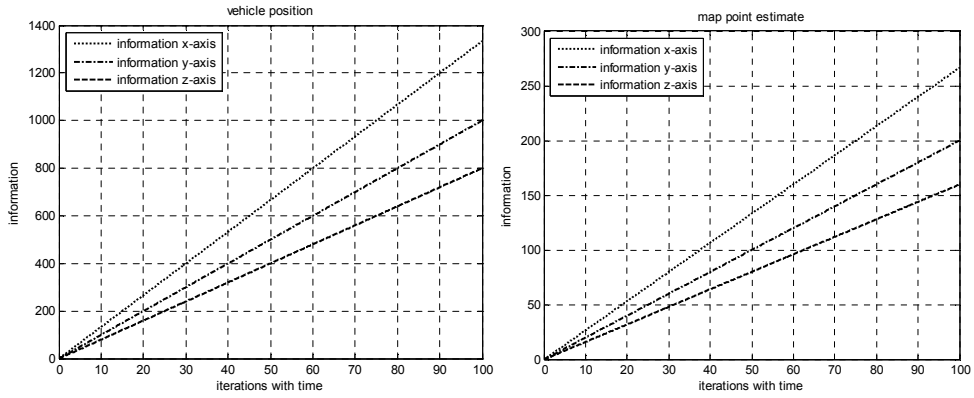

Figure 6.7: Fisher Information Matrix for the vehicle position and the map point estimate (stationary vehicle) with the assumption of no attitude errors

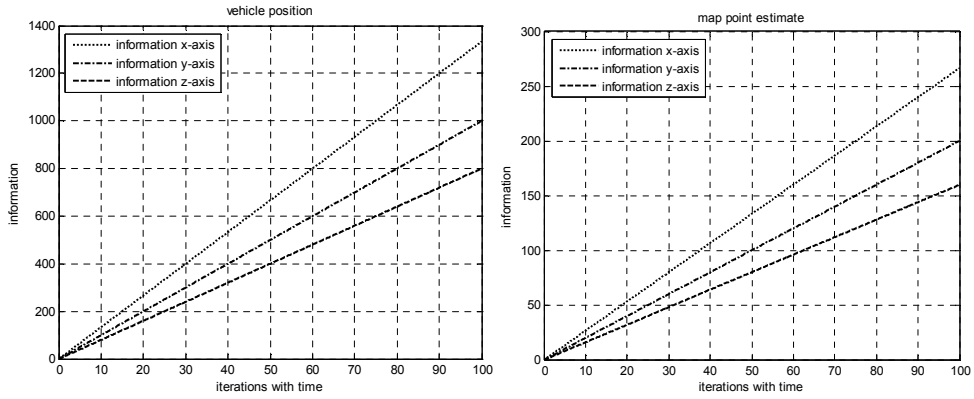

Figure 6.8: Fisher Information Matrix for the vehicle position and the map point estimate (vehicle performs CT maneuver) with the assumption of no attitude errors

Figure (6.7) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle is stationary. Figure (6.8) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle performs CT maneuver. If we compare these results (figure (6.7) and figure (6.8)) with the previous (figure (6.5) and figure (6.6)) we can see that the amount of information content with the assumption of no attitude errors is "blooming" compared with the information content without the assumption of no attitude errors. From this observability analysis conclusion can be drawn that the measurements given by equation (6.24) give more information and will give better results then measurements given by equation (6.15) in the update process of the augmented state vector, see equations (6.30) to (6.34).

6.4 On the Performance of Inertial Navigation Aided by Simultaneous Localization and Mapping Assuming no Attitude Errors

6.4.1 Simulation Scenario and Results

A simulation scenario where the vehicle performs certain maneuvers (circles) around a map point is used to validate the implementation of SLAM as sensor fusion algorithm for aiding IN with vision sensors. The processes of constraining the IN position divergence and reducing the covariance of the map points will be shown. In the simulation scenario the true map point position is $M = [\ 400 \quad 550 \quad 3\ ]^T$.

Real vision sensors i.e. video cameras consist of an illumination source, a lens to gather and focus light, an image detector to capture the image, and an interface to pass the data. The image detector is the heart of the vision sensor. In our simulation we have simulated an image detector with $4mm$ in width and $4mm$ in length. Its performance is perturbed by white noise sequence with standard deviation of $\sigma_c = 0.07mm$, which introduces errors into the camera measurements (observations errors). The lens has focal length of $8mm$.

In the simulation for the measurements errors in the accelerometer and gyro measurements we have assumed that the accelerometer bias α_k^a and the gyro bias α_k^g are not present. Only the accelerometer noise w_k^a and gyro noise w_k^g were simulated. We have taken in consideration that each accelerometer and gyro are perturbed by same white noise sequence. The standard deviation of the white noise sequence for the accelerometers is $\sigma_{w^a} = 0.02236 m/s^2$, whereas the standard deviation of the white noise sequence for the gyros is $\sigma_{w^g} = 0.01732 \deg/s$. The parameters of the video camera and the inertial sensors used in the simulation summarized in a table are as follows:

Name	Notation	Value	Unit
Video camera	White noise std σ_c	0.07	mm
Accelerometer	White noise std σ_{w^a}	0.02236	m/s^2
Gyro	White noise std σ_{w^g}	0.01732	$\deg/s$

Table 6.1: Video camera and inertial sensors performance

The initial vehicle state estimate, vehicle error covariance matrix, process noise matrix, and measurement noise matrix are

$$x_0^v = [\ 555.688 \quad 394.212 \quad 200 \quad 22 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1\ ]^T,$$

$$P_0^{vv} = diag[\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ],$$

$$Q_0^v = diag[\ 2.3 \quad 1.8 \quad 1.3 \quad 0 \quad 0 \quad 0 \quad 0.0003 \quad 0.0003 \quad 0.0003 \quad 0.0003\ ],$$

$$R_0 = diag[\ 0.00005 \quad 0.00005 \quad 0.00005\ ].$$

Note that the initial condition of the vehicle state estimate is usually given as $x_0^v = [\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]^T$ which means that the starting vehicle position is at the coordinate origin and $P_0^{vv} = diag[\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]$ which means that no map point is observed and there is no uncertainty associated with the vehicle yet.

After the start of the simulation using equation (6.12) the state vector is augmented with five map point estimates and its covariance built using equation (6.13). These five map point estimates are shown on figure (6.9). After the augmentation of the state vector with these map positions the vehicle continues to circle around the map point.

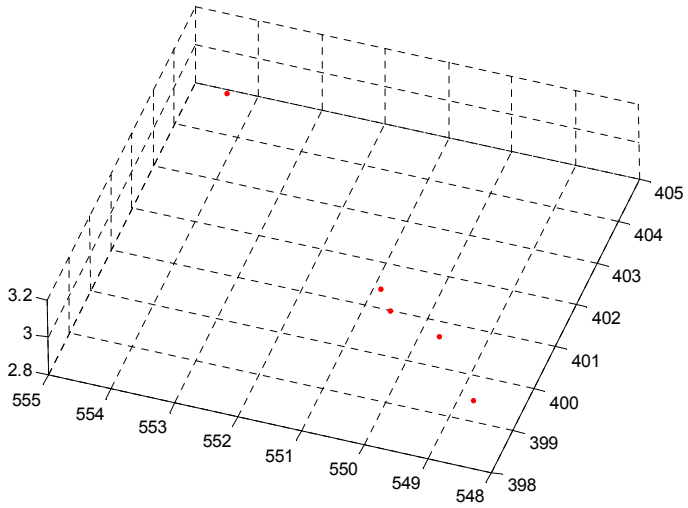

Figure 6.9: Map point estimates

These maneuvers provide the necessary relative measurements for the update stage of the estimation process. Using relative measurements, the EKF updates the augmented state vector as a whole (the vehicle state and each map point estimate). Figure (6.10) shows the uncorrected IN vehicle trajectory (red dashed line) and it can be seen that it is significantly divergent from the true vehicle trajectory (black solid line). Figure (6.11) gives closer look at the true trajectory (black solid line), the estimated trajectory (green dotted line) and the uncorrected divergent IN vehicle trajectory (red dashed line). From this figure it can be seen that the estimated vehicle trajectory is following very closely the true trajectory where it is very difficult to distinguish one from another. Further to the process of constraining the position divergence, the position errors in each axis are shown in figures (6.12), (6.13) and (6.14). We can see that they are bounded, vary cyclically with each encirclement, and have zero mean. Figures (6.15), (6.16), (6.17) show the process of convergence of the map point estimates in the update stage of the estimation process.

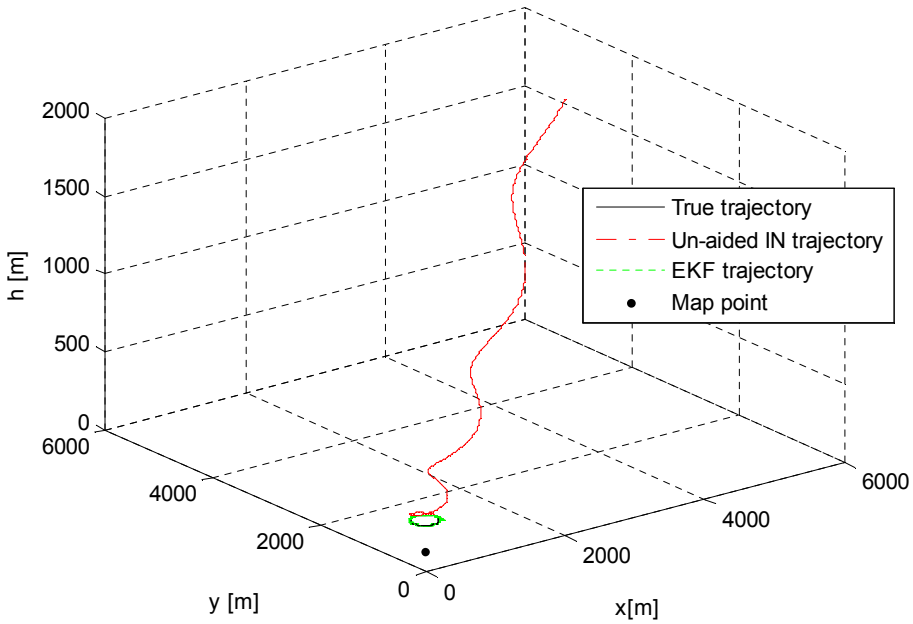

Figure 6.10: True, estimated and divergent vehicle trajectories (four circles around the map point)

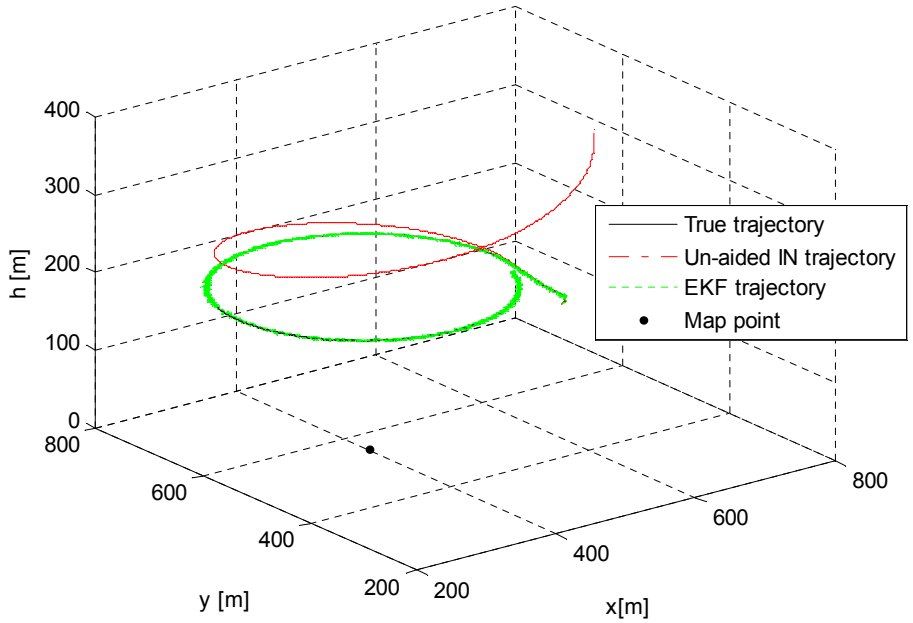

Figure 6.11: True, estimated and divergent vehicle trajectories (circle around the map point)

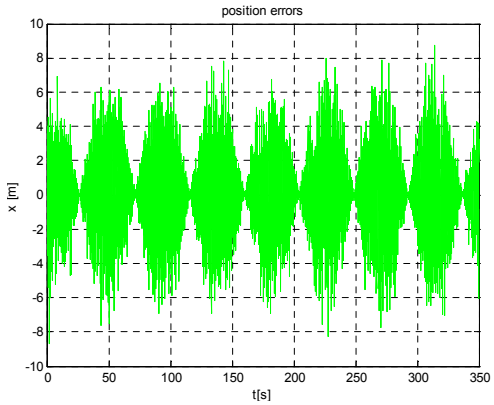

Figure 6.12: Position errors x axis

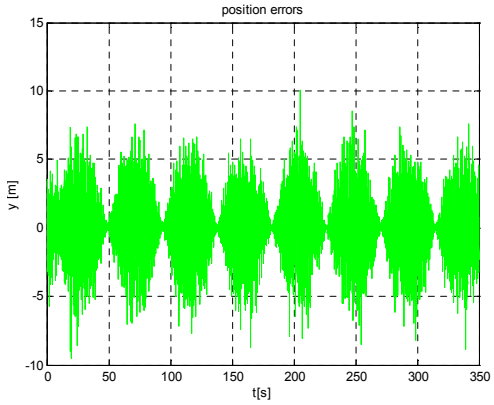

Figure 6.13: Position errors y axis

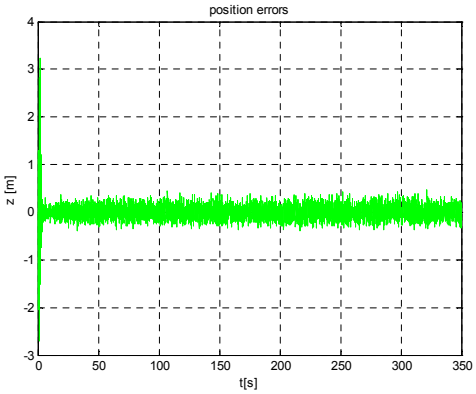

Figure 6.14: Position errors z axis

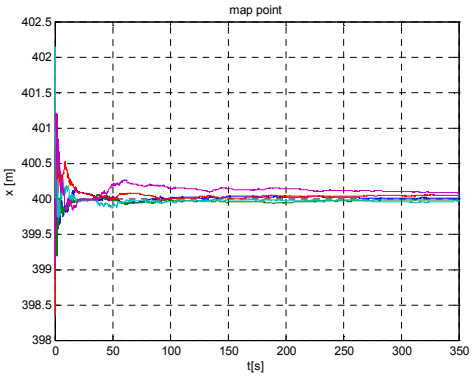

Figure 6.15: Map point estimates x axis

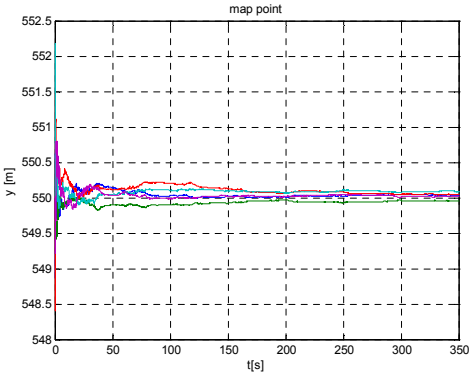

Figure 6.16: Map point estimates y axis

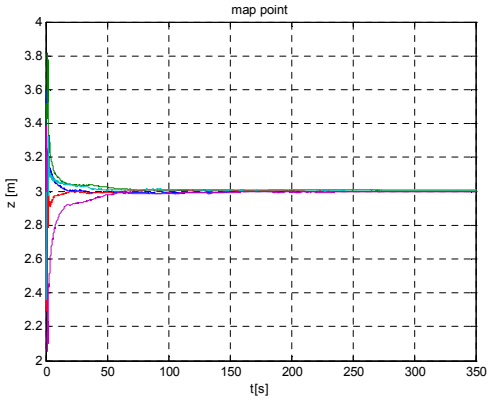

Figure 6.17: Map point estimates z axis

Because of the correlation of the map point estimates, the EKF updates all the map point estimates at the same time. As time progresses and the vehicle circles around the map point (in our case four circles), the map point estimates are updated and all converge to the true map point of $M = [\ 400 \quad 550 \quad 3\ ]^T$.

6.4.2 Interpretation of the results

In the interpretation of the results, we will consider the covariance matrix and its elements. In the analysis of the covariance matrix, two key results from the SLAM theory are assessed to see if they are satisfied by this approach. The first result from [Csorba, 1997] states that: *"The determinant of the covariance matrix of all the map point estimates and the determinant of the covariance matrix of any group of map points estimates are both monotonically non-increasing functions of the time step"*.

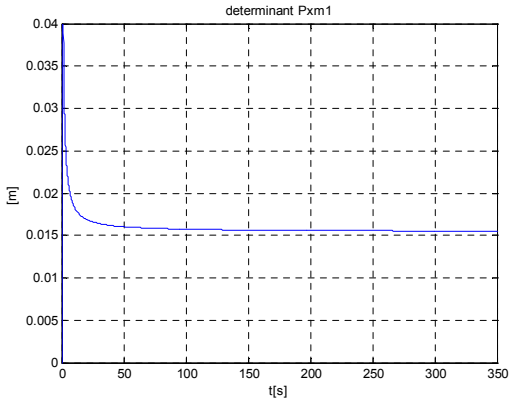

Figure 6.18: Determinant of the covariance matrix of the first map point estimate

It can be seen from figure (6.18) that the determinant of the covariance matrix of the first map point estimate decreases monotonically and therefore satisfies the first result. This result also applies to all the other map point estimates.
Figure (6.19) shows the 95% contour ellipsoid of the covariance matrix of the first map point estimate at the beginning and at the end of the simulation. We can see that the covariance is reduced by about half. The second result from [Csorba, 1997] states that: *"The relative distance vectors between all possible pairs of map point estimates are monotonically decreasing"*.

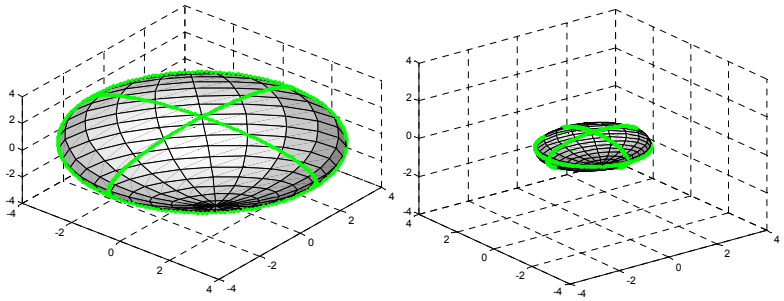

Figure 6.19: The 95% contour ellipsoid of the covariance matrix of the first map point estimate

As can be seen from figure (6.20), the relative distance between the first and second map point estimates is monotonically decreasing and satisfies the second result. These results apply for all the other map point estimates.

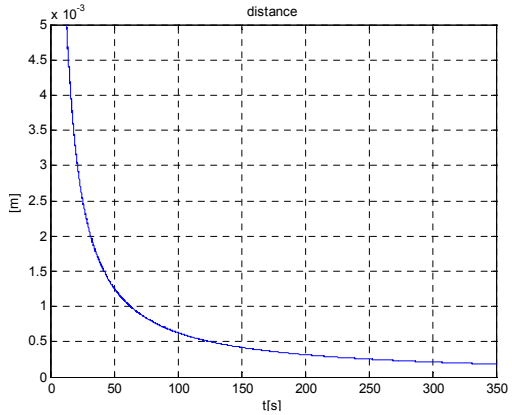

Figure 6.20: The relative distance between the first and second map point estimates

Next figure (6.21) shows the correlation coefficient between the first and second map point estimate. As we can see from the beginning of the simulation, the correlation coefficient very fast reaches the limit (perfect positive correlation) and stays there till the end of the simulation. This result applies for the rest of the correlation coefficients of the map point estimates. Following the proof in section 6.2 regarding the convergence of the map point estimates, figure (6.21) assures that at the limit (perfect positive correlation) when one map point estimate is updated all of the map point estimates are updated.

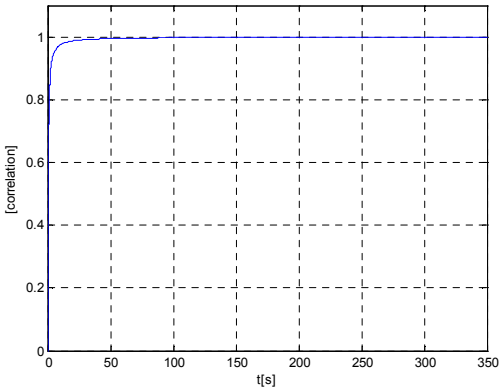

Figure 6.21: Correlation coefficient between the first and second map point estimate (x-axis)

Further to the comparison with the SLAM theory, we should also analyze the covariance matrix of the vehicle. In [Csorba, 1997] it is stated that *"The determinant of the covariance matrix of the vehicle is not monotonic and is evolving as separate unit to the map covariance"*. We can see from figure (6.22) that the determinant of the covariance matrix of the vehicle position is not monotonic but has sinusoidal shape.

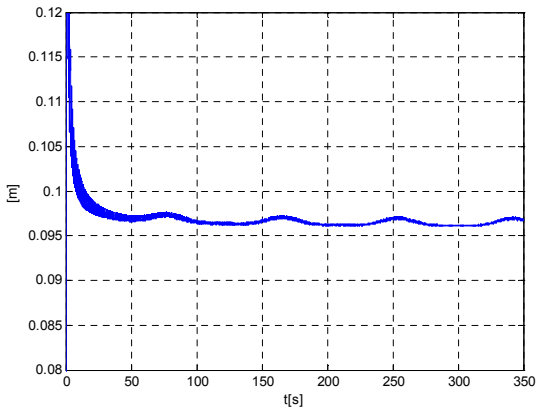

Figure 6.22: Determinant of the covariance matrix of the vehicle position

Figure (6.23) shows the 95% contour ellipsoid of the covariance matrix of the vehicle position at the beginning and at the end of the simulation.

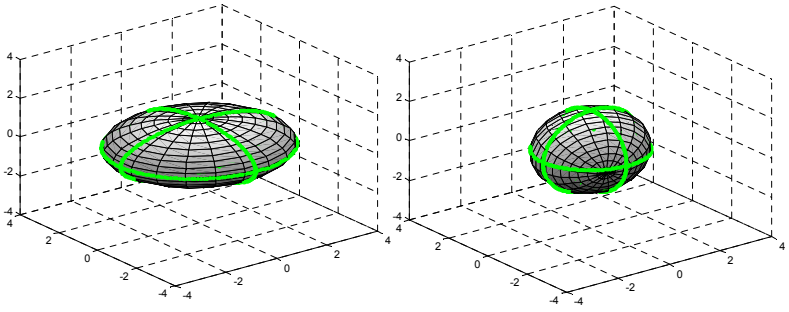

Figure 6.23: The 95% contour ellipsoid of the covariance matrix of the vehicle position

Chapter 7

Inertial Navigation Aided by Bearing Only Simultaneous Localization and Mapping

Bearing Only Simultaneous Localization and Mapping (BOSLAM) is very attractive these days because it permits the use of single camera as sensor for measuring the bearing i.e. unit direction[1] to the map points. BOSLAM as solution to the SLAM problem when single camera is used has great potential for autonomous navigation. Major importance comes from the fact that the single camera address many of the critical requirements and needs of the aerial systems, for example the requirement for limited payload, then the need for low cost and low power consumption sensors. The major drawback of this solution is the problem of map point initialization from a single measurement. In this chapter following the concept of aiding IN by SLAM presented in chapter 6 and the results given there for aiding IN by BOSLAM we augment the state vector with certain number of estimates for each map point. We take repeated measurements (unit direction vectors) of the map points with certain maneuvers in vicinity of the map point. It is shown that it is possible to constrain the Inertial Navigation position divergence and to eliminate some of the uncertainty of the map point estimates. This concept brings new parameterization of the map point in BOSLAM.

[1] the terms unit direction and bearing will be used synonymously in this thesis

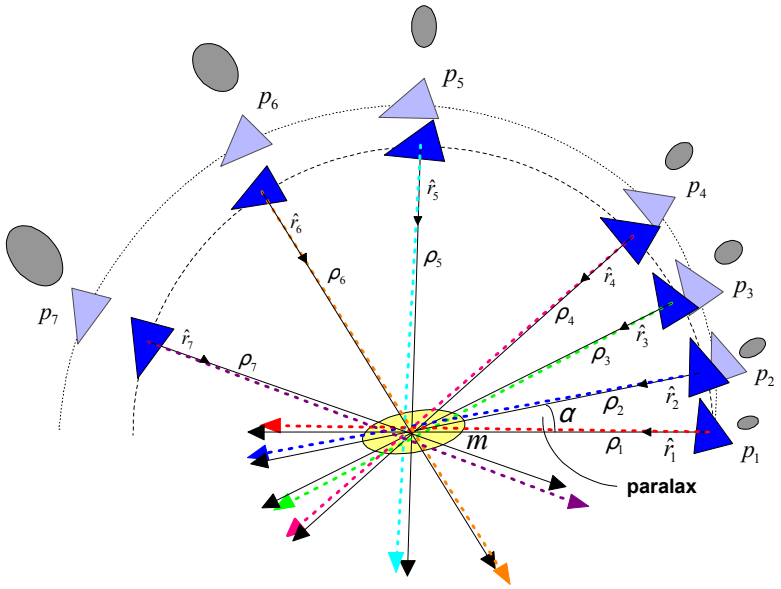

Figure 7.1: IN aided by BOSLAM with repeated observations of the map point with a circular maneuver

7.1 Bearing Only Simultaneous Localization and Mapping as Sensor Fusion Algorithm

As in chapter 6 in IN aided by BOSLAM the IN mechanization equations represent the vehicle state and are given by equation (6.3). The vehicle is equipped with single camera which provides unit direction vectors represented by observation models (6.21) and (6.27), see chapter 3 for details.

In IN aided by BOSLAM as in BOSLAM the range to the map points is not available so we cannot compute the map point estimates directly. From figure (7.1) we see that this is not a problem. The range can be considered as constant real number which in theory can have a value from zero to infinity $\rho \in [0, \infty]$. Typically we will constrain the range into an interval for a given application and will have values from zero to the maximum range $\rho \in [0, \rho_{\max}]$. Using the equation for computing the map point estimates, equation (6.1) and moving the range from zero to the maximum range in some units (can be in meters or parts of meters) $\rho = 0, 1, 2, 3, \ldots, m$ instead of having single map point estimate we will have m

map point estimates from each vehicle position $p_1, p_2, \ldots, p_n$:

$$
\begin{aligned}
x_k^{m1} &= [\ x_k^{m10^T} \quad x_k^{m11^T} \quad x_k^{m12^T} \quad \ldots \quad x_k^{m1m^T}\ ]^T \\
x_k^{m2} &= [\ x_k^{m20^T} \quad x_k^{m21^T} \quad x_k^{m22^T} \quad \ldots \quad x_k^{m2m^T}\ ]^T \\
&\vdots \\
x_k^{mn} &= [\ x_k^{mn0^T} \quad x_k^{mn1^T} \quad x_k^{mn2^T} \quad \ldots \quad x_k^{mnm^T}\ ]^T.
\end{aligned}
\tag{7.1}
$$

Calculation of each of the elements in equation (7.1) in more general form is

$$
x_k^{mij} = \begin{bmatrix} x^{ij} \\ y^{ij} \\ z^{ij} \end{bmatrix} = \begin{bmatrix} x_i + j * \hat{r}_{xi} \\ y_i + j * \hat{r}_{yi} \\ z_i + j * \hat{r}_{zi} \end{bmatrix}
\tag{7.2}
$$

where $p_i = [\ x_i \quad y_i \quad z_i\ ]^T \quad i = 1, 2, 3, \ldots, n$ are the vehicle positions and j is a parameter that moves from zero to the maximum range i.e. $j = 0, 1, 2, 3, \ldots, m$. The map point estimates vectors $x_k^{mi}, \quad i = 1, 2, \ldots, n$ given by equation (7.1) of the unknown map point represent the map

$$
x_k^m = [\ x_k^{m1^T} \quad x_k^{m2^T} \quad \ldots \quad x_k^{mn^T}\ ]^T
\tag{7.3}
$$

The augmented state vector containing the vehicle state and the map stays same as in IN aided by SLAM, see equation (6.9):

$$
x_k = [\ x_k^{v^T} \quad x_k^{m1^T} \quad \ldots \quad x_k^{mn^T}\ ]^T
\tag{7.4}
$$

In this way we have succeeded in maintaining the mapping process within IN aided by BOSLAM. Also we have succeeded to maintain the method for augmenting the map point estimates, equation (6.12) and building the covariance matrix, equation (6.13) which ensured the consistency of the map. The Jacobian ∇g_k^x with respect to the augmented state x_k stays same as in section 6.1. The Jacobian ∇g_k^z with

respect to the observation z_k is changed and is given in Appendix G. One issue appears at this moment and that is the huge dimension of the augmented state vector. This requires bigger memory and will bring additional computational cost in the algorithm. However, since we are not updating the augmented state vector x_k at this moment, this is not a problem. In order to solve it we need an algorithm or criterion that will choose a certain number of map point estimates and will remove the other from the augmented state vector. This issue would be a critical aspect of a real application. Simple approach is first to calculate an map point estimate using equation (3.10) from two consecutive vehicle positions and then to search for its nearest neighbor in the augmented state vector with some of the nearest-neighbor search algorithms. Algorithms such as the simple nearest-neighbor search which performs best for small number of map points (say $n \leq 100$) or other more sophisticated algorithms such as the kd-tree or Voronoi diagrams can be implemented as suggested in [Skiena, 2008]. After choosing the "most promising" map point estimates and removing the other from the augmented state vector as the vehicle circles around the map point and provides repeated observations of the map point, we can begin updating the augmented state vector x_k by using one of the map points estimates at a time. As in IN aided by SLAM the Extended Kalman filter (EKF) can be implemented, equations (6.29) to (6.34). The matrix ∇h_k^x i.e. the Jacobian of the nonlinear observation function $h(\cdot)$ (6.21) and (6.27) with respect to the state x_k evaluated at $\hat{x}_{k|k-1}$ i.e. the vehicle state $\hat{x}_{k|k-1}^v$ and the map point estimate used in the update $\hat{x}_{k|k-1}^{mi}$. The Jacobians ∇h_k^x are given in Appendix H and Appendix J, respectively.

7.2 On the Map Point Initialization and Parametrization in Bearing Only Simultaneous Localization and Mapping

Much of the research in BOSLAM is focused towards the problem of initialization of the map points from single camera measurements. In the literature two techniques are proposed to address the problem of map point initialization. The first technique involves delaying the map point initialization until a criterion is

fulfilled and sufficient baseline is available from different vehicle positions to initialize the map point [Davison, 2003], [Deans and Hebert, 2000], [Bailey, 2003], [Bryson and Sukkarieh, 2008b]. The second technique tries to avoid the delay and initialize the map point from a single measurement [Vidal-Calleja et al., 2007], [Sola et al., 2005]. The fact that after the first observation, the map point lies along on the line from the vehicle to the map point (the projection ray) is used. The range along the line is not known and uncertain and can be modeled in the range between the minimum and maximum range. In [Vidal-Calleja et al., 2007] multiple hypotheses are used each with different range. In the augmented state vector each hypothesis is treated as a separate map point. As successive observations of the map point are made all but one hypothesis (the most promising) is removed from the augmented state vector. In [Sola et al., 2005], since the range is unknown, they initialize the map point estimate to lie between the minimum and maximum range for their application and create on the optical ray a priori uniform pdf. Then they approximate the a priori pdf with a sum of Gaussians. After additional observations of the map point they prune the less likely Gaussians of the ray leaving only one and declare this as a map point. In [Montiel et al., 2006] authors proposed a new unified parameterization for the map points within BOSLAM. They use the direct parameterization of the inverse depth (inverse of the range to the map point). This is based on the fact that due to the inverse depth parameterization the measurement equation has low linearization error and thus allows estimation uncertainty to be accurately modeled as Gaussian. This method doubles the size of the map point state which increases the computational complexity and it suffers from the issue of negative depth. A solution to the latter problem is proposed in [Parsley and Julier, 2008].
The approach proposed in section 7.1 to augment the state vector not only with one map point estimate but with certain number of estimates for each map point brings new parameterization of the map point in BOSLAM. The novelty comes from the usage of the certain number of map point estimates for update of the whole augmented state vector together with a combination of repeated measurements and motion in vicinity of the map point. This specific update process constrains the vehicle position divergence and makes the map point estimates to converge to the true map point. This approach brings delayed initialization of

the map points since it uses number of vehicle positions for choosing the "most promising" map point estimates augmented in the state vector. The combination of motion and repeated measurements for update of the whole augmented state vector distinguishes this approach from the before mentioned initialization approaches.

7.3 On the Observability Analysis of Inertial Navigation Aided by Bearing-Only Simultaneous Localization and Mapping using Fisher Information Matrix

The observability analysis of the IN aided by BOSLAM follows the observability analysis of IN aided by SLAM. Very similar to section 6.3, we study the information theoretical point of view of the observability analysis of IN aided by BOSLAM. The Fisher Information Matrix is used to quantify the information content provided by the applied observation models. Scenarios with stationary and maneuvering vehicle trajectories are used to better understand, compare and provide realistic measurement sequences for determining the best achievable estimation performance. The differences between this observability analysis and the observability analysis of the IN aided by SLAM section 6.3, are as follows. With the vision sensor (single camera) we are taking repeated measurements (6.21) of the map point. The Jacobian of the nonlinear function (6.21) with respect to the augmented state vector x_k is given in Appendix H. When assuming no attitude errors with the vision sensor (single camera) we are taking repeated measurements (6.27) of the map point. The Jacobian of the nonlinear function (6.27) with respect to the augmented state vector x_k is given in Appendix J.

7.3.1 Simulation Scenario and Results

A simulation scenario where the vehicle is stationary and where it performs the CT maneuver around a map point are used for observability analysis of IN aided by BOSLAM without and with the assumption of no attitude errors, see figure

(6.3) and (6.4) from section 6.3.5. Figure (7.2) and (7.3) present the simulation results for the observability analysis of IN aided by BOSLAM without the assumption of no attitude errors. Figure (7.2) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle is stationary. Figure (7.3) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle performs CT maneuver. If we compare the results we can see that the maneuver is increasing the amount of information content.

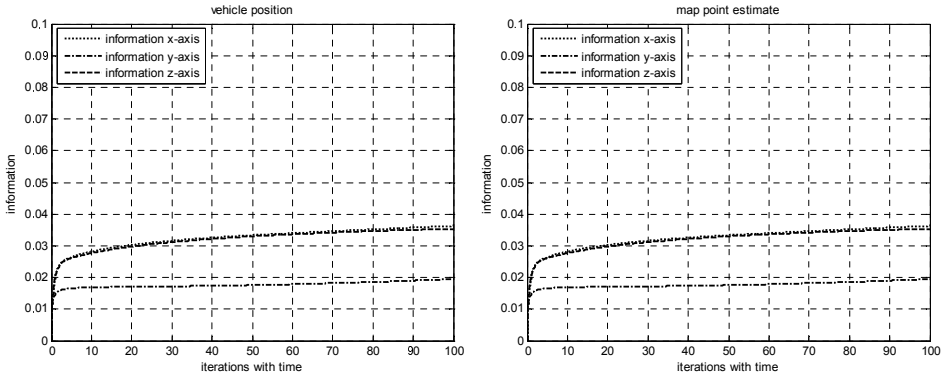

Figure 7.2: Fisher Information Matrix for the vehicle position and the map point estimate (stationary vehicle) without the assumption of no attitude errors

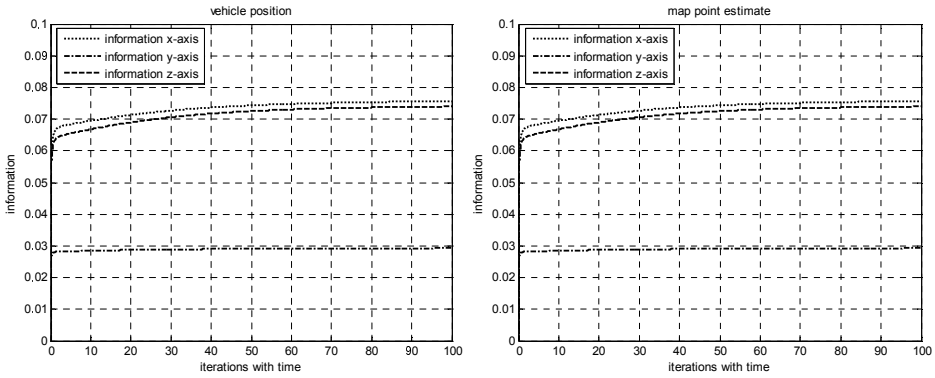

Figure 7.3: Fisher Information Matrix for the vehicle position and the map point estimate (vehicle performs CT maneuver) without the assumption of no attitude errors

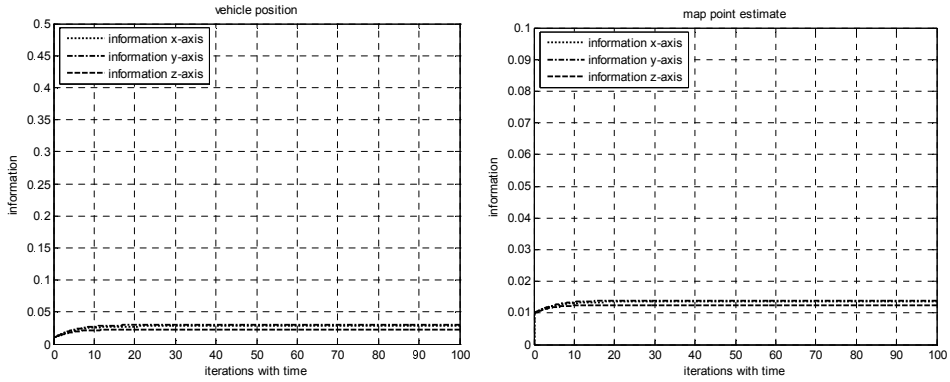

Figure 7.4: Fisher Information Matrix for the vehicle position and the map point estimate (stationary vehicle) with the assumption of no attitude errors

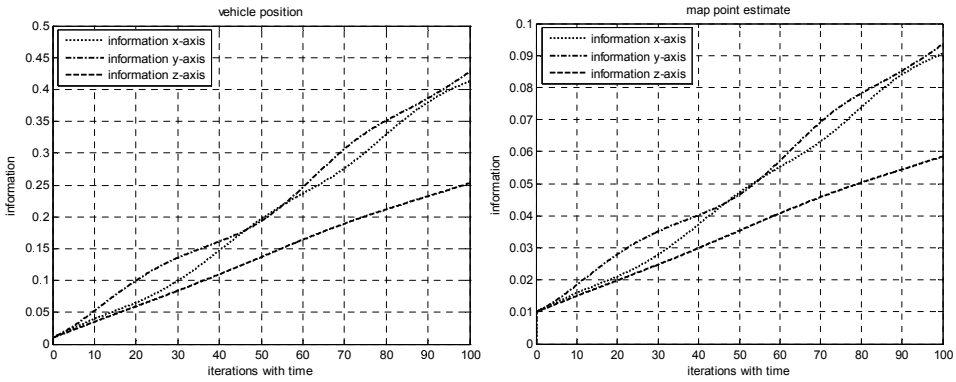

Figure 7.5: Fisher Information Matrix for the vehicle position and the map point estimate (vehicle performs CT maneuver) with the assumption of no attitude errors

Figure (7.4) and (7.5) present the simulation results for the observability analysis of IN aided by SLAM with the assumption of no attitude errors. Figure (7.4) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle is stationary. Figure (7.5) shows the information content for the vehicle position and the map point estimate in the scenario when the vehicle performs CT maneuver. If we compare the results we can see that the maneuver is increasing the amount of information content. In addition we have bigger amount of information content with the assumption of no attitude errors. From this observability analysis conclusion can be drawn that the measurements given by equation (6.27) give more information and will give better results then measurements given by equation (6.21) in the update process of the augmented state vector, see equations (6.29) to (6.34).

7.4 The effect of nonlinearities in Inertial Navigation Aided by Bearing-Only Simultaneous Localization and Mapping

IN aided by BOSLAM as well as many other navigation problems are nonlinear and must be linearized (approximated) before applying the popular Kalman-like filtering algorithms. An EKF presents one such approximation. The EKF is very commonly used algorithm and, because of its simplicity, is very often chosen as the "best" algorithm for implementation.
EKF will be the "best" choice as long as deviations from the true trajectory are "small" (say in the least square sense), and the higher order terms in the Taylor series expansion are negligible. In a case where the higher order terms in the Taylor series expansion are large they have a biasing effect on the estimate [Jazwinski, 1970]. Inspection of the approximate nonlinear filters reveals the size of nonlinearities depends not only on the size of the second order partial derivatives (f_{xx} and h_{xx}), but also on the estimation error variance. The second order terms appear as Ph_{xx} in the case of measurement nonlinearity, and as Pf_{xx} in the case of system nonlinearity. These are the expected values of the second order terms in the Taylor series expansion. As a consequence, these nonlinear terms

can be large because the second partial derivatives are large (real nonlinearity), or because the estimation error variance is large (induced nonlinearity), or both (mixed nonlinearity) [Jazwinski, 1970].

7.5 On the Performance of Inertial Navigation Aided by Bearing-Only Simultaneous Localization and Mapping

As with section 6.4 when the performance of IN aided by SLAM assuming no attitude errors was validated, for IN aided by BOSLAM the same simulation scenario is used. The true map point position is same $M = [\ 400 \quad 550 \quad 3\ ]^T$. The parameters of the video camera and the inertial sensors used in the simulation scenario are same as in Table 6.1. The initial vehicle state estimate, vehicle error covariance matrix, process noise matrix, and measurement noise matrix are

$$x_0^v = [\ 555.688 \quad 394.212 \quad 200 \quad 22 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1\ ]^T,$$

$$P_0^{vv} = diag[\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ],$$

$$Q_0^v = diag[\ 0.8 \quad 0.8 \quad 0.2 \quad 0 \quad 0 \quad 0 \quad 0.0003 \quad 0.0003 \quad 0.0003 \quad 0.0003\ ],$$

$$R_0 = diag[\ 0.00005 \quad 0.00005 \quad 0.00005\ ].$$

Note that the initial condition of the vehicle state estimate is usually given as $x_0^v = [\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]^T$ which means that the starting vehicle position is at the coordinate origin and $P_0^{vv} = diag[\ 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\ ]$ which means that no map point is observed and there is no uncertainty associated with the vehicle yet. After the start of the simulation the process of initialization of the map point estimate starts as described in in section 7.1. After the initialization, investigation of the EKF, IEKF, UKF and UPF update equations is performed. Figure (7.6) shows the true, IN divergent and estimated EKF, IEKF, UKF and UPF vehicle trajectories. Figure (7.7), (7.8) and (7.9) show position errors in each axis. Figure (7.10), (7.11) and (7.12) show the map point estimates in each axis with EKF, IEKF, UKF and UPF implementation.

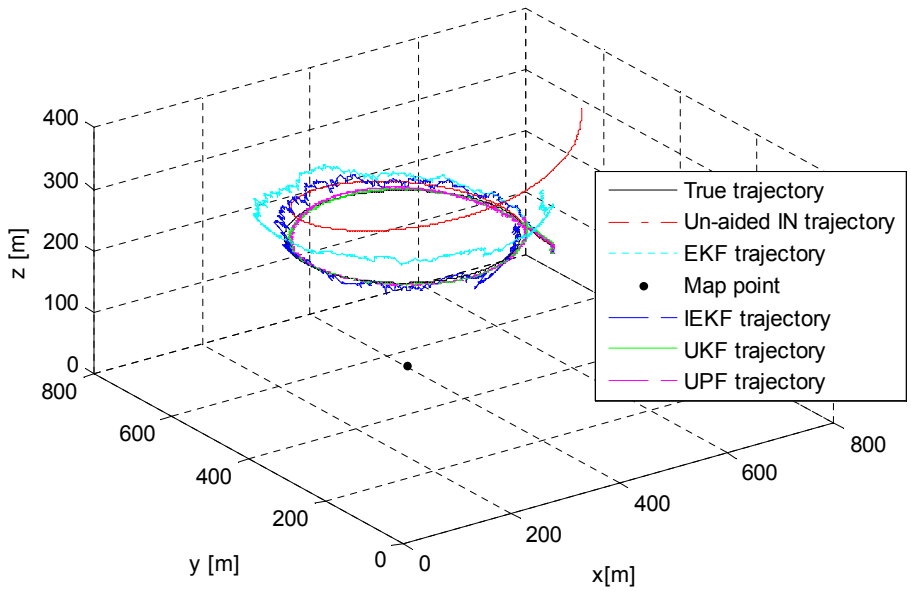

Figure 7.6: True, estimated and divergent vehicle trajectories

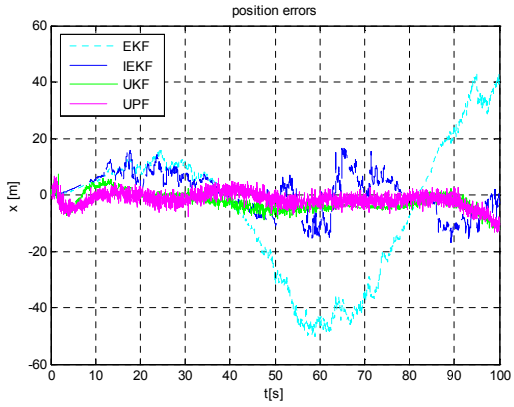

Figure 7.7: Position errors x axis

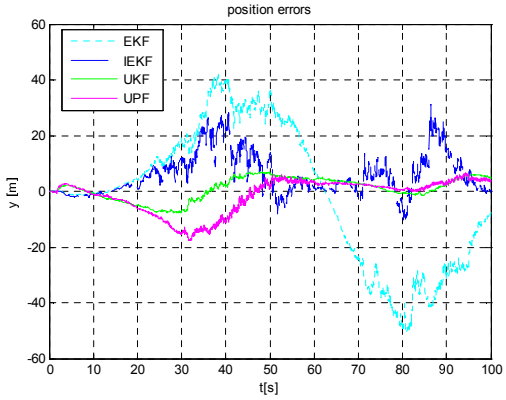

Figure 7.8: Position errors y axis

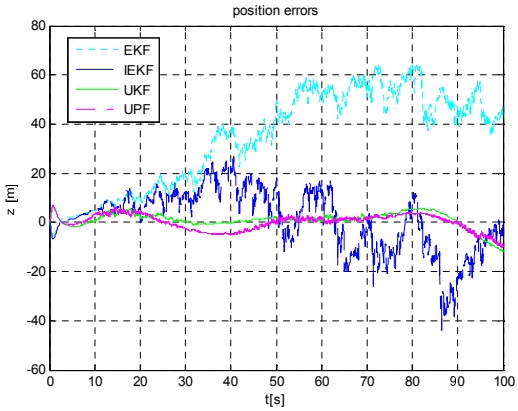

Figure 7.9: Position errors z axis

These simulation results show the performance of the four nonlinear filters (EKF, IEKF, UKF and UPF). The problem of IN aided by BOSLAM exhibits a high degree of nonlinearity. In these applications the EKF introduces large estimation errors (observe the vehicle position error variance on figure (7.7), (7.8) and (7.9). IEKF shows improvements in the results but still the position error variance is large. UKF and UPF demonstrate best performance and appears to be an efficient estimators for the problem of IN aided by BOSLAM. While the UKF and UPF produce good performance with the vehicle state they exhibit large initial variance with the map point estimates and converge slowly then the EKF and IEKF, see figure (7.10), (7.11) and (7.12) as well as (7.15) and (7.16).

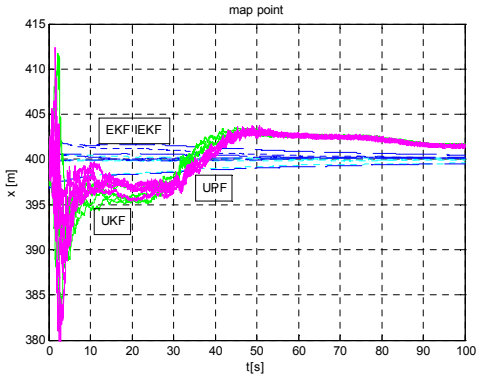

Figure 7.10: Map point estimates x axis

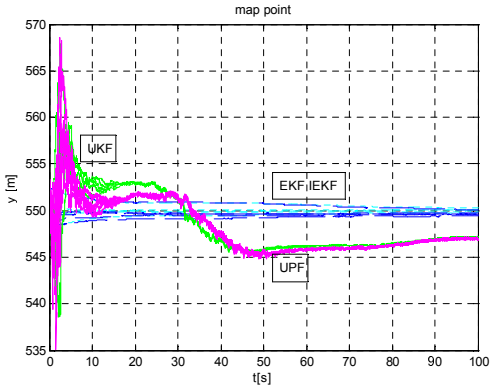

Figure 7.11: Map point estimates y axis

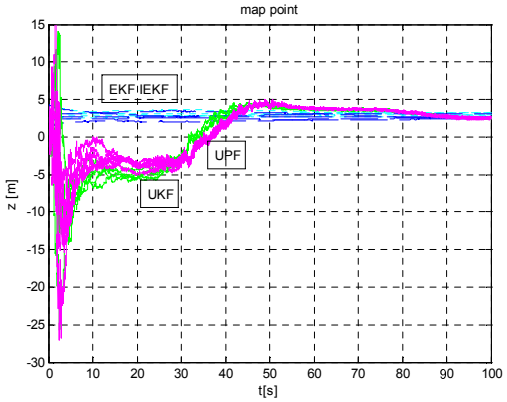

Figure 7.12: Map point estimates z axis

The UPF show small improvements then the UKF, as the map point estimates try slightly faster to converge to the true map point. Both EKF and IEKF show similar performance with the map point estimates and slowly converge to the true map point, (7.13), (7.14). The computational load of the four algorithms should be noted. The simulation results showed that the UKF not only outperforms the EKF and IEKF in accuracy, it also does that with no extra computational cost. The superior performance of the UKF over the EKF and IEKF has been noted in numerous publications. The UPF compared to UKF requires bigger computational power and did not improved the UKF results much. In the simulation scenario the process and measurement noises were modeled as white Gaussian noise sequences and this may be one of the reasons for the noted UPF performance.

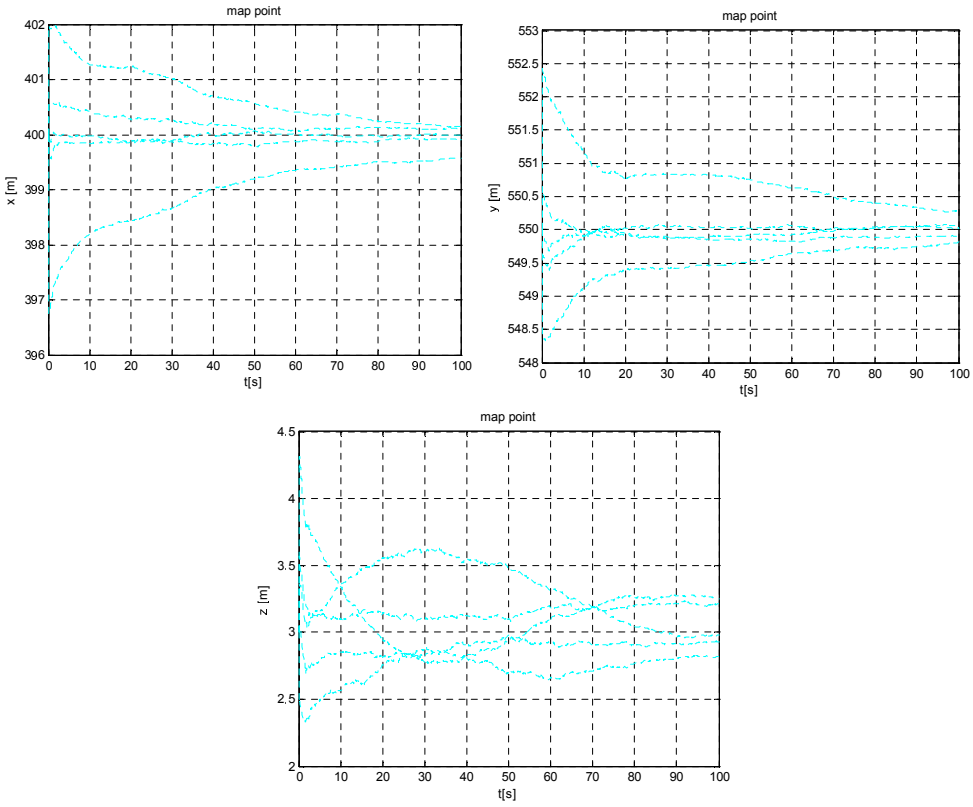

Figure 7.13: Map point estimates EKF implementation

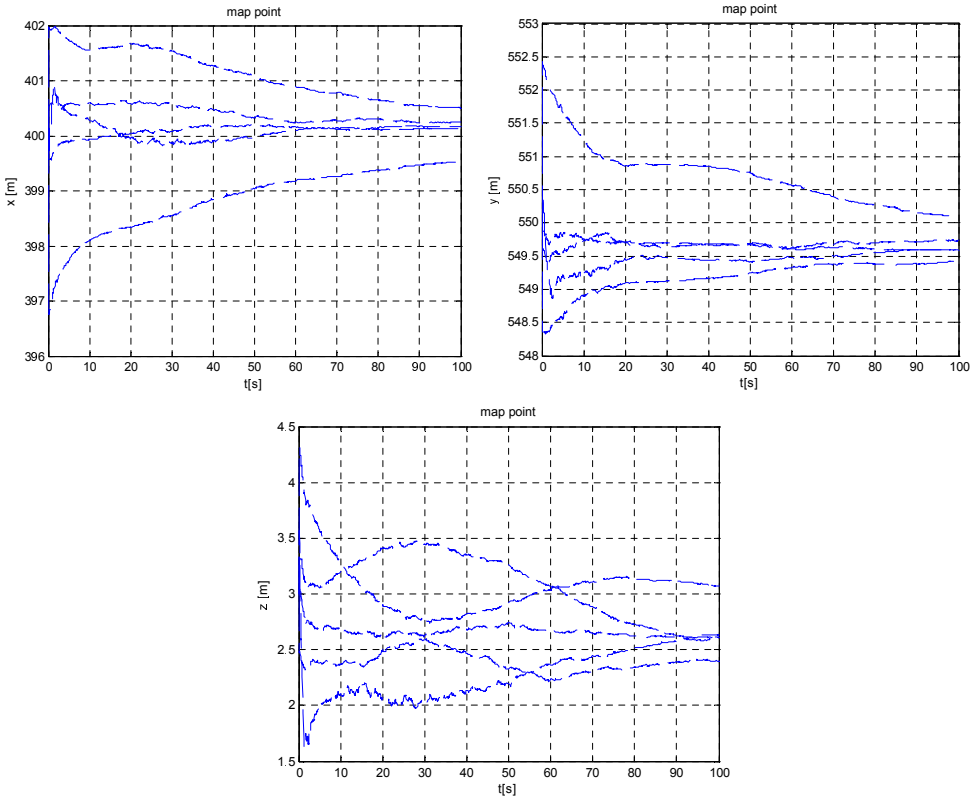

Figure 7.14: Map point estimates IEKF implementation

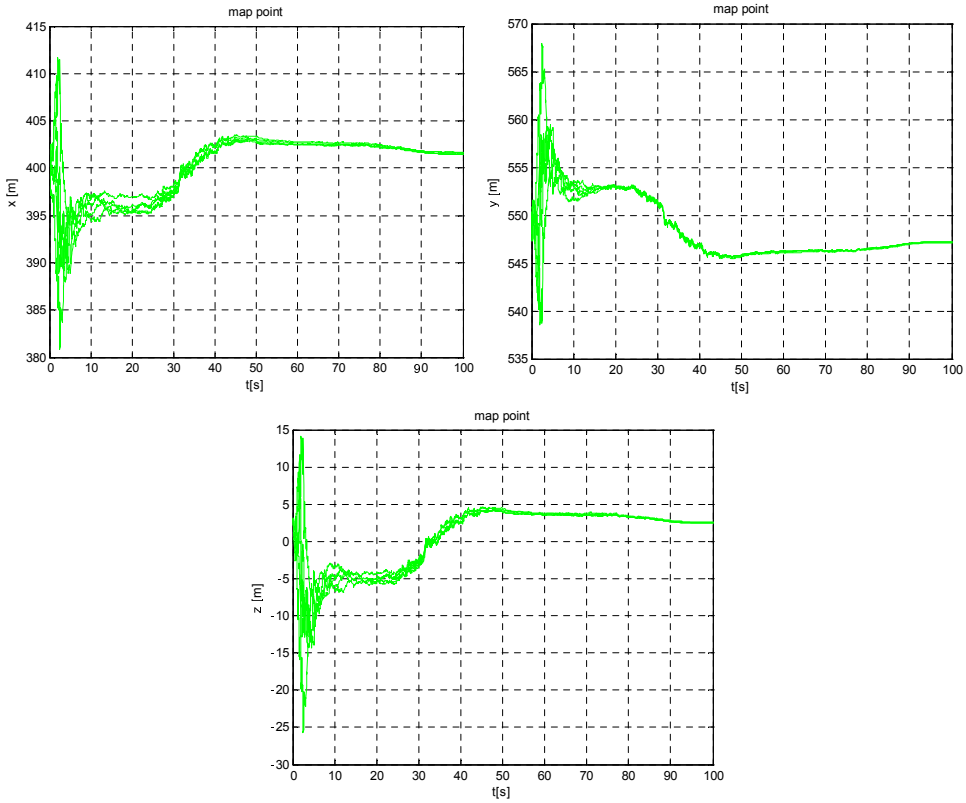

Figure 7.15: Map point estimates UKF implementation

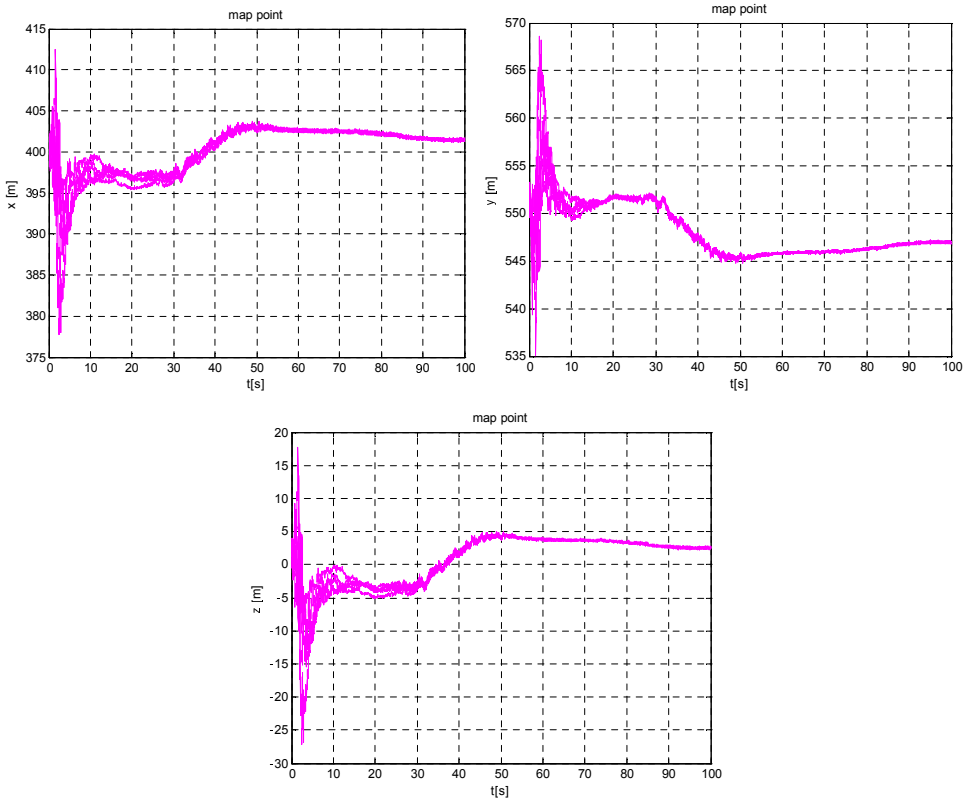

Figure 7.16: Map point estimates UPF implementation

Chapter 8

Conclusions

Many of today's operational UAV navigation systems rely on inertial sensors as a primary measurement source. IN alone however suffers from slow divergence with time. This divergence is often compensated for by employing some additional source of navigation information external to IN. From the 1990's to the present day Global Positioning System (GPS) has been the dominant navigation aid for IN. In a number of scenarios, GPS measurements may be completely unavailable or they simply may not be precise (or reliable) enough to be used to adequately update the IN hence alternative methods of aiding IN have seen great attention. Aiding IN with vision sensors has been the favoured solution over the past several years.

In this thesis we address the IN position divergence and propose solutions (i.e. sensor fusion algorithms) that provide a source of reliable aiding information to IN from vision sensors. Using vision sensors and with no a priori knowledge of the environment we show that, with certain manoeuvres around a map point, it is possible to both constrain the IN position divergence and to reduce some of the uncertainty at the map point estimates.

In chapter 3 we used vectors to explain the vision sensors geometry. Many of the authors in SLAM use trigonometric functions in their observation models. The use of trigonometric functions has, apart from the fact that they may have singularities at certain points, the disadvantage of making computer processing slow. In this thesis each measurement from the vision sensors is represented by a line. Instead of using angles like bearing or azimuth to the map points we use

unit direction vectors to the map points. This novel approach presents an important trigonometric substitution and is more efficient in terms of computational speed and accuracy. Further the use of vectors for the vision sensors geometry (see sections 3.1 and 3.2), contributes to the SLAM theory in using vectors in the vision sensors observation models.

In chapter 6 we propose changes to the SLAM augmented state vector in order to take repeated measurements of the map point and implement SLAM for aiding IN. The novel contributions in this chapter are based on the following. Certain movements of the vehicle around or nearby a map point provide valuable information's of the spatial relationship of the map point and the vehicle. SLAM theory being interested in how to build a map of the environment and simultaneously to use the map to compute the vehicle position does not use these information's. As the vehicle moves and circles, the map point is observed from the vehicle positions. The problem arises that the estimated vehicle positions differ from the true vehicle positions due to accumulated uncertainty in the vehicle position. Because of this accumulated uncertainty in the vehicle position and the measurement errors we cannot compute the map point position exactly. Instead from each vehicle position an map point position estimate can be calculated. This requires the spatial variables i.e. the map point estimates and the vehicle states be treated as an intrinsic part of the spatial representation. The general representation used in SLAM where vehicle states are augmented with the map point estimates and where the associated covariance matrix represents the uncertainty of each of the map point estimates and their inter-dependencies is used. This representation brings different initialization of the map point and we refer to it as concept of aiding IN by SLAM. The approach proposed is to augment the state vector not only with one map point estimate but with certain number of estimates for each map point. After the augmentation of the state vector with manoeuvring, for example circling around the map point and using the relative measurements from the vision sensors and updating the augmented state vector the vehicle position divergence is constrained and the uncertainty of the map points is reduced, as shown in section 6.4. There the key results from the SLAM theory are assessed to see if they are satisfied by this novel approach. Further to the concept of aiding IN by SLAM, in this thesis we have contributed

and investigated how a bearing only sensor such as single camera can be used for aiding IN. The results of IN aided by SLAM were used. New parameterization of the map point in BOSLAM is proposed in chapter 7. This new parametrization uses the map point estimates for update of the whole augmented state vector with a combination of repeated measurements and motion in vicinity of the map point. This concept of aiding IN by BOSLAM in parallel with bounding the position errors also bounds the velocity errors as shown in [Silson and Sazdovski, 2011]. Aiding IN by BOSLAM exhibits a high degree of nonlinearity and typically in these applications an EKF introduces large estimation errors. Because of this and a number of other significant problems such as Jacobian implementation and the neglect of the higher order error moments that appear when implementing the EKF, other algorithms such as IEKF, UKF and UPF were implemented. It is shown in section 7.5 that the UKF and UPF demonstrate best performance and appear to be efficient estimators for the concept of IN aided by BOSLAM.
The SLAM aided IN and BOSLAM aided IN sensor fusion algorithm present reliable solutions that provide aiding information to IN from vision sensors. These algorithms successfully integrate the inertial and vision sensors with no a priori knowledge of the environment. In section 6.4, when the performance of the IN aided by SLAM was assessed, we find out that such an integrated navigation systems require further integration/coordination with the guidance and control measurements and the vehicle task itself to perform the needing manoeuvres and to achieve the needing navigation accuracy. Further with the observability analysis of the IN aided by SLAM and IN aided by BOSLAM (section 6.3 and 7.3), we have shown that manoeuvres increase the amount of information content. This means that for certain or desired accuracy of the navigation parameters the manoeuvres are essential. Simply passing or flying by a map point with no manoeuvre will not help much for autonomous navigation. These facts bring new challenges to the practical design of these modern jam proof GPS free autonomous navigation systems.

Chapter 9

Future Work

Our future research work will be focused on the practical aspects of the both SLAM aided IN and BOSLAM aided IN sensor fusion algorithms. We will work on the practical implementation of the proposed concepts and propose to validate these algorithms on aerial vehicle. Quadrotor UAV is the chosen platform for the practical experiments.

In the thesis the simulation model process and measurement noises were modelled as white Gaussian noise sequences. We would like our simulation model to better represent the "harsh" real environments in which the proposed integrated navigation systems will operate, but we are aware that this is very difficult to achieve and we expect that our model to a confident degree is appropriate representation of the reality. We are aware that the non-Gaussian nature of the input noise and the measurements when the practical experiments will be carried out may lead to different results from the simulated one. From our previous practical experiences we expect that the practical experiments of the proposed algorithms will follow the simulation results. We are looking at the nonlinear filters (UKF and UPF) to achieve satisfactory estimation accuracy and to appear as efficient estimators for the both SLAM aided IN and BOSLAM aided IN.

Further we are looking forward the implementation of the Rao-Blackwellized Particle Filter. The idea is to partition the augmented state vector so that the map partition can be worked out analytically using the EKF (note the satisfactory performance of the map point estimates within the EKF framework in section 7.5, figure (7.13)). The Particle Filter is to be used for the IN partition of the

state vector. Rao-Blackwellization is widely accepted variance reduction method and is potentially applicable in both SLAM-aided IN and BOSLAM-aided IN. The experiences with RBPF given in chapter 12 [Ristic et al., 2004] and chapter 24 [Doucet et al., 2001] will be used. Use of the SLAM-aided IN algorithm for exploring unknown environments from a practical point of view presents an interesting challenge. The idea here is that, at the beginning of the exploration of the environment, the UAV performs SLAM-aided IN with certain manoeuvres around or nearby a map points. This will constrain both the IN position divergence and will reduce the covariance of the map point estimates. When the desired accuracy of navigation parameters is achieved the UAV can start navigating through the environment and perform SLAM.

The theory side of our future work will include the following two investigations. The first one will investigate and try to develop intelligent manoeuvre strategies which are connected to the energy efficiency of the vehicles. We think that there is no point of UAV circling around a map point, trying to constrain its navigation parameters and spend its whole battery power or fuel. We will be looking for energy gaining manoeuvres that will help the UAV to conserve its energy and use environment factors like the winds, for example.

The second area of future theoretical work is the investigation of cooperation between several UAV's performing SLAM aided IN over same map point. This investigation is expected to address the issues when low accuracy vision sensors are used on UAV's. In this scenario the UAV's can take measurements not just of the map feature but also of each other. These measurements can be used to accelerate the convergence of the localization and mapping processes on both UAV's.

Appendix A

Vector Algebra

This section gives some basic rules. Here we consider how vectors may be used to describe lines and we look at the practical use of vectors in finding distances.

Representation of a line in three dimensional space

Consider the line passing through a fixed point $\boldsymbol{P}$ with position vector $\underline{p}$ and having a direction $\underline{r}$, figure (9.1) [Riley et al., 2006]. It is clear that the position vector $\underline{p}$ of a general point $\boldsymbol{B}$ on the line can be written as

$$\underline{b} = \underline{p} + \mu\underline{r} \tag{9.1}$$

since $\boldsymbol{B}$ can be reached by starting from $\boldsymbol{O}$, going along the translation vector $\underline{p}$ to the point $\boldsymbol{P}$ on the line and then adding some multiple $\mu\underline{r}$ of the vector $\underline{r}$. Different values of μ give different points $\boldsymbol{B}$ on the line. We may also find the equation of the line that passes through two fixed points $\boldsymbol{P}$ and $\boldsymbol{C}$ with position vectors $\underline{p}$ and $\underline{c}$. Since $\boldsymbol{PC}$ is given by $\underline{c}-\underline{p}$, the position vector of a general point on the line is

$$\underline{b} = \underline{p} + \mu(\underline{c} - \underline{p}) \tag{9.2}$$

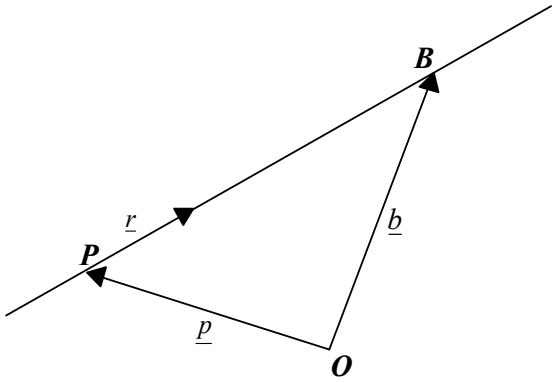

Figure 9.1: The equation of a line

Distance from a point to a line

Figure (9.2), shows a line having direction $\underline{r}$ that passes through a point $\boldsymbol{P}$ whose position vector is $\underline{p}$. To find the **minimum distance d** of the line from a point $\boldsymbol{M}$ whose position vector is $\underline{m}$, we must solve the right-angled triangle shown. We see that $d = \left|\underline{m} - \underline{p}\right| \sin\theta$, so from the definition of vector product, it

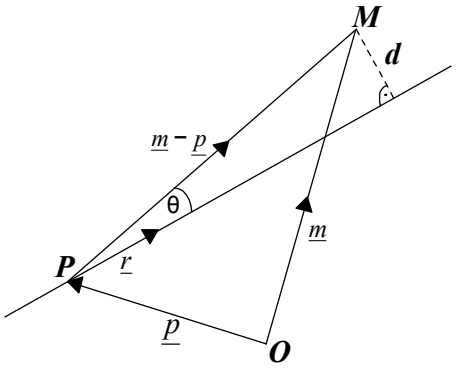

Figure 9.2: The minimum distance from a point to a line

follows that

$$d = \left|(\underline{m} - \underline{p}) \times \hat{r}\right| \tag{9.3}$$

Appendix B

Geometry of Multivariate Normal Distribution (the contour ellipsoids)

If we have $n \times 1$ random vector $\underline{x}$ that has multivariate normal distribution with mean vector μ and covariance matrix Σ, then this random vector $\underline{x}$ has probability density function given by

$$p(\underline{x}) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} |\Sigma|^{-1/2} \exp\{-\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu})\} \tag{9.4}$$

Things to note about the multivariate normal distribution:

1. The term appearing inside the exponent of the multivariate normal distribution is a quadratic form:

$$(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \tag{9.5}$$

   This particular quadratic form is called the squared **Mahalanobis distance** between the random vector $\underline{x}$ and the mean vector $\underline{\mu}$.

2. If the variables are uncorrelated then the covariance matrix is diagonal with variances of the variables appearing on the diagonal elements of the matrix

and zeros elsewhere:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

From equation (9.6) we note that the density function $p(\underline{x})$ only depends on $\underline{x}$ through the squared Mahalanobis distance:

$$(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \tag{9.6}$$

Thus the density is constant for all values of $p(\underline{x})$ such that the Mahalanobis distance equals a constant c^2

$$(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) = c^2 \tag{9.7}$$

This is the equation for a n-dimensional ellipsoid centered at $\underline{\mu}$. For a bivariate normal distribution where $n = 2$ we have an ellipse as shown below on figure (9.3). The question that we should ask now is: "What is the probability that an arbitrary or random observation will fall inside the ellipsoid?". The probability can be determined by the following preposition.

Preposition: If we have $\underline{n \times 1}$ random vector $\underline{x}$ then the squared Mahalanobis distance between and the mean vector $\underline{\mu}$ is going to be chi-square distributed with n degrees of freedom.

$$(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \approx \chi_n^2 \tag{9.8}$$

So if we define a specific n-dimensional ellipsoid by taking the squared Mahalanobis distance equal to a critical value χ_n^2 of the chi-square distribution with n degrees of freedom and evaluate this at α the so-called significance level, then the

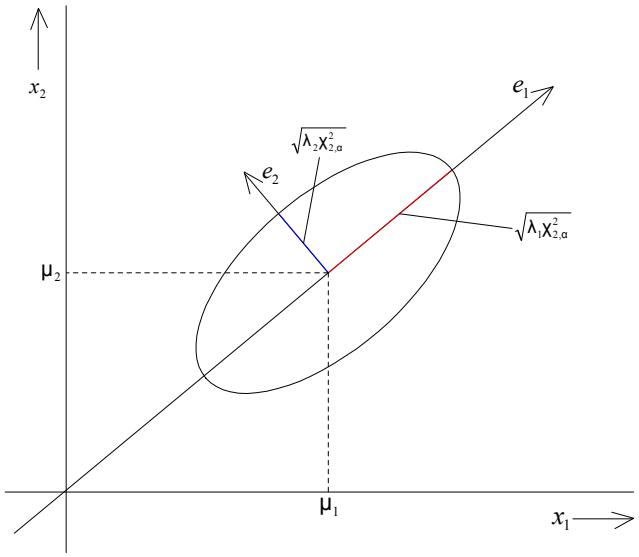

Figure 9.3: Contour ellipse for bivariate normal distribution

probability that the random value $\underline{x}$ will fall inside the ellipsoid is going to be

$$\Pr\{(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu}) \leq \chi^2_{n,\alpha}\} = 1-\alpha \tag{9.9}$$

The ellipsoid is defined by

$$(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu}) = \chi^2_{n,\alpha} \tag{9.10}$$

and this particular ellipsoid is called the $(1-\alpha)\times 100\%$ contour ellipsoid for a multivariate normal vector with mean vector $\underline{\mu}$ and covariance matrix Σ.
In order this ellipsoid to be n-dimensional, Σ must be positive-definite and symmetric with all its elements real. Under this conditions and assuming that the mean vector $\underline{\mu} = 0$ there exists an orthogonal matrix S such that Σ can be transformed to a diagonal matrix D by the relation

$$D = S^T\Sigma S \tag{9.11}$$

The diagonal elements of D are the eigenvalues of Σ and the column vectors of S are the orthonormal eigenvectors of Σ. Since Σ is nonnegative definite, the eigenvalues of are nonnegative. They are all positive when Σ is positive-definite. The number of nonzero eigenvalues is equal to the rank of Σ. Let

$$\underline{x} = S^{-1}y = S^T\underline{y}$$

and suppose that Σ is positive definite. The inverse of S exists and is equal to S^T since S is orthogonal. Thus,

$$\underline{x}^T\Sigma^{-1}\underline{x} = \underline{y}^T S\Sigma^{-1}S^T\underline{y} = \underline{y}^T D^{-1}\underline{y}$$

or

$$\underline{x}^T\Sigma^{-1}\underline{x} = \sum_{i=1}^{n}\frac{y_i^2}{\lambda_i} = \chi^2_{n,\alpha} \tag{9.12}$$

Divide this relation by $\chi^2_{n,\alpha}$,

$$\sum_{i=1}^{n}\frac{y_i^2}{\chi^2_{n,\alpha}\lambda_i} = 1 \tag{9.13}$$

Equation (9.13) is the normal form of an n-dimensional ellipsoid. The n-principal semiaxes of the ellipsoid are

$$l_i = \sqrt{\chi^2_{n,\alpha}\lambda_i} \quad i = 1, 2, \ldots, n \tag{9.14}$$

Since the columns e_i of the matrix S are orthonormal eigenvectors of Σ, it follows immediately that the e_i define the directions of the axes of the ellipsoid. When an eigenvalue is zero, the corresponding eigenvector indicates the direction normal to the subspace which contains $(n-1)$ dimensional ellipsoid.
The significance of the ellipsoids stems from the fact that they have a simple probabilistic interpretation. For the important case when $n = 3$, $\chi^2_{3,\alpha} = 7.815$ yields 3-dimensional contour ellipsoid of 95% probability. More data can be checked in the table of chi-square distribution with 3-degrees of freedom.

Appendix C

The Jacobians ∇g_k^x and ∇g_k^z are Jacobians of function $g(\cdot)$ (6.2) with respect to the state and the observation respectively are:

$$\nabla g_k^x = \begin{bmatrix} \frac{\partial gx_i}{\partial x} & \frac{\partial gx_i}{\partial y} & \frac{\partial gx_i}{\partial z} & \frac{\partial gx_i}{\partial V_x} & \frac{\partial gx_i}{\partial V_y} & \frac{\partial gx_i}{\partial V_z} & \frac{\partial gx_i}{\partial q_0} & \frac{\partial gx_i}{\partial q_1} & \frac{\partial gx_i}{\partial q_2} & \frac{\partial gx_i}{\partial q_3} \\ \frac{\partial gy_i}{\partial x} & \frac{\partial gy_i}{\partial y} & \frac{\partial gy_i}{\partial z} & \frac{\partial gy_i}{\partial V_x} & \frac{\partial gy_i}{\partial V_y} & \frac{\partial gy_i}{\partial V_z} & \frac{\partial gy_i}{\partial q_0} & \frac{\partial gy_i}{\partial q_1} & \frac{\partial gy_i}{\partial q_2} & \frac{\partial gy_i}{\partial q_3} \\ \frac{\partial gz_i}{\partial x} & \frac{\partial gz_i}{\partial y} & \frac{\partial gz_i}{\partial z} & \frac{\partial gz_i}{\partial V_x} & \frac{\partial gz_i}{\partial V_y} & \frac{\partial gz_i}{\partial V_z} & \frac{\partial gz_i}{\partial q_0} & \frac{\partial gz_i}{\partial q_1} & \frac{\partial gz_i}{\partial q_2} & \frac{\partial gz_i}{\partial q_3} \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\nabla g_k^z = \begin{bmatrix} \frac{\partial gx_i}{\partial \rho_x} & \frac{\partial gx_i}{\partial \rho_y} & \frac{\partial gx_i}{\partial \rho_z} \\ \frac{\partial gy_i}{\partial \rho_x} & \frac{\partial gy_i}{\partial \rho_y} & \frac{\partial gy_i}{\partial \rho_z} \\ \frac{\partial gz_i}{\partial \rho_x} & \frac{\partial gz_i}{\partial \rho_y} & \frac{\partial gz_i}{\partial \rho_z} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{\rho_x \hat{r}_x}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_y \hat{r}_x}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_z \hat{r}_x}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} \\ \frac{\rho_x \hat{r}_y}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_y \hat{r}_y}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_z \hat{r}_y}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} \\ \frac{\rho_x \hat{r}_z}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_y \hat{r}_z}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} & \frac{\rho_z \hat{r}_z}{\sqrt{\rho_x^2+\rho_y^2+\rho_z^2}} \end{bmatrix}$$

Appendix D

The Jacobian ∇f_k^x of the augmented process model (6.10) with respect to the vehicle state x_k^v is:

$$\nabla f_k^x = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ \phi_4 & \phi_5 & \phi_6 \\ \phi_7 & \phi_8 & \phi_9 \end{bmatrix}$$

where

$$\phi_1 = \begin{bmatrix} \frac{\partial x}{\partial x} & \frac{\partial x}{\partial y} & \frac{\partial x}{\partial z} \\ \frac{\partial y}{\partial x} & \frac{\partial y}{\partial y} & \frac{\partial y}{\partial z} \\ \frac{\partial z}{\partial x} & \frac{\partial z}{\partial y} & \frac{\partial z}{\partial z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\phi_2 = \begin{bmatrix} \frac{\partial x}{\partial V_x} & \frac{\partial x}{\partial V_y} & \frac{\partial x}{\partial V_z} \\ \frac{\partial y}{\partial V_x} & \frac{\partial y}{\partial V_y} & \frac{\partial y}{\partial V_z} \\ \frac{\partial z}{\partial V_x} & \frac{\partial z}{\partial V_y} & \frac{\partial z}{\partial V_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\phi_3 = \begin{bmatrix} \frac{\partial x}{\partial q_0} & \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_0} & \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_0} & \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \end{bmatrix} = 0^{4\times 3}$$

$$\phi_4 = \begin{bmatrix} \frac{\partial V_x}{\partial x} & \frac{\partial V_x}{\partial y} & \frac{\partial V_x}{\partial z} \\ \frac{\partial V_y}{\partial x} & \frac{\partial V_y}{\partial y} & \frac{\partial V_y}{\partial z} \\ \frac{\partial V_z}{\partial x} & \frac{\partial V_z}{\partial y} & \frac{\partial V_z}{\partial z} \end{bmatrix} = 0^{3\times 3}$$

$$\phi_5 = \begin{bmatrix} \frac{\partial V_x}{\partial V_x} & \frac{\partial V_x}{\partial V_y} & \frac{\partial V_x}{\partial V_z} \\ \frac{\partial V_y}{\partial V_x} & \frac{\partial V_y}{\partial V_y} & \frac{\partial V_y}{\partial V_z} \\ \frac{\partial V_z}{\partial V_x} & \frac{\partial V_z}{\partial V_y} & \frac{\partial V_z}{\partial V_z} \end{bmatrix} = 0^{3\times 3}$$

$$\phi_6 = \begin{bmatrix} \frac{\partial V_x}{\partial q_0} & \frac{\partial V_x}{\partial q_1} & \frac{\partial V_x}{\partial q_2} & \frac{\partial V_x}{\partial q_3} \\ \frac{\partial V_y}{\partial q_0} & \frac{\partial V_y}{\partial q_1} & \frac{\partial V_y}{\partial q_2} & \frac{\partial V_y}{\partial q_3} \\ \frac{\partial V_z}{\partial q_0} & \frac{\partial V_z}{\partial q_1} & \frac{\partial V_z}{\partial q_2} & \frac{\partial V_z}{\partial q_3} \end{bmatrix} =$$

where

$$\begin{aligned} \frac{\partial V_x}{\partial q_0} &= 2q_0a_x - 2q_3a_y + 2q_2a_z \\ \frac{\partial V_x}{\partial q_1} &= 2q_1a_x + 2q_2a_y + 2q_3a_z \\ \frac{\partial V_x}{\partial q_2} &= -2q_2a_x + 2q_1a_y + 2q_0a_z \\ \frac{\partial V_x}{\partial q_3} &= -2q_3a_x - 2q_0a_y + 2q_1a_z \end{aligned}$$

$$\begin{aligned} \frac{\partial V_y}{\partial q_0} &= 2q_3a_x + 2q_0a_y - 2q_1a_z \\ \frac{\partial V_y}{\partial q_1} &= 2q_2a_x - 2q_1a_y - 2q_0a_z \\ \frac{\partial V_y}{\partial q_2} &= 2q_1a_x + 2q_2a_y + 2q_3a_z \\ \frac{\partial V_y}{\partial q_3} &= 2q_0a_x - 2q_3a_y + 2q_2a_z \end{aligned}$$

$$\begin{aligned} \frac{\partial V_z}{\partial q_0} &= -2q_2a_x + 2q_1a_y + 2q_0a_z \\ \frac{\partial V_z}{\partial q_1} &= 2q_3a_x + 2q_0a_y - 2q_1a_z \\ \frac{\partial V_z}{\partial q_2} &= -2q_0a_x + 2q_3a_y - 2q_2a_z \\ \frac{\partial V_z}{\partial q_3} &= 2q_1a_x + 2q_2a_y + 2q_3a_z \end{aligned}$$

$$\phi_7 = \begin{bmatrix} \frac{\partial q_0}{\partial x} & \frac{\partial q_0}{\partial y} & \frac{\partial q_0}{\partial z} \\ \frac{\partial q_1}{\partial x} & \frac{\partial q_1}{\partial y} & \frac{\partial q_1}{\partial z} \\ \frac{\partial q_2}{\partial x} & \frac{\partial q_2}{\partial y} & \frac{\partial q_2}{\partial z} \\ \frac{\partial q_3}{\partial x} & \frac{\partial q_3}{\partial y} & \frac{\partial q_3}{\partial z} \end{bmatrix} = 0^{3\times 4}$$

$$\phi_8 = \begin{bmatrix} \frac{\partial q_0}{\partial V_x} & \frac{\partial q_0}{\partial V_y} & \frac{\partial q_0}{\partial V_z} \\ \frac{\partial q_1}{\partial V_x} & \frac{\partial q_1}{\partial V_y} & \frac{\partial q_1}{\partial V_z} \\ \frac{\partial q_2}{\partial V_x} & \frac{\partial q_2}{\partial V_y} & \frac{\partial q_2}{\partial V_z} \\ \frac{\partial q_3}{\partial V_x} & \frac{\partial q_3}{\partial V_y} & \frac{\partial q_3}{\partial V_z} \end{bmatrix} = 0^{3\times 4}$$

$$\phi_9 = \begin{bmatrix} \frac{\partial q_0}{\partial q_0} & \frac{\partial q_0}{\partial q_1} & \frac{\partial q_0}{\partial q_2} & \frac{\partial q_0}{\partial q_3} \\ \frac{\partial q_1}{\partial q_0} & \frac{\partial q_1}{\partial q_1} & \frac{\partial q_1}{\partial q_2} & \frac{\partial q_1}{\partial q_3} \\ \frac{\partial q_2}{\partial q_0} & \frac{\partial q_2}{\partial q_1} & \frac{\partial q_2}{\partial q_2} & \frac{\partial q_2}{\partial q_3} \\ \frac{\partial q_3}{\partial q_0} & \frac{\partial q_3}{\partial q_1} & \frac{\partial q_3}{\partial q_2} & \frac{\partial q_3}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & -p/2 & -q/2 & -r/2 \\ p/2 & 0 & r/2 & -q/2 \\ q/2 & -r/2 & 0 & p/2 \\ r/2 & q/2 & -p/2 & 0 \end{bmatrix}$$

Appendix E

The Jacobian ∇h_k^x of the nonlinear observation function $h(\cdot)$ (6.15) with respect to the augmented state vector i.e. the vehicle state and the map point estimate used in the update is:

$$\nabla h_k^x = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}$$

where

$$\varphi_1 = \begin{bmatrix} \frac{\partial h\rho_{bx}}{\partial x} & \frac{\partial h\rho_{bx}}{\partial y} & \frac{\partial h\rho_{bx}}{\partial z} \\ \frac{\partial h\rho_{by}}{\partial x} & \frac{\partial h\rho_{by}}{\partial y} & \frac{\partial h\rho_{by}}{\partial z} \\ \frac{\partial h\rho_{bz}}{\partial x} & \frac{\partial h\rho_{bz}}{\partial y} & \frac{\partial h\rho_{bz}}{\partial z} \end{bmatrix}$$

$$\begin{aligned} \frac{\partial h\rho_{bx}}{\partial x} &= -q_0^2 - q_1^2 + q_2^2 + q_3^2 \\ \frac{\partial h\rho_{bx}}{\partial y} &= -2(q_1 q_2 + q_0 q_3) \\ \frac{\partial h\rho_{bx}}{\partial z} &= -2(-q_0 q_2 + q_1 q_{3)} \end{aligned}$$

$$\begin{aligned} \frac{\partial h\rho_{by}}{\partial x} &= -2(q_1 q_2 - q_0 q_3) \\ \frac{\partial h\rho_{by}}{\partial y} &= -q_0^2 + q_1^2 - q_2^2 + q_3^2 \\ \frac{\partial h\rho_{by}}{\partial z} &= -2(q_0 q_1 + q_2 q_3) \end{aligned}$$

$$\begin{aligned} \frac{\partial h\rho_{bz}}{\partial x} &= -2(q_0 q_2 + q_1 q_3) \\ \frac{\partial h\rho_{bz}}{\partial y} &= -2(-q_0 q_1 + q_2 q_3) \\ \frac{\partial h\rho_{bz}}{\partial z} &= -q_0^2 + q_1^2 + q_2^2 - q_3^2 \end{aligned}$$

$$\varphi_2 = \begin{bmatrix} \frac{\partial h\rho_{bx}}{\partial V_x} & \frac{\partial h\rho_{bx}}{\partial V_y} & \frac{\partial h\rho_{bx}}{\partial V_z} & \frac{\partial h\rho_{bx}}{\partial q_0} & \frac{\partial h\rho_{bx}}{\partial q_1} & \frac{\partial h\rho_{bx}}{\partial q_2} & \frac{\partial h\rho_{bx}}{\partial q_3} \\ \frac{\partial h\rho_{by}}{\partial V_x} & \frac{\partial h\rho_{by}}{\partial V_y} & \frac{\partial h\rho_{by}}{\partial V_z} & \frac{\partial h\rho_{by}}{\partial q_0} & \frac{\partial h\rho_{by}}{\partial q_1} & \frac{\partial h\rho_{by}}{\partial q_2} & \frac{\partial h\rho_{by}}{\partial q_3} \\ \frac{\partial h\rho_{bz}}{\partial V_x} & \frac{\partial h\rho_{bz}}{\partial V_y} & \frac{\partial h\rho_{bz}}{\partial V_z} & \frac{\partial h\rho_{bz}}{\partial q_0} & \frac{\partial h\rho_{bz}}{\partial q_1} & \frac{\partial h\rho_{bz}}{\partial q_2} & \frac{\partial h\rho_{bz}}{\partial q_3} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \frac{\partial h\rho_{bx}}{\partial q_0} & \frac{\partial h\rho_{bx}}{\partial q_1} & \frac{\partial h\rho_{bx}}{\partial q_2} & \frac{\partial h\rho_{bx}}{\partial q_3} \\ 0 & 0 & 0 & \frac{\partial h\rho_{by}}{\partial q_0} & \frac{\partial h\rho_{by}}{\partial q_1} & \frac{\partial h\rho_{by}}{\partial q_2} & \frac{\partial h\rho_{by}}{\partial q_3} \\ 0 & 0 & 0 & \frac{\partial h\rho_{bz}}{\partial q_0} & \frac{\partial h\rho_{bz}}{\partial q_1} & \frac{\partial h\rho_{bz}}{\partial q_2} & \frac{\partial h\rho_{bz}}{\partial q_3} \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial h\rho_{bx}}{\partial q_0} &= 2q_0(-x+x_i) + 2q_3(-y+y_i) - 2q_2(-z+z_i) \\
\frac{\partial h\rho_{bx}}{\partial q_1} &= 2q_1(-x+x_i) + 2q_2(-y+y_i) + 2q_3(-z+z_i) \\
\frac{\partial h\rho_{bx}}{\partial q_2} &= -2q_2(-x+x_i) + 2q_1(-y+y_i) - 2q_0(-z+z_i) \\
\frac{\partial h\rho_{bx}}{\partial q_3} &= -2q_3(-x+x_i) + 2q_0(-y+y_i) + 2q_1(-z+z_i)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial h\rho_{by}}{\partial q_0} &= -2q_3(-x+x_i) + 2q_0(-y+y_i) + 2q_1(-z+z_i) \\
\frac{\partial h\rho_{by}}{\partial q_1} &= 2q_2(-x+x_i) - 2q_1(-y+y_i) + 2q_0(-z+z_i) \\
\frac{\partial h\rho_{by}}{\partial q_2} &= 2q_1(-x+x_i) + 2q_2(-y+y_i) + 2q_3(-z+z_i) \\
\frac{\partial h\rho_{by}}{\partial q_3} &= -2q_0(-x+x_i) - 2q_3(-y+y_i) + 2q_2(-z+z_i)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial h\rho_{bz}}{\partial q_0} &= 2q_2(-x+x_i) - 2q_1(-y+y_i) + 2q_0(-z+z_i) \\
\frac{\partial h\rho_{bz}}{\partial q_1} &= 2q_3(-x+x_i) - 2q_0(-y+y_i) - 2q_1(-z+z_i) \\
\frac{\partial h\rho_{bz}}{\partial q_2} &= 2q_0(-x+x_i) + 2q_3(-y+y_i) - 2q_2(-z+z_i) \\
\frac{\partial h\rho_{bz}}{\partial q_3} &= 2q_1(-x+x_i) + 2q_2(-y+y_i) + 2q_3(-z+z_i)
\end{aligned}$$

$$\varphi_3 = \begin{bmatrix} \frac{\partial h\rho_{bx}}{\partial x_i} & \frac{\partial h\rho_{bx}}{\partial y_i} & \frac{\partial h\rho_{bx}}{\partial z_i} \\ \frac{\partial h\rho_{by}}{\partial x_i} & \frac{\partial h\rho_{by}}{\partial y_i} & \frac{\partial h\rho_{by}}{\partial z_i} \\ \frac{\partial h\rho_{bz}}{\partial x_i} & \frac{\partial h\rho_{bz}}{\partial y_i} & \frac{\partial h\rho_{bz}}{\partial z_i} \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial h\rho_{bx}}{\partial x_i} &= q_0^2 + q_1^2 - q_2^2 - q_3^2 \\
\frac{\partial h\rho_{bx}}{\partial y_i} &= 2(q_1q_2 + q_0q_3) \\
\frac{\partial h\rho_{bx}}{\partial z_i} &= 2(-q_0q_2 + q_1q_3)
\end{aligned}$$

$$\frac{\partial h\rho_{by}}{\partial x_i} = 2(q_1q_2 - q_0q_3)$$
$$\frac{\partial h\rho_{by}}{\partial y_i} = q_0^2 - q_1^2 + q_2^2 - q_3^2$$
$$\frac{\partial h\rho_{by}}{\partial z_i} = 2(q_0q_1 + q_2q_3)$$

$$\frac{\partial h\rho_{bz}}{\partial x_i} = 2(q_0q_2 + q_1q_3)$$
$$\frac{\partial h\rho_{bz}}{\partial y_i} = 2(-q_0q_1 + q_2q_3)$$
$$\frac{\partial h\rho_{bz}}{\partial z_i} = q_0^2 - q_1^2 - q_2^2 + q_3^2$$

Appendix F

The Jacobian ∇h_k^x of the nonlinear observation function $h(\cdot)$ (6.24) with respect to the augmented state vector i.e. the vehicle state and the map point estimate used in the update is:

$$\nabla h_k^x = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}$$

where

$$\varphi_1 = \begin{bmatrix} \frac{\partial h\rho_x}{\partial x} & \frac{\partial h\rho_x}{\partial y} & \frac{\partial h\rho_x}{\partial z} \\ \frac{\partial h\rho_y}{\partial x} & \frac{\partial h\rho_y}{\partial y} & \frac{\partial h\rho_y}{\partial z} \\ \frac{\partial h\rho_z}{\partial x} & \frac{\partial h\rho_z}{\partial y} & \frac{\partial h\rho_z}{\partial z} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\varphi_2 = \begin{bmatrix} \frac{\partial h\rho_x}{\partial V_x} & \frac{\partial h\rho_x}{\partial V_y} & \frac{\partial h\rho_x}{\partial V_z} & \frac{\partial h\rho_x}{\partial q_0} & \frac{\partial h\rho_x}{\partial q_1} & \frac{\partial h\rho_x}{\partial q_2} & \frac{\partial h\rho_x}{\partial q_3} \\ \frac{\partial h\rho_y}{\partial V_x} & \frac{\partial h\rho_y}{\partial V_y} & \frac{\partial h\rho_y}{\partial V_z} & \frac{\partial h\rho_y}{\partial q_0} & \frac{\partial h\rho_y}{\partial q_1} & \frac{\partial h\rho_y}{\partial q_2} & \frac{\partial h\rho_y}{\partial q_3} \\ \frac{\partial h\rho_z}{\partial V_x} & \frac{\partial h\rho_z}{\partial V_y} & \frac{\partial h\rho_z}{\partial V_z} & \frac{\partial h\rho_z}{\partial q_0} & \frac{\partial h\rho_z}{\partial q_1} & \frac{\partial h\rho_z}{\partial q_2} & \frac{\partial h\rho_z}{\partial q_3} \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\varphi_3 = \begin{bmatrix} \frac{\partial h\rho_x}{\partial x_i} & \frac{\partial h\rho_x}{\partial y_i} & \frac{\partial h\rho_x}{\partial z_i} \\ \frac{\partial h\rho_y}{\partial x_i} & \frac{\partial h\rho_y}{\partial y_i} & \frac{\partial h\rho_y}{\partial z_i} \\ \frac{\partial h\rho_z}{\partial x_i} & \frac{\partial h\rho_z}{\partial y_i} & \frac{\partial h\rho_z}{\partial z_i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Appendix G

The Jacobian ∇g_k^z of the function $g(\cdot)$ (6.2) with respect to observation z_k is:

$$\nabla g_k^z = \begin{bmatrix} \frac{\partial g x_i}{\partial \hat{r}_x} & \frac{\partial g x_i}{\partial \hat{r}_y} & \frac{\partial g x_i}{\partial \hat{r}_z} \\ \frac{\partial g y_i}{\partial \hat{r}_x} & \frac{\partial g y_i}{\partial \hat{r}_y} & \frac{\partial g y_i}{\partial \hat{r}_z} \\ \frac{\partial g z_i}{\partial \hat{r}_x} & \frac{\partial g z_i}{\partial \hat{r}_y} & \frac{\partial g z_i}{\partial \hat{r}_z} \end{bmatrix} = \begin{bmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & \rho \end{bmatrix}$$

Appendix H

The Jacobian ∇h_k^x of the nonlinear observation function $h(\cdot)$ (6.21) with respect to the augmented state vector $x(k)$ i.e. the vehicle state and the map point estimate used in the update is:

$$\nabla h_x(k) = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}$$

where

$$\varphi_1 = \begin{bmatrix} \frac{\partial h\hat{r}_{bx}}{\partial x} & \frac{\partial h\hat{r}_{bx}}{\partial y} & \frac{\partial h\hat{r}_{bx}}{\partial z} \\ \frac{\partial h\hat{r}_{by}}{\partial x} & \frac{\partial h\hat{r}_{by}}{\partial y} & \frac{\partial h\hat{r}_{by}}{\partial z} \\ \frac{\partial h\hat{r}_{bz}}{\partial x} & \frac{\partial h\hat{r}_{bz}}{\partial y} & \frac{\partial h\hat{r}_{bz}}{\partial z} \end{bmatrix}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial x} = \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_1q_2+q_0q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_2+q_1q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{q_0^2+q_1^2-q_2^2-q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial y} = \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_1q_2+q_0q_3)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_2+q_1q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_1q_2+q_0q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial z} = \frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_1q_2+q_0q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_2+q_1q_3)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_2+q_1q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial x} = \frac{2(q_1q_2-q_0q_3)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_0q_1+q_2q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_1q_2-q_0q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial y} = \frac{2(q_1q_2-q_0q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_0q_1+q_2q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{q_0^2-q_1^2+q_2^2-q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial z} = \frac{2(q_1q_2-q_0q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_0q_1+q_2q_3)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_0q_1+q_2q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial x} = \frac{2(q_0q_2+q_1q_3)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_1+q_2q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_0q_2+q_1q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial y} = \frac{2(q_0q_2+q_1q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_1+q_2q_3)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_1+q_2q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial z} = \frac{2(q_0q_2+q_1q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_1+q_2q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{q_0^2-q_1^2-q_2^2+q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\varphi_2 = \begin{bmatrix} \frac{\partial h\hat{r}_{bx}}{\partial V_x} & \frac{\partial h\hat{r}_{bx}}{\partial V_y} & \frac{\partial h\hat{r}_{bx}}{\partial V_z} & \frac{\partial h\hat{r}_{bx}}{\partial q_0} & \frac{\partial h\hat{r}_{bx}}{\partial q_1} & \frac{\partial h\hat{r}_{bx}}{\partial q_2} & \frac{\partial h\hat{r}_{bx}}{\partial q_3} \\ \frac{\partial h\hat{r}_{by}}{\partial V_x} & \frac{\partial h\hat{r}_{by}}{\partial V_y} & \frac{\partial h\hat{r}_{by}}{\partial V_z} & \frac{\partial h\hat{r}_{by}}{\partial q_0} & \frac{\partial h\hat{r}_{by}}{\partial q_1} & \frac{\partial h\hat{r}_{by}}{\partial q_2} & \frac{\partial h\hat{r}_{by}}{\partial q_3} \\ \frac{\partial h\hat{r}_{bz}}{\partial V_x} & \frac{\partial h\hat{r}_{bz}}{\partial V_y} & \frac{\partial h\hat{r}_{bz}}{\partial V_z} & \frac{\partial h\hat{r}_{bz}}{\partial q_0} & \frac{\partial h\hat{r}_{bz}}{\partial q_1} & \frac{\partial h\hat{r}_{bz}}{\partial q_2} & \frac{\partial h\hat{r}_{bz}}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \frac{\partial h\hat{r}_{bx}}{\partial q_0} & \frac{\partial h\hat{r}_{bx}}{\partial q_1} & \frac{\partial h\hat{r}_{bx}}{\partial q_2} & \frac{\partial h\hat{r}_{bx}}{\partial q_3} \\ 0 & 0 & 0 & \frac{\partial h\hat{r}_{by}}{\partial q_0} & \frac{\partial h\hat{r}_{by}}{\partial q_1} & \frac{\partial h\hat{r}_{by}}{\partial q_2} & \frac{\partial h\hat{r}_{by}}{\partial q_3} \\ 0 & 0 & 0 & \frac{\partial h\hat{r}_{bz}}{\partial q_0} & \frac{\partial h\hat{r}_{bz}}{\partial q_1} & \frac{\partial h\hat{r}_{bz}}{\partial q_2} & \frac{\partial h\hat{r}_{bz}}{\partial q_3} \end{bmatrix}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial q_0} = \frac{2q_0(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_3(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} - \frac{2q_2(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial q_1} = \frac{2q_1(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_2(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_3(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial q_2} = -\frac{2q_2(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_1(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} - \frac{2q_0(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial q_3} = -\frac{2q_3(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_0(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_1(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial q_0} = -\frac{2q_3(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_0(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_1(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial q_1} = \frac{2q_2(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_1(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_0(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial q_2} = \frac{2q_1(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_2(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_3(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial q_3} = -\frac{2q_0(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_3(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_2(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial q_0} = \frac{2q_2(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_1(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_0(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial q_1} = \frac{2q_3(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_0(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_1(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial q_2} = \frac{2q_0(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_3(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}- \frac{2q_2(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial q_3} = \frac{2q_1(-x+x_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}} + \frac{2q_2(-y+y_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}+ \frac{2q_3(-z+z_i)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\varphi_3 = \begin{bmatrix} \frac{\partial h\hat{r}_{bx}}{\partial x_i} & \frac{\partial h\hat{r}_{bx}}{\partial y_i} & \frac{\partial h\hat{r}_{bx}}{\partial z_i} \\ \frac{\partial h\hat{r}_{by}}{\partial x_i} & \frac{\partial h\hat{r}_{by}}{\partial y_i} & \frac{\partial h\hat{r}_{by}}{\partial z_i} \\ \frac{\partial h\hat{r}_{bz}}{\partial x_i} & \frac{\partial h\hat{r}_{bz}}{\partial y_i} & \frac{\partial h\hat{r}_{bz}}{\partial z_i} \end{bmatrix}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial x_i} = -\frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_1q_2+q_0q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_2+q_1q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{q_0^2+q_1^2-q_2^2-q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial y_i} = -\frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_1q_2+q_0q_3)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_2+q_1q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_1q_2+q_0q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bx}}{\partial z_i} = -\frac{(q_0^2+q_1^2-q_2^2-q_3^2)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_1q_2+q_0q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_2+q_1q_3)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_2+q_1q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial x_i} = -\frac{2(q_1q_2-q_0q_3)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_0q_1+q_2q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_1q_2-q_0q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial y_i} = -\frac{2(q_1q_2-q_0q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_0q_1+q_2q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{q_0^2-q_1^2+q_2^2-q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{by}}{\partial z_i} = -\frac{2(q_1q_2-q_0q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2+q_2^2-q_3^2)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(q_0q_1+q_2q_3)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_0q_1+q_2q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial x_i} = -\frac{2(q_0q_2+q_1q_3)(-x+x_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_1+q_2q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(q_0q_2+q_1q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial y_i} = -\frac{2(q_0q_2+q_1q_3)(-x+x_i)(-y+y_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_1+q_2q_3)(-y+y_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{2(-q_0q_1+q_2q_3)}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

$$\frac{\partial h\hat{r}_{bz}}{\partial z_i} = -\frac{2(q_0q_2+q_1q_3)(-x+x_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{2(-q_0q_1+q_2q_3)(-y+y_i)(-z+z_i)}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} - \frac{(q_0^2-q_1^2-q_2^2+q_3^2)(-z+z_i)^2}{((-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2)^{3/2}} + \frac{q_0^2-q_1^2-q_2^2+q_3^2}{\sqrt{(-x+x_i)^2+(-y+y_i)^2+(-z+z_i)^2}}$$

Appendix J

The Jacobian ∇h_k^x of the nonlinear observation function $h(\cdot)$ (6.27) with respect to the augmented state vector $x(k)$ i.e. the vehicle state and the map point estimate used in the update is:

$$\nabla h_x(k) = \begin{bmatrix} \varphi_1 & \varphi_2 & \varphi_3 \end{bmatrix}$$

where

$$\varphi_1 = \begin{bmatrix} \frac{\partial h\hat{r}_x}{\partial x} & \frac{\partial h\hat{r}_x}{\partial y} & \frac{\partial h\hat{r}_x}{\partial z} \\ \frac{\partial h\hat{r}_y}{\partial x} & \frac{\partial h\hat{r}_y}{\partial y} & \frac{\partial h\hat{r}_y}{\partial z} \\ \frac{\partial h\hat{r}_z}{\partial x} & \frac{\partial h\hat{r}_z}{\partial y} & \frac{\partial h\hat{r}_z}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{(x_i-x)^2}{r^{3/2}} - \frac{1}{\sqrt{r}} & \frac{(x_i-x)(y_i-y)}{r^{3/2}} & \frac{(x_i-x)(z_i-z)}{r^{3/2}} \\ \frac{(x_i-x)(y_i-y)}{r^{3/2}} & \frac{(y_i-y)^2}{r^{3/2}} - \frac{1}{\sqrt{r}} & \frac{(y_i-y)(z_i-z)}{r^{3/2}} \\ \frac{(x_i-x)(z_i-z)}{r^{3/2}} & \frac{(y_i-y)(z_i-z)}{r^{3/2}} & \frac{(z_i-z)^2}{r^{3/2}} - \frac{1}{\sqrt{r}} \end{bmatrix}$$

$$\varphi_2 = \begin{bmatrix} \frac{\partial h\hat{r}_x}{\partial V_x} & \frac{\partial h\hat{r}_x}{\partial V_y} & \frac{\partial h\hat{r}_x}{\partial V_z} & \frac{\partial h\hat{r}_x}{\partial q_0} & \frac{\partial h\hat{r}_x}{\partial q_1} & \frac{\partial h\hat{r}_x}{\partial q_2} & \frac{\partial h\hat{r}_x}{\partial q_3} \\ \frac{\partial h\hat{r}_y}{\partial V_x} & \frac{\partial h\hat{r}_y}{\partial V_y} & \frac{\partial h\hat{r}_y}{\partial V_z} & \frac{\partial h\hat{r}_y}{\partial q_0} & \frac{\partial h\hat{r}_y}{\partial q_1} & \frac{\partial h\hat{r}_y}{\partial q_2} & \frac{\partial h\hat{r}_y}{\partial q_3} \\ \frac{\partial h\hat{r}_z}{\partial V_x} & \frac{\partial h\hat{r}_z}{\partial V_y} & \frac{\partial h\hat{r}_z}{\partial V_z} & \frac{\partial h\hat{r}_z}{\partial q_0} & \frac{\partial h\hat{r}_z}{\partial q_1} & \frac{\partial h\hat{r}_z}{\partial q_2} & \frac{\partial h\hat{r}_z}{\partial q_3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\varphi_3 = \begin{bmatrix} \frac{\partial h\hat{r}_x}{\partial x_i} & \frac{\partial h\hat{r}_x}{\partial y_i} & \frac{\partial h\hat{r}_x}{\partial z_i} \\ \frac{\partial h\hat{r}_y}{\partial x_i} & \frac{\partial h\hat{r}_y}{\partial y_i} & \frac{\partial h\hat{r}_y}{\partial z_i} \\ \frac{\partial h\hat{r}_z}{\partial x_i} & \frac{\partial h\hat{r}_z}{\partial y_i} & \frac{\partial h\hat{r}_z}{\partial z_i} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{1}{\sqrt{r}} - \frac{(x_i-x)^2}{r^{3/2}} & -\frac{(x_i-x)(y_i-y)}{r^{3/2}} & -\frac{(x_i-x)(z_i-z)}{r^{3/2}} \\ -\frac{(x_i-x)(y_i-y)}{r^{3/2}} & \frac{1}{\sqrt{r}} - \frac{(y_i-y)^2}{r^{3/2}} & -\frac{(y_i-y)(z_i-z)}{r^{3/2}} \\ -\frac{(x_i-x)(z_i-z)}{r^{3/2}} & -\frac{(y_i-y)(z_i-z)}{r^{3/2}} & \frac{1}{\sqrt{r}} - \frac{(z_i-z)^2}{r^{3/2}} \end{bmatrix}$$

where $r = (x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2$.

References

J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability when building fully correlated maps. *IEEE Transactions on Robotics*, 2005. 81

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Special Issue of IEEE Transactions on Signal Processing*, 2002. 47, 48, 49, 50, 51

T. Bailey. Constrained initialization for bearing-only slam. *IEEE International Conference on Robotics and Automation*, 2003. 107

T. Bailey and H. Durrant-White. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics and Automation Magazine*, 2006. 61, 65, 80

I.Y. Bar-Itzhack. Optimal updating of ins using sighting devices. *Journal of Guidance, Control and Dynamics*, 1(5), 1978. 5

Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001. 78, 79

D. J. Biezad. *Integrated Navigation and Guidance Systems*. AIAA Education Series, 1999. 9

R. Britting. *Inertial Navigation System Analysis*. John Wiley & Sons Inc., New York, 1971. 17, 22

R. Brown and Y. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, New York, 1992. 9

M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne slam. *IEEE Transactions on Aerospace and Electronic Systems*, 2008a. 80

M. Bryson and S. Sukkarieh. Building a robust implementation of bearing-only inertial slam for a uav. *Journal of Field Robotics, Special issue on SLAM in the field*, 2008b. 107

D. Brzezinska, L.R. Haala, and N.C. Toth. Multi-sensor systems for land-based and airborne mapping: technology and the future. *ISPRS Commision II Symp. Data Integration: Systems and Techniques, Cambridge, UK*, pages 320–326, 1998. 6, 10

B. Call, R. Beard, and C. Taylor. Obstacle avoidance for unmanned air vehicles using image feature tracking. *AIAA Guidance Navigation and Control Conference and Exhibit*, 2006. 6, 10

Ling Ching-Fang. *Modern Navigation Guidance and Control Processing*. Prentice-Hall Inc., New Yersey, 1991. xi, 5

T.M. Cover and Y.A. Thomas. *Elements of Information Theory*. Second edition, Wiley Interscience, 2006. 83

J.L. Crassidis. Sigma-point kalman filtering for integrated gps and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756, 2006. 9

M. Csorba. *Simultaneous Localization and Map Building*. PhD thesis, University of Oxford, 1997. 59, 65, 74, 98, 100

F. Daum. Nonlinear filters: Beyond the kalman filter. *IEEE Aerospace and Electronics Systems Magazine*, 2005. 51

A.J. Davison. Real-time simultaneous localization and mapping with a single camera. *Proceedings of the 9th International Conference on Computer Vision*, pages 1403–1410, 2003. 107

M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearings only sensor. *7th International Symposium on Experimental Robotics*, 2000. 107

M.W.M. Gamini Dissanayake, P. Newman, S. Clark, H.F. Durrant-White, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3), 2001. 59, 65, 74

A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 2000. 47, 49

A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001. 51, 126

H. Durrant-White and T. Bailey. Simultaneous localization and mapping (slam): Part i. *IEEE Robotics and Automation Magazine*, 2006. 57, 58, 60, 61, 65, 80

J.A. Farrell and M. Barth. *The Global Positioning System and Inertial Navigation*. The McGraw-Hill Inc., 1999. 2, 3, 4, 9, 16, 17, 18, 19

J.A. Farrell, T. Givargis, and M.Barth. Real time diferential carrier phase gps-aided ins. *IEEE Transactions on Control Systems Technology*, 8(4), 2000. 27

N. Franceschini. Visual guidance based on optic flow: a biorobotic approach. *Journal of Physiology*, pages 281–292, 2004. 6, 10

G. Franklin, D. Powell, and M. Workman. *Digital Control of Dynamic Systems Second edition*. Addison-Wesley, 2000. 26

A. Giremus, A. Doucet, V. Calmettes, and J.Y. Tourneret. A rao-blackwellized particle filter for ins/gps integration. *IEEE Conference on Acoustics, Speech and Signal Processing*, 3:964–967, 2004. 9

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings for Radar and Signal Processing*, 1993. 46, 53

M.S. Grewal and A.P.Andrews. *Kalman Filtering: Theory and Practice Using Matlab, Second Edition*. John Wiley and Sons Inc., New York, 2001. 9

M.S. Grewal, L.R. Weill, and A.P.Andrews. *Global Positioning System Inertial Navigation and Integration*. Wiley Interscience, New York, 2001. 9

S. Haykin. *Kalman Filtering and Neural Networks*. John Wiley and Sons Inc., 2001. 42, 43

G. Hendeby, R. Karlsson, and F. Gustafsson. A rao-blackwellized particle filter: A filter bank implementation. *EURASIP Journal of Advances in Signal Processing*, 2010. 51

R. Hermann and A.J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic control*, 1977. 79

S. Hong and H.S. Kwon L. M. Chun. Observability of error states in gps/ins integration. *IEEE Transactions on Vehicular Technology*, 54(2), 2005. 26

D. Huang and H. Leung. Em-imm based land-vehicle navigation with gps/ins. *IEEE Intelligent Transportation Systems Conference*, pages 624–629, 2004. 9

H.W.Sorenson. *Kalman Filtering theory and application*. The Institute of Electrical and Electronic Engineers, New York, 1985. 9

G. Ivey and E. Johnson. Investigation of methods for simultaneous localization and mapping using vision sensors. *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2006. 7, 11, 67

A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970. 39, 40, 41, 79, 82, 88, 111, 112

J.Sola., A.Monin., M.Devy., and T.Vidal-Calleja. Fusing monocular information in multicamera slam. *IEEE Transactions on Robotics*, 24(5), 2008. 34

S.J. Julier. A general method for approximating nonlinear transformations of probability distributions. *Robotics Research Group, Department of Engineering Science, University of Oxford*, 1996. 43

S.J. Julier. *Process Models for the Navigation of High-Speed Land Vehicles*. PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 1997. 42, 44

S.J. Julier and J.K. Uhlmann. A counter example to the theory of simultaneous localization and map building. *in Proc. IEEE Int. Conf. Robot. Automat.*, 2001. 61

S.J. Julier and J.K. Uhlmann. Using covariance intersection for slam. *Robotics and Autonomous Systems*, 2007. 60

I. Jung and S. Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. *The Ninth IEEE International Conference on Computer Vision*, 2003. 7, 11, 67

M. Kayton. One hundred years of aircraft electronics. *Journal of Guidance, Control and Dynamics*, 26(2), 2003. 4

J. Kim. *Autonomous navigation for Airborne Applications*. PhD thesis, Australian Centre for Field Robotics, The University of Sydney, 2004. 7, 11, 27, 67

A. Koch, H. Wittich, and F. Thielecke. A vision based navigation algorithm for a vtol-uav. *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2006. 6, 11

M. Koifman and I.Y. Bar-Itzhack. Inertial navigation system aided by aircraft dynamics. *IEEE Transactions on Control System Technology*, 7(4):487–493, 1999. 5

N.M. Kwok and G. Dissanayake. An efficient multiple hypothesis filter for bearing only slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004. 67

J.W. Langelaan. *State Estimation for Autonomous Flight in Cluttered Environments*. PhD thesis, Stanford University, 2006. 7, 11, 67

K.W. Lee, W. S. Wijesoma, and I.G. Javier. On the observability and observability analysis of slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. 80

C. T. Leondes. *Advances in the Techniques and Technology of the Application of Nonlinear Filters and Kalman Filters*. AGARD-AG-256, 1982. 9

R. Lerner, E. Rivlin, and P. Rotstein. Error analysis for a navigation algorithm based on optical-flow and a digital terrain map. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004. 6, 10

J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag Inc., New York, 2001. 46, 47

P. Maybeck. *Stochastic Models Estimations and Control, Volume 1,2,*. Academic Press, 1982. 79, 82, 83

R.R. Mohler and C.S. Hwang. Nonlinear data observability and information. *Journal of the Franklin Institute*, 1988. 82

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-slam: A factored solution to the simultaneous localization and mapping problem. *in Proceedings of the AAAI National Conference of Artificial Intelligence*, 2002. 61

J.M.M. Montiel, J. Civera, and A.J. Davison. Unified inverse depth parameterization for monocular slam. *obotics Science and Systems Conference*, 2006. 107

M. Niculescu. Sensor fusion algorithms for unmanned air vehicles. *Aerosonde Robotic Aircraft*, 2001. 37

J. Nygards, P. Skoglar, and M. Ulvklo. Navigation aided image processing in uav surveilance: Preliminary results and design of an airborne experimental system. *Journal of Robotic Systems*, 21(2):63–72, 2004. 6, 10

M.P. Parsley and S.J. Julier. Avoiding negative depth in inverse depth bearing-only slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008. 107

L.D.L Perera, A. Melkumyan, and E. Nettleton. On the linear and nonlinear observability analysis of the slam problem. *IEEE International conference on Mechatronics*, 2009. 80

S. Rathiman, Z. Kim, and R. Senupta. Vision based following of structures using an unmanned aerial vehicle. *Institute of Transportation Studies, University of California at Berkley, Research Report, UCB-ITS-RR-2006-1*, 2006. 6, 10

K.F. Riley, M.P Hobsonand, and S.J. Bence. *Mathematical Methods for Physics and Engineering*. Third edition, Cambridge University Press, 2006. 127

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter Particle filters for Tracking Applications*. Artech House Radar Library, 2004. 38, 40, 41, 48, 50, 51, 52, 53, 80, 83, 84, 126

S. Sarkka. *Recursive Bayesian Inference on Stochastic Differential Equations*. PhD thesis, Helsinki University of Technology, 2006. 44

P.G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance Control and Dynamics*, 21(1), 1998a. 15

P.G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithm. *Journal of Guidance Control and Dynamics*, 21(2), 1998b. 15

D.H. Shim, H. Chung, H.J. Kim, and S. Sastry. Autonomous exploration in unknown urban environments for unmanned aerial vehicles. *AIAA Guidance Navigation and Control Conference*, 2005. 1

P.M.G. Silson and V. Sazdovski. Ins velocity aiding using bearing-only measurements of unknown landmarks. *Infotech@Aerospace Coonference*, 2011. 123

B. Sinopoli, M. Micheli, G. Donato, and T.J. Koo. Vision based navigation for an unmanned aerial vehicle. *Proc. of the IEEE International Conference on Robotics an Automation*, 2001. 6, 10

R. Sivakumar and R. Sengupta. Safe uav navigation with sensor processing delays in an unknown environment. *43rd IEEE Conference on Decision and Control*, pages 1081–1086, 2004. 2

Steven S. Skiena. *The Algorithm Design Manual*. Second edition, Springer-Verlag London Limited, 2008. 106

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles, Springer Verlag*, pages 167–193, 1990. 61, 65, 70, 74

J. Sola, A. Monin, M., and T. Lemaire. Undelayed initialization in bearing-only slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005. 107

J.H. Taylor. The cramer-rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Conference on Decision and Control*, 1978. 84

P. Thicavsky, C.H. Muravchik, and A. Nehorai. Posterior cramer-rao bounds for discrete time nonlinear filtering. *IEEE Transactions on Signal Processing*, 1998. 83

S. Thrun, W. Buggard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2005. 55

D. Titterton. *Strapdown Inertial Navigation Technology*. Peter Peregrinus Ltd., 1997. 16, 20, 25

R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. *Technical Report CUED/F-INFENG/TR 380, Engineering Department, Cambridge University*, 2000. 52, 53

P. Vernaza and D.D. Lee. Rao-blackwellized particle filter for 6-dof estimation of attitude and position via gps and inertial sensors. *IEEE International Conference on Robotics and Automation*, pages 1571–1578, 2006. 9

T. Vidal-Calleja, M. Bryson, and S. Sukkarieh. On the observability of bearing-only slam. *IEEE International Conference on Robotics and Automation*, 2007. 80, 107

Michael Wade and Mohinder S. Grewal. Analysis of a cascaded ins calibration filter. *IEEE Coonference*, 1988. 31

K. Walchko. Low Cost for Inertial Navigation: Learning to Integrate Noise and Find Your Way, 2002. 26

Y. Watanabe, E. Jonhson, and A. Calise. Optimal 3-d guidance from a 2-d vision sensor. *AIAA Giudance, Navigation and Control Conference and Exhibit*, 2004. 10

Y. Watanabe, E. Johnson, and A. Calise. Vision-based approach to obstacle avoidance. *AIAA Guidance Navigation and Control Conference and Exhibit*, 2005. 6, 10

A.S. Watkins. *Vision Based Map Building and Trajectory Planning to Enable Autonomous Flight Through Urban Environments*. PhD thesis, University of Florida, 2007. 7, 11, 67

T. Webb and R. Prazenica. Vision-based state estimation for autonomous micro air vehicle. *Journal of Guidance, Control and Dynamics*, 30(3), 2007. 6, 11

J.G. Webster. The measurement, instrumentation and sensors handbook. *CRC Press LLC*, 1999. 87

P. Zhang, J. Gu, E.E. Milios, and P. Huynh. Navigation with imu/gps/digital compass with unscented kalman filter. *IEEE International Conference on Mechatronics and Automation*, pages 1497–1502, 2005. 9

C. Zhenhe, S. Jagath, and R. Ranga. Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics*, 21, 2007. 10

Z.Wang and G. Dissanayake. Observability analysis of slam using fisher information matrix. *10th International Conference on Control Automation Robotics and Vision*, 2008. 81