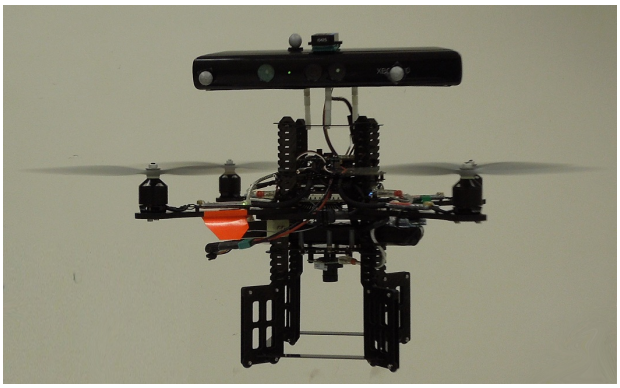
TÉCNICO LISBOA



Analysis, Design, and Implementation of an Integrated Simultaneous Localization and Mapping Algorithm

Pedro António Duarte Marques Lourenço

Dissertação para a obtenção de Grau de Mestre em

Engenharia Aeroespacial

Júri

Presidente:	Prof. Doutor João Manuel Lage de Miranda Lemos
Orientador:	Prof. Doutor Paulo Jorge Coelho Ramalho Oliveira
Vogais:	Prof. Doutor Pedro Tiago Martins Batista
	Prof. Doutor Rodrigo Martins de Matos Ventura
	Prof. Doutora Alexandra Bento Moutinho

Novembro de 2012

Acknowledgments

A Masters thesis is the first moment in the life of a student where one is permitted to focus on a single subject during a period of time longer than that of the usual project for a university course. For that reason, it must pose a truly challenging and interesting problem. I would like to thank my advisor Professor Paulo Oliveira for introducing this challenge to my academic life, and, along with my co-advisors Professor Carlos Silvestre and Pedro Batista, for their invaluable guidance, help, and motivation throughout this work. I would also like to thank Bruno Guerreiro for the patience and time answering my questions, as well as introducing me to the details of SLAM. My thanks to Professor Carlos Silvestre for the opportunity to shortly visit the SCORE Lab at the Faculty of Science and Technology of the University of Macau, which provided me high-end equipment to perform the experimental part of this work, as well as helped me broaden my mind by contacting with a whole different culture.

To all the friends I have made in the 5 years of Técnico that now culminate in this work, to which they contributed knowingly or not, my many thanks – you know who you are! I would like to thank Valter and Miranda for the everyday lunches that helped relieve the tension and put my mind on course during this semester. Finally, to the guys in the 8^{th} floor, specially those in the *Sala de Bolseiros*, for the amazing work environment they created.

Having spent the last few months focused in this work, I could not have reached this point without the many laughs with my long-time friends in Setúbal. To Ganço, Gonçalo, Miguel, Nuno and Paulo, who helped me free my head on a almost daily basis: thank you!

Last, but not the least, to my family to which I am devoted, and most importantly, to my parents who supported me throughout my life, endured me in the ups and downs, and without whom this work would not exist. Thank you sincerely for being who you are.

This work was funded by the Fundação para a Ciência e Tecnologia and the IST-ID through the project LARSyS-FCT[PEst-OE/EEI/LA009/2011].

Resumo

ESTA dissertação aborda a concepção, análise, implementação, avaliação de desempenho e validação experimental de um novo algoritmo para localização e mapeamento simultâneos, com aplicação em veículos aéreos não-tripulados. A formulação inicial é expressa no espaço dos sensores e o problema é modificado de forma a que a estrutura do sistema possa ser vista como sendo linear variante no tempo para efeitos de análise de observabilidade, o que resulta naturalmente num filtro de Kalman cuja dinâmica de erro é globalmente assimptoticamente estável. A solução proposta inclui a estimação da velocidade linear e do desvio das medidas angulares, ambos expressos no referencial solidário ao veículo.

A segunda contribuição desta dissertação consiste na formulação, desenvolvimento e validação de um algoritmo para estimação do mapa inercial e da trajectória do veículo baseando-se no mapa local fornecido pelo filtro. Para tal, é formulado um problema de optimização com solução em forma fechada, e a incerteza associada à estimação é determinada recorrendo à teoria das perturbações. No decorrer deste texto é ainda detalhada a implementação do algoritmo como um todo, sob a forma dos aspectos mais pertinentes da mesma, e são incluidos neste trabalho resultados de simulação como de uma experiência preliminar, utilizando um quadrotor instrumentado com uma câmara RGB-D, para ilustrar o desempenho do algoritmo em condições reais.

Palavras-chave: Localização e mapeamento simultâneos, Filtragem, Optimização, Fusão de sensores, Veículos aéreos não tripulados

Abstract

THIS thesis presents the design, analysis, implementation, performance evaluation and preliminary experimental validation of a novel integrated simultaneous localization and mapping algorithm (SLAM) with application to unmanned aerial vehicles (UAV). The SLAM problem is first formulated in a sensor-based framework and modified in such a way that the system structure may be regarded as time-varying for observability analysis purposes, from which a Kalman filter with globally asymptotically stable error dynamics follows naturally. The proposed solution includes the estimation of both body-fixed linear velocity and rate-gyro measurement biases. Furthermore, the formulation, solution, and validation of the problem of estimating the inertial map and trajectory with uncertainty using the sensor-based map provided by the SLAM filter are also addressed in this work. An optimization problem with a closed-form solution is formulated, and its uncertainty description is derived resorting to perturbation theory. Aspects of the implementation of the complete algorithm are detailed and both simulation results and preliminary experimental results, using an instrumented quadrotor equipped with a RGB-D camera, are included in this work to illustrate the performance of the algorithm under realistic conditions.

Keywords: Simultaneous localization and mapping, Filtering, Optimization, Sensor fusion, Unmanned aerial vehicles

Contents

List of Tables

List of Figures

List of Theorems, Lemmas, Propositions, and Definitions

Theorems

Lemmas

Propositions

Definitions

List of Symbols

Covariances

$\mathbf{\Sigma}_{\Omega}$ Rotation covariance.

$\mathbf{\Sigma}_{p_{ij}}$ Sensor-based landmark covariance.

$\mathbf{\Sigma}_{^{I}p_{ij}}$ Inertial landmark covariance.

$\mathbf{\Sigma}_{^{I}p}$ Translation covariance.

Error models

$\mathbf{a}^{(0)}$ Deterministic quantity.

$\mathbf{a}^{(1)}$ Stochastic quantity.

Landmarks

$\mathbf{d}_i$ Direction landmark in the body-fixed frame.

$\mathbf{p}_i$ Position landmark in the body-fixed frame.

$^{I}\mathbf{d}_i$ Direction landmark in the inertial frame.

$^{I}\mathbf{p}_i$ Position landmark in the inertial frame.

Vehicle variables

$\mathbf{R}$ Rotation matrix from the body-fixed to the inertial frame.

$\boldsymbol{\omega}$ Angular velocity of the vehicle.

$\mathbf{b}_{\omega}$ Angular measurement bias.

$^{I}\mathbf{p}$ Vehicle position in the inertial frame.

Notation

Symbol conventions All symbols with any dimension greater than 1, i.e., belonging to $\mathbb{R}^{m\times n}$ with either $m > 1$, $n > 1$ or both $m, n > 1$ are represented in non-italic bold. Vectors are represented in small letters, e.g. $\mathbf{a} \in \mathbb{R}^m$, and matrices are represented in capital letters, e.g. $\mathbf{A} \in \mathbb{R}^{m\times n}$. Scalar symbols are expressed in italic: constants by capital letters, e.g. A, and scalar variables in small letters, e.g. a.

Reference Frames The superscript ^{I} indicates a vector expressed in the inertial frame. For the sake of clarity, when no superscript is present, the vector is expressed in the body-fixed frame. Note that the denominations body-fixed frame and sensor-based frame are synonyms and will therefore be used in this text.

Matrices and Matrix Operations $\mathbf{I}_n$ is the identity matrix of dimension n, and $\mathbf{0}_{n\times m}$ is a n by m matrix filled with zeros. If m is omitted, the matrix is square. The superscript ^{T} applied to a matrix expresses the transpose of the said matrix. The matrix $\mathrm{skew}(\mathbf{A}) = \frac{1}{2}\left(\mathbf{A} - \mathbf{A}^T\right)$ represents the skew-symmetric part of matrix $\mathbf{A}$, and, as such, satisfies the relation $\mathrm{skew}(\mathbf{A}) = -\mathrm{skew}(\mathbf{A})^T$. $\mathbf{S}[\mathbf{a}]$ is a special skew-symmetric matrix, henceforth called the cross-product matrix as $\mathbf{S}[\mathbf{a}]\,\mathbf{b} = \mathbf{a} \times \mathbf{b}$ with $\mathbf{a} \in \mathbb{R}^3$. The determinant of any skew-symmetric matrix is zero in the tridimensional case. The matrix norm of a generic matrix $\mathbf{A}$ is defined as the Frobenius norm $\|\mathbf{A}\| := \sqrt{\mathrm{tr}\left(\mathbf{A}\mathbf{A}^\mathbf{T}\right)}$, and its determinant as $|\mathbf{A}|$, which can be used interchangeably with $\det(\mathbf{A})$.

Statistical operators The expected value of any quantity (scalar, vector or matrix) is denoted by the symbol $\langle . \rangle$, whereas the covariance symbol depends of the dimension of the quantity. The covariance of two scalar values x and y is denoted by $\sigma_{xy}^2 := \langle (x - \langle x \rangle)(y - \langle y \rangle) \rangle$. If the quantities $\mathbf{x}$ and $\mathbf{y}$ are vectors, the covariance takes the form of the covariance matrix $\mathbf{\Sigma}_{xy} := \langle (\mathbf{x} - \langle \mathbf{x} \rangle)(\mathbf{y} - \langle \mathbf{y} \rangle)^T \rangle$. In the case when both quantities are the same, the covariance becomes the variance and the subscript becomes $(.)_x$ or $(.)_y$ for both scalar and matrix variances, instead of $(.)_{xy}$.

1

Introduction

SINCE the term robot was first coined by Karel Čapek in his play *Rossum's Universal Robots* [5], published in 1920, mankind, in particular the scientific community, has strived to achieve better and more autonomous robots. In fact, as far back in time as the Ancient Greece, the mathematician and engineer Heron of Alexandria described in his work *Pneumatica et Automata* [6] over a hundred machines and automata, including machines powered by air, steam or water pressure, as well as mechanical or pneumatic means of performing wonders in Greek temples. The thrill of creation, which has always permeated Man's actions, led to the objective of recreating Nature's achievements: an example of an early attempt to do so is Leonardo da Vinci's humanoid robot [7] designed in the XV century. However well intended accomplishments such as these were, it was the pressure of war that helped the idea of autonomous vehicles flourish. In the XIX century, Giovanni Lupis envisioned a floating unmanned device, remotely controlled through a wired link, for attacking and destroying ships - the first self-propelled torpedo. That century saw the surging of several such weapons, and Nikola Tesla presented his own, with the novelty of being wirelessly controlled, by a radio link.

The advent of electronics and computers in the XX century paved the way to the thriving field of autonomous robotics that has been the subject of intensive research in the last decades, as autonomous vehicles pose the very interesting possibility of being able to perform tasks without the continuous guidance and supervision by humans. Autonomous robot missions raise the need for dependable navigation and positioning algorithms, which spawned an entire field of research. In outdoor mis-

sions, where the vehicle has access to global navigation satellite systems, this problem, albeit still a subject of research, may be considered as solved, with solutions available commercially, by means of absolute positioning systems such as the Global Positioning System. However, in what concerns indoor missions, or missions in environments where such systems cannot be used, either because of their absence or unreliability, relative positioning solutions have been proposed, in the form of aided navigation techniques, typically using range sensors, such as laser range finders or sonars, acoustic or visual marks, or, more recently, vision sensors. One of the solutions to this problem is probabilistic Simultaneous Localization and Mapping (SLAM), to which the research community has devoted tremendous effort. The SLAM problem is of paramount importance in the quest for achieving genuinely autonomous mobile robots, as it encompasses the process of placing a robot at an unknown location in an unexplored environment, and letting the robot gradually build a consistent map of its surroundings, thus localizing itself in its internal map.

The work in this thesis aims at solving this problem, by proposing an online SLAM algorithm using tridimensional (3-D) vision sensors, that integrates a Sensor-based SLAM filter with globally asymptotically stable error dynamics, and an Inertial Map and Trajectory Estimation optimal algorithm.

1.1 Problem statement

Simultaneous localization and mapping is the problem of navigating a vehicle in an unknown environment, by building a map of the area and using this map to deduce its location, without the need for a priori knowledge of location (see [8]). The objectives are twofold: i) obtaining a detailed environment model; and ii) maintaining an accurate sense of the robot's location. Despite significant advances, this problem still possesses appealing aspects for a researcher in the field, e.g., the possibility of mapping unstructured and dynamic detail-rich environments or developing computationally efficient online filters, better association of measured and state landmarks, and the absence of global convergence results for the most popular strategies.

The problem treated in this thesis is that of developing an online SLAM algorithm for an unmanned aerial vehicle (UAV), such as a quadrotor, by designing

- a SLAM filter in the space of the sensors, providing a sensor-based map and the velocity of the vehicle. The maps are represented by tridimensional landmarks, which may include up to three directions for each position landmark;

- an Inertial Map and Trajectory Estimation algorithm resulting from an optimization process by using the sensor-based estimate of the SLAM filter.

1.2 State of the Art

The SLAM problem first appeared in the scientific community in the 1980s and, since then, a variety of solutions have appeared. The three main paradigms in solving SLAM are: the EKF[1]-based solutions, that owe a lot to the seminal work in [10] and of which [11] and [12] are good examples; the Graph-based solutions that use sparse nonlinear optimization methods, see [13]; and the Particle filter approach,a non-parametrical statistical filtering technique, first introduced in SLAM in [14], and of which FastSLAM [15] is the most prominent result.

The main decisions when designing a SLAM algorithm include (see [16] for further details)

- choosing between full SLAM or Online SLAM - the full approach is usually offline and aims at determining the vehicle trajectory and a detail-rich map of the environment while the online algorithms are filters that process data incrementally;
- deciding whether the map framework will be volumetric or feature-based, topological or metric;
- identify the correspondence between what is observed and the map, either unknown or not;
- work on dynamic or static environments; and
- handle large or small uncertainty.;

The EKF solutions, as any state-space formulation, incorporate the maps in the filter state, which can quickly become very large, thus limiting the dimension of the maps it can handle. A partial solution to this problem is presented in [11], by building local submaps. The Graph-based algorithms model the problem as a sparse graph of soft constraints, each corresponding to a motion or measurement. These methods are usually employed offline as full SLAM solutions for constructing large-scale maps. Finally, the Particle filter methods work by using samples of the robot pose distribution to represent the estimation and numerous independent Gaussians for the landmarks. A very interesting review of the SLAM problem was performed in [16], and a two-part tutorial that oversees the present state of SLAM can be found in [8] and [17].

The algorithm proposed in this thesis is an online state-space filter, that builds metric feature-based maps of static unknown environments, handling uncertainty large enough for allowing effective loop closing. There are two approaches that bear resemblance to the algorithm developed, apart from [18] and [19], on which this work was based. The first is the robocentric map joining, [11], that proposes a filter framework in the body-fixed frame. The RGB-D SLAM system of [20] is also similar in the sense that it uses a RGB-D camera to inspect the environment, and a feature extracting algorithm as the SURF/SIFT.

[1] The Extended Kalman Filter is a derivation of the Kalman filter, see [9], that linearizes the models of the system to be able to propagate an approximation of the uncertainty linearly.

1.3 Contributions

As mentioned before, the work presented in this thesis is twofold: i) a Sensor-based SLAM filter; and ii) an Inertial Map and Trajectory Estimation algorithm. The main contributions of the first part or this thesis are the design, analysis, and experimental validation of a novel sensor-based SLAM filter for tridimensional environments. This work extends that presented in [18], where a bidimensional (2-D) sensor-based SLAM filter was proposed. The assumption that the environment is structured vertically, needed in the 2-D case, is dropped in this work, where only the static property of the landmarks is assumed. The second part of this thesis proposes a methodology for obtaining the inertial map and the pose of the vehicle, building on [19], using the formulation of the extended Procrustes problem there presented, and providing the uncertainty characterization of the obtained transformation. The proposed integrated algorithm

- has globally asymptotically stable (GAS) error dynamics;
- resorts to the linear and angular motion kinematics, that are exact;
- uses the low-cost *Microsoft Kinect*^{TM}, in opposition to the 2-D landmark approach, which requires the use of considerably more expensive laser range finders;
- builds on the well-established linear time-varying Kalman filtering theory;
- explicitly estimates the rate-gyro bias, fusing high bandwidth dynamic measurements with information on static landmarks;
- finds the vehicle pose in the inertial frame by solving an optimization problem;
- has fully characterized uncertainty, both provided by a Kalman filter and by the proposed novel formulation.

In comparison with other works in this field, the main contribution of this dissertation is the formal proof of global convergence for a SLAM algorithm, something that is, to the best of the author's knowledge, absent from the literature. The work in which the algorithm here proposed is based also has global convergence results, although for a system with different dynamics and aiming at the bidimensional case, where limiting assumptions need to be taken. Thus, the work performed by the author improves that of [18] and [19] by building on the idea there exposed, dropping the assumptions regarding the structure of the environment, deriving a new dynamical system and different proofs and conditions for the observability and convergence of this new system, as well as determining the uncertainty characterization of the optimization problem, that is in itself also different as it does not compute incremental transformations but the full transformation between frames, which was left undone in that work.

Finally, the algorithm was validated in all its aspects through simulation, as well as real-data results from preliminary experiments using an instrumented quadrotor.

1.4 Thesis Outline

This thesis is organized in seven chapters, including this introductory chapter. Chapter 2 is devoted to the development of a Sensor-based SLAM filter, and the establishment of theoretical results regarding its observability and convergence. Next, Chapter 3 proposes an algorithm to obtain the inertial map and the trajectory of the vehicle using the sensor-based estimates of the filter proposed in Chapter 2. This is done by defining an optimization problem while detailing a novel characterization of the uncertainty of the optimization problem. Chapter 4 details the algorithm proposed as a whole, extending and complementing the work of the previous chapters. The two following chapters report the validation of the algorithms, by presenting the results obtained with the SLAM algorithm both in simulation (Chapter 5), and experimentally (Chapter 6). Lastly, the conclusions of this dissertation are exposed in Chapter 7. This work includes two appendices that address respectively the analysis of the complexity of the algorithm (Appendix A) and the conversion of computational loops into matrix operations (Appendix B).

2

Sensor-based Simultaneous Localization and Mapping in 3-D

THE problem of designing a navigation system for a vehicle operating in an unknown environment is addressed in this chapter. This problem is solved resorting to a novel SLAM algorithm, where no linearization or approximation is used whatsoever. The only available sensors are a triaxial rate-gyro, and a RGB-D[1] camera, such as the *Microsoft Kinect™*, which provide, respectively, angular rate measurements and RGB-D images, from where 3-D landmarks may be extracted. It is worth noticing that the filtering framework is sensor-based, i.e., it takes place in the body-fixed frame. As the sensors output measurements in this frame, the sensor-based approach has the advantage of not requiring the transposition of the measurements into the filtering framework, thus relieving the system of the representation of the pose of the vehicle, and avoiding the high nonlinearities associated.

This chapter is organized as follows. Section 2.1 presents a short description of the problem, with the definition of the system dynamics. The observability analysis is performed in Section 2.2. Finally, the filter design is addressed in Section 2.3, which includes not only the standard Kalman filter prediction and update steps, but also landmark detection, data association and loop closure procedures.

[1] A RGB-D camera is a device that combines the information of two cameras: a colour camera (hence the RGB denomination – Red Green Blue) and a depth (usually infrared) camera.

2.1 Description of the problem

2.1.1 Basic notions

Consider Fig. 2.1, where the frame $\{I\}$ is defined by the orthonormal basis $\{{}^I\mathbf{i}_I, {}^I\mathbf{j}_I, {}^I\mathbf{k}_I\} \in \mathbb{R}^3$, and is fixed with the Earth, which, for the purpose of this work is considered stationary, making $\{I\}$ an inertial frame. The frame $\{B\}$ is solidary with the vehicle, hence called the body-fixed frame or sensor-based frame. It is defined by the orthonormal basis $\{{}^I\mathbf{i}_B, {}^I\mathbf{j}_B, {}^I\mathbf{k}_B\} \in \mathbb{R}^3$ expressed in inertial frame coordinates.

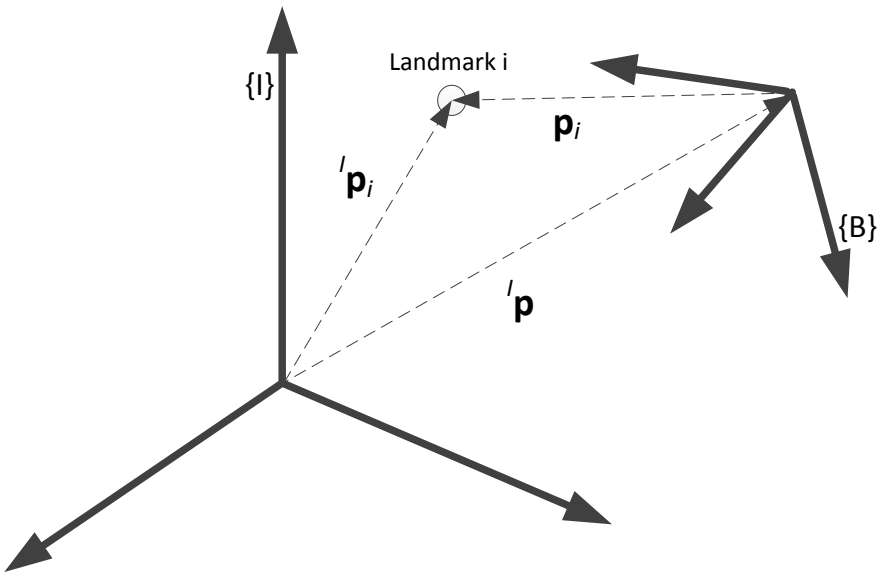


Figure 2.1: Schematic representation of both the inertial and body-fixed frames.

The transformation between both frames is fully defined by a translation and a rotation. The translation takes the form of the vector ${}^I\mathbf{p} \in \mathbb{R}^3$ in Fig. 2.1, which represents the vehicle position as well as the coordinates of the origin of the body-fixed frame expressed in the inertial frame $\{I\}$. The rotation is expressed by the rotation matrix ${}^I_B\mathbf{R}(t) \in \mathrm{SO}(3)$. A matrix $\mathbf{R}$ belonging to the special orthogonal group of rotation matrices represents a linear transformation that preserves the length of the vectors and orientation of the transformed space, in the sense that angles between transformed vectors are the same as those between the original vectors. The rotation matrix respects the following conditions:

- The rotation matrix maps the body-fixed frame basis to the inertial frame, i.e.

$$\begin{bmatrix} {}^I\mathbf{i}_B & {}^I\mathbf{j}_B & {}^I\mathbf{k}_B \end{bmatrix} = {}^I_B\mathbf{R} \begin{bmatrix} \mathbf{i}_B & \mathbf{j}_B & \mathbf{k}_B \end{bmatrix}. \tag{2.1}$$

- The determinant of a rotation matrix is unitary, $\left|{}^I_B\mathbf{R}\right| = 1$,
- The inverse of a rotation matrix is its transpose, ${}^I_B\mathbf{R}^T {}^I_B\mathbf{R} = \mathbf{I}$, and inverts the mapping, i.e., ${}^I_B\mathbf{R}^{-1} = {}^B_I\mathbf{R}$,

- The derivative of a rotation matrix is given by

$$ {}^{I}_{B}\dot{\mathbf{R}}(t) = {}^{I}_{B}\mathbf{R}(t)\mathbf{S}\left[\boldsymbol{\omega}(t)\right], \tag{2.2} $$

where $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ is the angular velocity of the body-fixed frame expressed in its coordinates, and the matrix $\mathbf{S}\left[\mathbf{a}\right]$ is the cross-product matrix such that $\mathbf{S}\left[\mathbf{a}\right]\mathbf{b} = \mathbf{a} \times \mathbf{b}$ with $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$,

$$ \mathbf{S}\left[\mathbf{a}\right] = \begin{bmatrix} 0 & -a_k & a_j \\ a_k & 0 & -a_i \\ -a_j & a_i & 0 \end{bmatrix}, $$

where $a_i = \mathbf{a} \cdot \mathbf{i}$, $a_j = \mathbf{a} \cdot \mathbf{j}$, and $a_k = \mathbf{a} \cdot \mathbf{k}$ are the components of $\mathbf{a}$.

For the sake of simplicity, and since $\{I\}$ and $\{B\}$ are the only frames used, $\mathbf{R}(t)$ will henceforth denote the rotation matrix from the body-fixed frame to the inertial frame, i.e., $\mathbf{R}(t) := {}^{I}_{B}\mathbf{R}(t)$.

2.1.2 Nonlinear System Dynamics

From Fig. 2.1 it is plain to see that the inertial coordinates of a landmark, denoted as ${}^{I}\mathbf{p}_i(t) \in \mathbb{R}^3$, are given by

$$ {}^{I}\mathbf{p}_i(t) = {}^{I}\mathbf{p}(t) + \mathbf{R}(t)\ \mathbf{p}_i(t), \tag{2.3} $$

where the vector ${}^{I}\mathbf{p}(t) \in \mathbb{R}^3$ represents the vehicle position (as well as the origin of the body-fixed frame) in the inertial frame $\{I\}$ at time t and $\mathbf{p}_i(t) \in \mathbb{R}^3$ denotes the position of landmark i with respect to the body-fixed frame expressed in the same frame $\{B\}$ in the same time instant. Taking its derivative with respect to time yields

$$ \frac{d}{dt}{}^{I}\mathbf{p}_i(t) = \frac{d}{dt}{}^{I}\mathbf{p}(t) + \frac{d}{dt}\mathbf{R}(t)\ \mathbf{p}_i(t) + \mathbf{R}(t)\ \frac{d}{dt}\mathbf{p}_i(t). \tag{2.4} $$

Given that landmarks are assumed to be static in the inertial frame, and that the derivative of the position of the origin of the body-fixed frame is the velocity of the vehicle expressed in the inertial frame, ${}^{I}\mathbf{v}(t) \in \mathbb{R}^3$, it is possible to write from (2.4),

$$ 0 = {}^{I}\mathbf{v}(t) + \mathbf{R}(t)\mathbf{S}\left[\boldsymbol{\omega}(t)\right]\mathbf{p}_i(t) + \mathbf{R}(t)\dot{\mathbf{p}}_i(t). \tag{2.5} $$

Summing up, the position and velocity of a landmark expressed in the body-fixed frame, $\mathbf{p}_i(t)$ and $\dot{\mathbf{p}}_i(t) \in \mathbb{R}^3$, satisfy

$$ \mathbf{p}_i(t) = \mathbf{R}^T(t)\ \left({}^{I}\mathbf{p}_i(t) - {}^{I}\mathbf{p}(t)\right) \tag{2.6} $$

and

$$\dot{\mathbf{p}}_i(t) = -\,\mathbf{R}^T(t)\ {}^I\mathbf{v}(t) - \mathbf{S}\left[\boldsymbol{\omega}(t)\right]\mathbf{p}_i(t), \tag{2.7}$$

respectively, where $\mathbf{R}^T(t)\ {}^I\mathbf{v}(t)$ will be substituted henceforth by $\mathbf{v}(t) \in \mathbb{R}^3$, which represents the velocity of the vehicle expressed in the body-fixed frame.

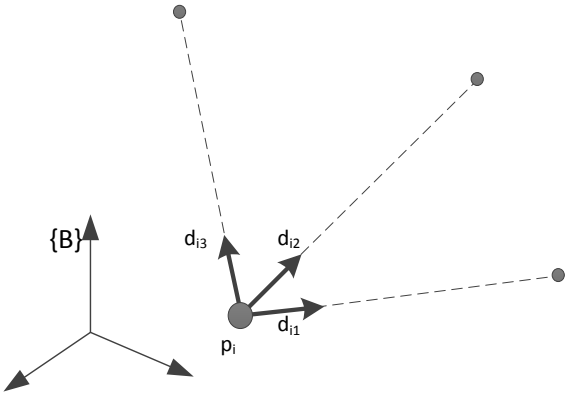


Figure 2.2: Example of a landmark composed by one position landmark and three directions.

A landmark direction is defined as the unit vector that describes the line passing through the associated position landmark and another point, as it is shown schematically in Fig. 2.2. Thus, it is possible to write,

$$\mathbf{d}_{i_j}(t) = \frac{\mathbf{p}_j(t) - \mathbf{p}_i(t)}{\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\|} = \frac{\mathbf{R}^T(t)\left({}^I\mathbf{p}_j(t) - {}^I\mathbf{p}_i(t)\right)}{\|\mathbf{R}^T(t)\left({}^I\mathbf{p}_j(t) - {}^I\mathbf{p}_i(t)\right)\|}, \tag{2.8}$$

where $\mathbf{d}_{i_j}(t) \in \mathbb{R}^3$ represents the direction j associated with landmark i at time t expressed in the body-fixed frame, thus uniting position i to j. The direction uniting the same two points, this time on $\{I\}$ is defined as

$${}^I\mathbf{d}_{i_j}(t) = \frac{{}^I\mathbf{p}_j(t) - {}^I\mathbf{p}_i(t)}{\|{}^I\mathbf{p}_j(t) - {}^I\mathbf{p}_i(t)\|}. \tag{2.9}$$

Noting that $\mathbf{R}^T(t)$, as any rotation matrix, preserves norm, it is straightforward to see that there is a correspondence between (2.8) with (2.9). This enables to write $\mathbf{d}_{i_j}(t)$ and its derivative, $\dot{\mathbf{d}}_{i_j}(t) \in \mathbb{R}^3$, given by

$$\mathbf{d}_{i_j}(t) = \mathbf{R}^T(t)\ {}^I\mathbf{d}_{i_j}(t) \tag{2.10}$$

and

$$\dot{\mathbf{d}}_{i_j}(t) = -\,\mathbf{S}\left[\boldsymbol{\omega}(t)\right]\mathbf{d}_{i_j}(t), \tag{2.11}$$

respectively. Note that, once again, landmarks are assumed static in its both forms: position and direction.

Consider that the vehicle is equipped with a triad of orthogonally mounted rate-gyros, that is

mounted at the center of mass of the vehicle and aligned with the body-fixed frame. The angular velocity of the vehicle expressed on the sensor-based frame is available through the noisy and biased rate-gyros measurements $\boldsymbol{\omega}_m(t) \in \mathbb{R}^3$,

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \mathbf{b}_\omega(t) + \mathbf{n}_\omega(t), \tag{2.12}$$

where the bias $\mathbf{b}_\omega(t) \in \mathbb{R}^3$ is assumed constant and $\mathbf{n}_\omega(t) \in \mathbb{R}^3$ corresponds to the rate-gyro noise, which is assumed to be zero-mean Gaussian white noise with standard deviation σ_{n_ω} in each component, i.e., $\mathbf{n}_\omega \sim N(0, \sigma_{n_\omega}^2 \mathbf{I}_3)$. In a deterministic setting, such as the one employed for observability analysis purposes, the noise is omitted. Taking this into account, and using the cross product property $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$, it is possible to rewrite (2.7) and (2.11) as

$$\dot{\mathbf{p}}_i(t) = -\mathbf{v}(t) - \mathbf{S}\left[\mathbf{p}_i(t)\right] \ \mathbf{b}_\omega - \mathbf{S}\left[\boldsymbol{\omega}_m(t)\right] \ \mathbf{p}_i(t) \tag{2.13}$$

and

$$\dot{\mathbf{d}}_{i_j}(t) = -\mathbf{S}\left[\mathbf{d}_{i_j}(t)\right] \ \mathbf{b}_\omega - \mathbf{S}\left[\boldsymbol{\omega}_m(t)\right] \mathbf{d}_{i_j}(t), \tag{2.14}$$

respectively. This simple step enables the inclusion of the measurement bias in the system state.

In a state-space SLAM formulation, as any algorithm using the Kalman filter or one of its variants, such as EKF-SLAM or the algorithm proposed in this work, the system state vector may be divided in two main parts, namely the vehicle state and the landmark state. With the dynamics of the landmarks now defined, the dynamics of the vehicle-related variables, i.e., the linear velocity and angular measurement bias, is yet to be defined. These will constitute the vehicle state, $\mathbf{x}_V(t) \in \mathbb{R}^{n_V}$:

$$\mathbf{x}_V(t) := \begin{bmatrix} \mathbf{v}^T(t) & \mathbf{b}_\omega^T(t) \end{bmatrix}^T, \tag{2.15}$$

with simple dynamics given by

$$\dot{\mathbf{x}}_V(t) = 0, \tag{2.16}$$

which means that both are assumed, in a deterministic framework, as constant. In the filtering framework, the inclusion of state disturbances allows to consider them as slowly time-varying. Note that assuming that the velocity of the vehicle is constant (or indeed, slowly time-varying) in the body-fixed frame is not as strong as assuming that it is so in the inertial frame, where it would imply uniform motion. In fact, the adjustment of the filtering parameters will allow the algorithm to overcome this constraint, as it will be shown in the later chapters of this dissertation.

A landmark state vector is an agglomeration of a position landmark and its associated directions. Each one of these landmark vectors constitute the landmark set $\mathcal{I}$, which is itself divided in the two complementary sets $\mathcal{I}_O$ and $\mathcal{I}_U$. The first is constituted by the N_O observed landmarks, also designated as visible, and the latter by the N_U unobserved or non-visible landmarks. Any landmark set is itself a map, with a dynamically varying dimension depending on what is observed throughout time.

It is now possible to proceed to derive the full state dynamics. For that purpose, consider the i-th

landmark vector $\mathbf{m}_i(t)$, $i \in \mathcal{I}$, where the respective landmark position and associated directions are agglomerated,

$$\mathbf{m}_i(t) := \begin{bmatrix} \mathbf{p}_i^T(t) & \mathbf{d}_{i_1}^T(t) & \mathbf{d}_{i_2}^T(t) & \mathbf{d}_{i_3}^T(t) \end{bmatrix}^T, \tag{2.17}$$

and the position and direction landmarks dynamics (2.13) and (2.14) respectively, which may now be expressed as a function of the vehicle and landmark state vectors, yielding

$$\dot{\mathbf{m}}_i(t) = \mathbf{A}_{MV_i}(\mathbf{m}_i(t))\, \mathbf{x}_V(t) + \mathbf{A}_{M_i}(t)\mathbf{m}_i(t), \tag{2.18}$$

where

$$\mathbf{A}_{MV_i}(\mathbf{m}_i(t)) = \begin{bmatrix} -\mathbf{I}_3 & -\mathbf{S}\left[\mathbf{p}_i(t)\right] \\ \mathbf{0}_3 & -\mathbf{S}\left[\mathbf{d}_{i_1}(t)\right] \\ \mathbf{0}_3 & -\mathbf{S}\left[\mathbf{d}_{i_2}(t)\right] \\ \mathbf{0}_3 & -\mathbf{S}\left[\mathbf{d}_{i_3}(t)\right] \end{bmatrix} \quad \text{and} \quad \mathbf{A}_{M_i}(t) = \begin{bmatrix} -\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right] & \cdots & \mathbf{0}_3 \\ \vdots & \ddots & \vdots \\ \mathbf{0}_3 & \cdots & -\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right] \end{bmatrix}.$$

Finally, the visible landmarks vector $\mathbf{x}_O(t) = \{\mathbf{m}_i(t)\} \in \mathbb{R}^{n_O}$, $i \in \mathcal{I}_O$, and the non-visible one $\mathbf{x}_U(t) = \{m_j(t)\} \in \mathbb{R}^{n_U}$, $j \in \mathcal{I}_U$, are concatenated in the landmark-based state vector, $\mathbf{x}_M(t) := \begin{bmatrix} \mathbf{x}_O^T(t) & \mathbf{x}_U^T(t) \end{bmatrix}^T \in \mathbb{R}^{n_M}$, thus labelling landmarks 1 to N_O as visible and landmarks $N_O + 1$ to $N_M := N_O + N_U$ as non-visible, without loss of generality. The two state vectors here defined constitute the full state vector $\mathbf{x}_F(t) := \begin{bmatrix} \mathbf{x}_V^T(t) & \mathbf{x}_M^T(t) \end{bmatrix}^T \in \mathbb{R}^{n_V + n_M}$, with the full state dynamics reading as

$$\begin{cases} \dot{\mathbf{x}}_F(t) = \mathbf{A}_F(t, \mathbf{x}_M(t))\mathbf{x}_F(t) \\ \mathbf{y}(t) = \mathbf{x}_O(t) \end{cases}, \tag{2.19}$$

with the dynamics matrix given by

$$\mathbf{A}_F(t, \mathbf{x}_M(t)) = \begin{bmatrix} \mathbf{0}_{n_V} & \mathbf{0}_{n_V \times n_M} \\ \mathbf{A}_{MV}(t, \mathbf{x}_M(t)) & \mathbf{A}_M(t) \end{bmatrix},$$

where the submatrices $\mathbf{A}_{MV}(t, \mathbf{x}_M(t))$ and $\mathbf{A}_M(t)$ are

$$\mathbf{A}_{MV}(t, \mathbf{x}_M(t)) = \begin{bmatrix} \mathbf{A}_{MV_1}^T(\mathbf{m}_1(t)) & \cdots & \mathbf{A}_{MV_{N_M}}^T(\mathbf{m}_{N_M}(t)) \end{bmatrix}$$

and

$$\mathbf{A}_M(t) = \operatorname{diag}\left(\mathbf{A}_{M_i}(t), \cdots, \mathbf{A}_{M_i}(t)\right),$$

respectively. From (2.19) it follows that the system may be expressed in a way similar to the usual linear dynamical system form. However, close observation leads to the conclusion that the system now designed is nonlinear, as the dynamics matrix depends explicitly on the landmarks that may be visible or not. Note that no linearization was performed, and that all the derivations here presented

are purely kinematic and therefore exact. Further analysing the dependence of the dynamics matrix on $\mathbf{x}_M(t)$, it follows that the subset of visible landmarks is observed, being the same as the output of the system, $\mathbf{y}(t) := \begin{bmatrix}\mathbf{y}_1(t) & \cdots & \mathbf{y}_{N_O}(t)\end{bmatrix} \in \mathbb{R}^{n_O}$. Strictly speaking, the dynamics matrix $\mathbf{A}(t, \mathbf{x}_M(t))$ depends on the system output $\mathbf{y}(t)$ and on the non-visible landmarks vector $\mathbf{x}_U(t)$.

Although the formulation here presented includes landmarks in all their forms, in the remainder of this work it is assumed that only position landmarks are present, and therefore, in all their subsequent uses, the landmark agglomerated vector $\mathbf{m}_i(t)$ is simply $\mathbf{p}_i(t)$ and the matrices $\mathbf{A}_{MV_i}(\mathbf{m}_i(t))$ and $\mathbf{A}_{M_i}(t)$ are redefined as $\mathbf{A}_{MV_i}(\mathbf{p}_i(t)) = \begin{bmatrix}-\mathbf{I}_3 & -\mathbf{S}\left[\mathbf{p}_i(t)\right]\end{bmatrix}$ and $\mathbf{A}_{M_i}(t) = -\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right]$ for all the $i \in \mathcal{I}$.

2.2 Observability analysis

Observability is of the utmost importance in any filtering problem, especially when in presence of nonlinear or quasi-linear systems. Although the observability of linear systems is well-documented in linear systems theory, the analysis of the observability of nonlinear systems lacks an expedite process, and is still an open field of research. The work presented in this section aims at analysing the observability of the dynamical system previously exposed, by establishing theoretical results on various degrees of observability of the proposed SLAM problem. It is important to notice that, although system (2.19) is inherently nonlinear, discarding the non-visible landmarks $\mathbf{x}_U(t)$ makes it possible to regard the resulting system as linear time-varying (LTV). Recall that only the dynamics of $\mathbf{x}_U(t)$ depends on the non-visible landmarks, and in fact, the remaining landmarks are the same as the output of the system. Linear systems theory may therefore be employed, which enables the establishment of sufficient conditions for the observability and uniform complete observability of the LTV system with a physical interpretation. These results are then extended to the original nonlinear system, allowing the design of a state observer with globally asymptotically error dynamics. The whole procedure is succinctly exposed in Fig. 2.3.

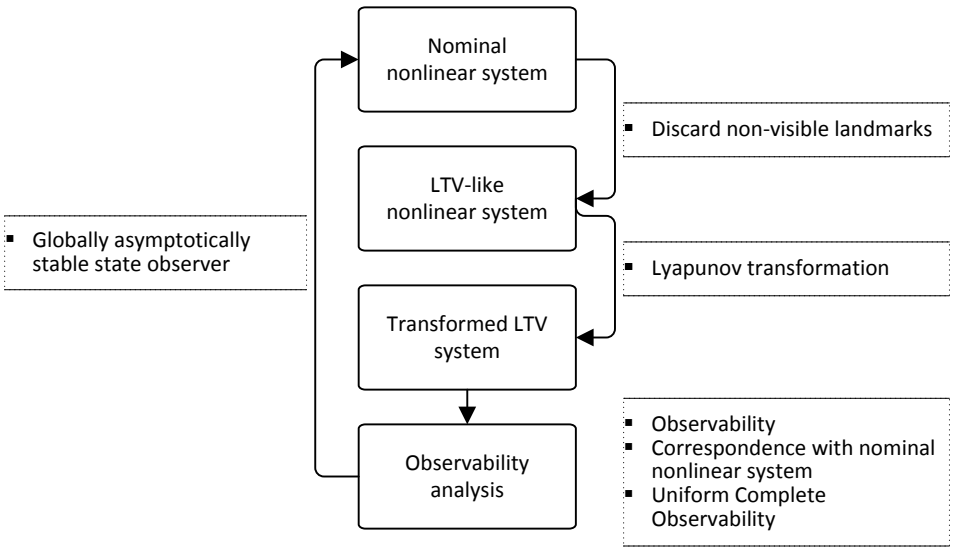


Figure 2.3: The procedure of the observability analysis.

Consider then the new state vector $\mathbf{x}(t) := \begin{bmatrix}\mathbf{x}_V^T(t) & \mathbf{x}_O(t)\end{bmatrix}^T \in \mathbb{R}^{n_V+n_O}$, which does not include the non-visible landmarks, for which the resulting system dynamics can be written as

$$\begin{cases}\dot{\mathbf{x}}(t) = \mathbf{A}(t,\mathbf{y}(t))\mathbf{x}(t)\\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)\end{cases} \tag{2.20}$$

where the equivalence between $\mathbf{y}(t)$ and $\mathbf{x}_O(t)$ was employed. Also,

$$\mathbf{A}(t,\mathbf{y}(t)) = \begin{bmatrix}\mathbf{0}_{n_V} & \mathbf{0}_{n_V\times n_M}\\ \mathbf{A}_{MV_O}(t,\mathbf{y}(t)) & \mathbf{A}_{M_O}(t)\end{bmatrix} \qquad \text{and} \qquad \mathbf{C} = \begin{bmatrix}\mathbf{0}_{n_O\times n_V} & \mathbf{I}_{n_O}\end{bmatrix},$$

with the submatrices $\mathbf{A}_{MV_O}(t,\mathbf{y}(t))$ and $\mathbf{A}_{M_O}(t,\mathbf{y}(t))$ being

$$\mathbf{A}_{MV_O}(t,\mathbf{y}(t)) = \begin{bmatrix}\mathbf{A}_{MV_1}^T(\mathbf{y}_1(t)) & \cdots & \mathbf{A}_{MV_{N_O}}^T(\mathbf{y}_{N_O}(t))\end{bmatrix} \in \mathbb{R}^{n_O\times n_V}$$

and

$$\mathbf{A}_{M_O}(t) = \operatorname{diag}\left(\mathbf{A}_{M_i}(t),\cdots,\mathbf{A}_{M_i}(t)\right) \in \mathbb{R}^{n_O\times n_O}.$$

Note that the matrix $\mathbf{A}(t,\mathbf{y}(t))$ depends not only on time but also on the system output. Nevertheless, the dependency on the system state is now absent and the system output is known for all $t \geq t_0$, thus, the system can be seen as a linear time-varying system for observability purposes. In fact, if the system output is simply considered as a function of t, the dynamics matrix becomes a truly time-varying matrix, $\mathbf{A}(t)$. The same approach was applied in [1].

The observability analysis follows now in a linear fashion, by defining two important quantities, the transition matrix and the observability Gramian associated with the system. These are presented in the following definitions.

Definition 1 (Transition matrix)**.** *Consider the generic system*

$$\begin{cases}\dot{\mathbf{x}}(t) = \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)\\ \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t).\end{cases} \tag{2.21}$$

The transition matrix $\phi(t,t_0)$ of a given state space model maps the state from time instant t_0 to time t. The general solution of the generic system (2.21) *is given by*

$$\mathbf{x}(t) = \boldsymbol{\phi}(t,t_0)\mathbf{x}(t_0) + \int_{t_0}^{t}\boldsymbol{\phi}(t,\tau)\mathbf{B}(\tau)\mathbf{u}(\tau)d\tau, \tag{2.22}$$

where the transition matrix $\phi(t,t_0)$ must always satisfy

$$\frac{d}{dt}\boldsymbol{\phi}(t,t_0) = \mathbf{A}(t)\boldsymbol{\phi}(t,t_0),$$

with the initial condition $\phi(t_0, t_0) = \mathbf{I}$.

The observability gramian of a given system is associated with the observability properties of the system.

Definition 2 (Observability Gramian)**.** *The observability gramian in the time interval* $\mathcal{T} = [t_0, t_f]$ *of system* (2.21) *is defined as*

$$\boldsymbol{\mathcal{W}}(t_0, t_f) := \int_{t_0}^{t_f} \boldsymbol{\phi}^T(\tau, t_0)\mathbf{C}^T(\tau)\mathbf{C}(\tau)\boldsymbol{\phi}(\tau, t_0)d\tau, \tag{2.23}$$

where $\phi(\tau, t_0)$ *is the transition matrix associated with* (2.21)*. If and only if* $\boldsymbol{\mathcal{W}}(t_0, t_f)$ *is non-singular in* $\mathcal{T}$*, then the system* (2.21) *is observable in the same time interval.*

With these quantities now defined, consider the following lemma that addresses the observability analysis for linear time-varying systems whose dynamics matrix depend on the system output and input.

Lemma 1 ([1, Lemma 1, Section 3])**.** *Consider the nonlinear system*

$$\begin{cases} \dot{\mathbf{x}}(t) = \boldsymbol{\mathcal{A}}(t, \mathbf{u}(t), \mathbf{y}(t))\,\mathbf{x}(t) + \boldsymbol{\mathcal{B}}(t)\mathbf{u}(t) \\ \mathbf{y}(t) = \boldsymbol{\mathcal{C}}(t)\mathbf{x}(t). \end{cases} \tag{2.24}$$

If the observability Gramian $\boldsymbol{\mathcal{W}}(t_0, t_f)$*, defined in Definition 2, associated with the pair*

$$(\boldsymbol{\mathcal{A}}(t, \mathbf{u}(t), \mathbf{y}(t)), \boldsymbol{\mathcal{C}}(t))$$

on $\mathcal{T} = [t_0, t_f]$ *is invertible then the nonlinear system* (2.24) *is observable in the sense that, given the system input* $\{\mathbf{u}(t), t \in \mathcal{T}\}$ *and the system output* $\{\mathbf{y}(t), t \in \mathcal{T}\}$*, the initial condition* $\mathbf{x}(t_0)$ *is uniquely defined.*

Proof. The proof can be found in [1, Section 3]. □

This result will be used throughout this section, with special emphasis in the theorem that follows. However, before proceeding with this analysis, the following assumption is considered.

Assumption 1 (Distinguishable landmarks)*.* Any two detected position landmarks are assumed to be different and nonzero, i.e., $\mathbf{y}_i(t), \mathbf{y}_j(t) \neq \mathbf{0}$ and $\mathbf{y}_i(t) \neq \mathbf{y}_j(t)$ for all $t \geq t_0$ and $i, j \in \mathcal{I}_O$.

Notice that this is a very mild assumption, as it is physically impossible to have two collinear landmarks, let alone equal, visible at the same time, because of the intrinsic characteristics of the camera: its angle of view is always inferior to $180°$. Also, as in the body-fixed frame, in which the output is expressed, the origin denotes the position of the vehicle, it is impossible to have it coincide with a landmark.

As shown in Fig. 2.3, the system (2.20) is transformed using a Lyapunov transformation in order to simplify the observability analysis. Consider then the definition of a Lyapunov transformation that follows.

Definition 3 (Lyapunov Transformation [4, Chapter 1, Section 8]). *A transformation* $\mathbf{z}(t) = \mathbf{L}(t)\mathbf{x}(t)$, $\mathbf{L}(t) \in \mathbb{R}^{N\times N}$ *is called a Lyapunov transformation if*

i. $\mathbf{L}$ *has a continuous derivative on the interval* $(-\infty, \infty)$,

ii. $\mathbf{L}$ *and* $\dot{\mathbf{L}}$ *are bounded on the interval* $(-\infty, \infty)$, *in the sense that* $|l_{ij}(t)| \leq M\ \forall i, j \in [1, N]$ *and* $M < \infty$ *for all* $-\infty < t < \infty$, *where* $l_{ij}(t)$ *is the entry in the* i*-th row and* j*-th of* $\mathbf{L}(t)$,

iii. there exists a constant m *such that* $0 < m \leq |\det(\mathbf{L}(t))|$ *for all* t.

Finally, the following theorem addresses the analysis of the observability of system (2.20).

Theorem 1 (Observability). *Consider system* (2.20) *and let* $\mathcal{T} := [t_0, t_f]$ *and* $\{t_1, t_2, t_3\} \in \mathcal{T}$. *The system is observable on* $\mathcal{T}$ *in the sense that, given the system output, the initial condition is uniquely defined, if at least one of these conditions holds:*

(i) There are, at least, three visible position landmarks at the same time t_1 *that define a plane.*

(ii) There exist two visible position landmarks in the interval $[t_1, t_2]$ *such that at least one of the landmark sets* $\{\mathbf{p}_1(t_1), \mathbf{p}_2(t_1), \mathbf{p}_2(t_2)\}$ *and* $\{\mathbf{p}_1(t_1), \mathbf{p}_2(t_1), \mathbf{p}_1(t_2)\}$ *defines a plane.*

(iii) There is a visible time-varying position landmark whose coordinates, $\{\mathbf{p}_1(t_1), \mathbf{p}_1(t_2), \mathbf{p}_1(t_3)\}$, *define a plane.*

Proof. The proof follows by transforming the system in question by means of a Lyapunov transformation in order to simplify the analysis, and then showing that the observability Gramian of the transformed system is non-singular.

Let $\mathbf{R}_m(t) \in \mathrm{SO}(3)$ be a rotation matrix respecting $\dot{\mathbf{R}}_m(t) = \mathbf{R}_m(t)\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right]$ and $\mathbf{T}(t)$ be a Lyapunov transformation such that

$$\mathbf{z}(t) = \mathbf{T}(t)\ \mathbf{x}(t), \tag{2.25}$$

where the transformation $\mathbf{T}(t) \in \mathbb{R}^{(n_V+n_O)\times(n_V+n_O)}$ and its derivative $\dot{\mathbf{T}}(t) \in \mathbb{R}^{(n_V+n_O)\times(n_V+n_O)}$ are given by

$$\mathbf{T}(t) = \mathrm{diag}\left(\mathbf{I}_{n_V}, \mathbf{R}_m(t), \cdots, \mathbf{R}_m(t)\right) \tag{2.26}$$

and

$$\dot{\mathbf{T}}(t) = \mathrm{diag}\left(\mathbf{0}_{n_V}, \dot{\mathbf{R}}_m(t), \cdots, \dot{\mathbf{R}}_m(t)\right), \tag{2.27}$$

respectively. Substitution of the dynamics of $\mathbf{R}_m(t)$ in (2.27) yields

$$\dot{\mathbf{T}}(t) = \mathrm{diag}\left(\mathbf{0}_{n_V}, \mathbf{R}_m(t)\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right], \cdots, \mathbf{R}_m(t)\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right]\right). \tag{2.28}$$

Also, as $\mathbf{T}(t)$ is a sort of rotation matrix, its inverse and its transpose are one and the same, $\mathbf{T}^{-1}(t) = \mathbf{T}^T(t)$. It is now straightforward to see that $\mathbf{T}(t)$ respects all the characteristics of a Lyapunov trans-

formation described in Definition 3: $\dot{\mathbf{T}}(t)$ is continuous and bounded, $\mathbf{T}(t)$ is bounded, and its determinant is $\det(\mathbf{T}(t)) = 1$ for all t. Differentiation of (2.25) to time, and use of the reverse transformation $\mathbf{x}(t) = \mathbf{T}^{-1}(t)\mathbf{z}(t)$, yields

$$\begin{cases} \dot{\mathbf{z}}(t) = \left(\dot{\mathbf{T}}(t)\mathbf{T}^{-1}(t) + \mathbf{T}(t)\mathbf{A}(t)\mathbf{T}^{-1}(t)\right)\mathbf{z}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{T}^{-1}(t)\ \mathbf{z}(t). \end{cases} \tag{2.29}$$

For the sake of clarity, let $\boldsymbol{\mathcal{A}}(t, \mathbf{y}(t)) := \dot{\mathbf{T}}(t)\mathbf{T}^{-1}(t) + \mathbf{T}(t)\mathbf{A}(t)\mathbf{T}^{-1}(t)$ and $\boldsymbol{\mathcal{C}}(t) := \mathbf{C}\mathbf{T}^{-1}(t)$ be the system dynamics and output matrices, respectively. The computation of $\boldsymbol{\mathcal{A}}(t)$ and $\boldsymbol{\mathcal{C}}(t)$ is simple, yielding

$$\begin{cases} \dot{\mathbf{z}}(t) = \boldsymbol{\mathcal{A}}(t, \mathbf{y}(t))\mathbf{z}(t) \\ \mathbf{y}(t) = \boldsymbol{\mathcal{C}}(t)\mathbf{z}(t) \end{cases}, \tag{2.30}$$

with

$$\boldsymbol{\mathcal{A}}(t) = \begin{bmatrix} \mathbf{0}_{n_V} & \mathbf{0}_{n_V \times n_O} \\ \boldsymbol{\mathcal{A}}_{MV}(t) & \mathbf{0}_{n_O} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{C}}(t) = \begin{bmatrix} \mathbf{0}_{n_O \times n_V} & \operatorname{diag}\left(\mathbf{R}_m^T(t), \cdots, \mathbf{R}_m^T(t)\right) \end{bmatrix},$$

where

$$\boldsymbol{\mathcal{A}}_{MV}(t) = \begin{bmatrix} \boldsymbol{\mathcal{A}}_{MV_1}(t) & \cdots & \boldsymbol{\mathcal{A}}_{MV_{N_O}}(t) \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{A}}_{MV_i}(t) = \begin{bmatrix} -\mathbf{R}_m(t) & -\mathbf{R}_m(t)\mathbf{S}\left[\mathbf{p}_i(t)\right] \end{bmatrix}.$$

A Lyapunov transformation preserves the observability properties of a system, hence, to prove the observability of the original system it suffices to prove that (2.30) is observable, which is done by showing that the observability Gramian associated with that system is invertible. This approach has been used successfully in the past, see [1] and [21]. Before computing the observability Gramian associated with (2.30), it is necessary to know its transition matrix. A simple computation of $\mathbf{z}(t)$ as a function of the initial condition $\mathbf{z}(t_0)$ by solving $\boldsymbol{\phi}(t, t_0)\mathbf{z}(t_0) = \mathbf{z}(t_0) + \int_{t_0}^{t} \boldsymbol{\mathcal{A}}(\tau, \mathbf{y}(\tau))\mathbf{z}(\tau)d\tau$ yields

$$\boldsymbol{\phi}(t, t_0) = \begin{bmatrix} \mathbf{I}_{n_V} & \mathbf{0}_{n_V \times n_O} \\ \boldsymbol{\phi}_{MV}(t, t_0) & \mathbf{I}_{n_O} \end{bmatrix}, \tag{2.31}$$

where $\boldsymbol{\phi}_{MV_i}(t, t_0) = \left[-\int_{t_0}^{t} \mathbf{R}_m(\sigma)d\sigma \quad -\int_{t_0}^{t} \mathbf{R}_m(\sigma)\mathbf{S}\left[\mathbf{p}_i(\sigma)\right]d\sigma\right]$. According to Lemma 1, if $\boldsymbol{\mathcal{W}}(t_0, t_f)$ is invertible, system (2.30) is observable, in the sense that given the system input and output, the initial condition $\mathbf{z}(t_0)$ is uniquely defined. This will be the focus of the next steps, which mean to prove, by contradiction, that the observability Gramian is indeed invertible, i.e., by assuming the $\boldsymbol{\mathcal{W}}(t_0, t_f)$ is singular, and showing that none of the conditions of the theorem can hold in that case. Consider the existence a unit vector $\mathbf{c} \in \mathbb{R}^{6+3N_O}$,

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T & \mathbf{c}_3^T & \cdots & \mathbf{c}_{2+N_O}^T \end{bmatrix}^T,$$

such that the observability Gramian is singular, i.e.,

$$\underset{\substack{\mathbf{c} \in \mathbb{R}^{n_x} \\ \|\mathbf{c}\| = 1}}{\exists} : \quad \mathbf{c}^T \boldsymbol{\mathcal{W}}(t_0, t_f)\mathbf{c} = 0. \tag{2.32}$$

The objective now is to expand $\mathbf{c}^T\boldsymbol{\mathcal{W}}(t_0, t_f)\mathbf{c}$, in order to show that $\mathbf{c}$ being unitary implies that none of the three conditions of the theorem can hold. It is a simple matter of computation to show that $\mathbf{c}^T\boldsymbol{\mathcal{W}}(t_0, t_f)\mathbf{c}$ is

$$\begin{aligned}
\mathbf{c}^T\boldsymbol{\mathcal{W}}(t_0, t_f)\mathbf{c} &= \int_{t_0}^{t_f} \mathbf{c}^T\boldsymbol{\phi}^T(\tau, t_0)\boldsymbol{\mathcal{C}}^T(\tau)\boldsymbol{\mathcal{C}}(\tau)\boldsymbol{\phi}(\tau, t_0)\mathbf{c}d\tau \\
&= \int_{t_0}^{t_f} \left(\boldsymbol{\mathcal{C}}(\tau)\boldsymbol{\phi}(\tau, t_0)\mathbf{c}\right)^T \boldsymbol{\mathcal{C}}(\tau)\boldsymbol{\phi}(\tau, t_0)\mathbf{c}d\tau \\
&= \int_{t_0}^{t_f} \|\mathbf{f}(\tau, t_0)\|^2 d\tau,
\end{aligned} \tag{2.33}$$

where $\mathbf{f}(\tau, t_0) \in \mathbb{R}^{n_O}$ is given by $\mathbf{f}(\tau, t_0) = \boldsymbol{\mathcal{C}}(\tau)\boldsymbol{\phi}(\tau, t_0)\mathbf{c}$. The evaluation of $\mathbf{f}(\tau, t_0)$ yields

$$\mathbf{f}(\tau, t_0) = \operatorname{diag}\left(\mathbf{R}_m^T(\tau), \cdots, \mathbf{R}_m^T(\tau)\right)\begin{bmatrix}\boldsymbol{\phi}_{MV}(\tau, t_0) & \mathbf{I}_{n_O}\end{bmatrix}\mathbf{c}. \tag{2.34}$$

Recall that a rotation matrix preserves the norm of the vector to which it is applied, and a higher-dimension rotation matrix as $\operatorname{diag}\left(\mathbf{R}_m^T(\tau), \cdots, \mathbf{R}_m^T(\tau)\right)$ is no different. Thus, the new function $\mathbf{g}(\tau, t_0) \in \mathbb{R}^{n_O}$ may be defined, with $\|\mathbf{g}(\tau, t_0)\|^2 = \|\mathbf{f}(\tau, t_0)\|^2$, such that

$$\mathbf{g}(\tau, t_0) := \operatorname{diag}\left(\mathbf{R}_m(\tau), \cdots, \mathbf{R}_m(\tau)\right)\mathbf{f}(\tau, t_0)$$

and its derivative $\frac{d}{d\tau}\mathbf{g}(\tau, t_0) \in \mathbb{R}^{n_O}$ are given by

$$\mathbf{g}(\tau, t_0) = \begin{bmatrix} \mathbf{c}_3 - \int_{t_0}^{\tau} \mathbf{R}_m(\sigma)\mathbf{c}_1 d\sigma - \int_{t_0}^{\tau} \mathbf{R}_m(\sigma)\mathbf{S}\left[\mathbf{p}_1(\sigma)\right]\mathbf{c}_2 d\sigma \\ \vdots \\ \mathbf{c}_{2+N_O} - \int_{t_0}^{\tau} \mathbf{R}_m(\sigma)\mathbf{c}_1 d\sigma - \int_{t_0}^{\tau} \mathbf{R}_m(\sigma)\mathbf{S}\left[\mathbf{p}_{N_O}(\sigma)\right]\mathbf{c}_2 d\sigma \end{bmatrix} \tag{2.35}$$

and

$$\frac{d}{d\tau}\mathbf{g}(\tau, t_0) = \begin{bmatrix} -\mathbf{R}_m(\tau)\mathbf{c}_1 - \mathbf{R}_m(\tau)\mathbf{S}\left[\mathbf{p}_1(\tau)\right]\mathbf{c}_2 \\ \vdots \\ -\mathbf{R}_m(\tau)\mathbf{c}_1 - \mathbf{R}_m(\tau)\mathbf{S}\left[\mathbf{p}_{N_O}(\tau)\right]\mathbf{c}_2 \end{bmatrix}, \tag{2.36}$$

respectively. In order for (2.32) to be true, both $\mathbf{g}(\tau, t_0)$ and $\frac{d}{d\tau}\mathbf{g}(\tau, t_0)$ must be zero for all $\tau \in \mathcal{T}$. Setting $\tau = t_0$ in $\mathbf{g}(\tau, t_0)$ immediately yields $\mathbf{c}_3 = \cdots = \mathbf{c}_{2+N_O} = 0$. The equality $\frac{d}{d\tau}\mathbf{g}(\tau, t_0) = \mathbf{0}$ implies

that

$$\begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_1(\tau)\right] \\ \vdots & \vdots \\ \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_{N_O}(\tau)\right] \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \mathbf{0}, \quad \forall \tau \in \mathcal{T} \tag{2.37}$$

or $\mathbf{\Gamma}(\tau)\chi = \mathbf{0}$, where $\mathbf{\Gamma}(\tau) = \begin{bmatrix} \mathbf{\Gamma}_1^T(\tau) & \cdots & \mathbf{\Gamma}_{N_O}^T(\tau) \end{bmatrix}^T$ with $\mathbf{\Gamma}_i(\tau) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_i(\tau)\right] \end{bmatrix}$ and $\chi = \begin{bmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T \end{bmatrix}^T$. Thus, it remains to show that, under the conditions of Theorem 1, matrix $\mathbf{\Gamma}(\tau)$ has full rank, i.e., that the only possible solution is $\mathbf{c}_1 = \mathbf{c}_2 = \mathbf{0}$. Consider then the situation where there are three observed landmarks $\mathbf{p}_i(t_1)$, $i \in \{1, 2, 3\}$. In this case, (2.37) can be rewritten as

$$\begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_2(t_1) - \mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_3(t_1) - \mathbf{p}_1(t_1)\right] \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \mathbf{0}, \tag{2.38}$$

after subtracting the first row to the other two. From this, it is simple to find that either $\mathbf{c_2} = \mathbf{0}$ or

$$\mathbf{c}_2 = \alpha\left(\mathbf{p}_2(t_1) - \mathbf{p}_1(t_1)\right) = \beta\left(\mathbf{p}_3(t_1) - \mathbf{p}_1(t_1)\right). \tag{2.39}$$

The first possibility immediately makes $\mathbf{c}_1 = \mathbf{0}$, contradicting the hypothesis that the observability Gramian is singular. The second possibility implies that the three landmarks are related as special linear combination of each other, i.e.,

$$\mathbf{p}_1(t_1) = \frac{\alpha}{\alpha - \beta}\mathbf{p}_2(t_1) - \frac{\beta}{\alpha - \beta}\mathbf{p}_3(t_1). \tag{2.40}$$

Note that the sum of the coefficients affecting $\mathbf{p}_2(t_1)$ and $\mathbf{p}_3(t_1)$ is unitary. This means that, if (2.39) is true, all three landmarks form a line, which contradicts the hypothesis of the theorem. In fact, it can easily be shown that for any number of landmarks, if all form a line, the null space of $\mathbf{\Gamma}(t_1)$ is infinite. Hence, either $\mathbf{c}$ is not a unit vector or condition (i) cannot hold, thus contradicting the original hypothesis of the proof. Consider now that there exist two landmarks $\mathbf{p}_1(t)$ and $\mathbf{p}_2(t)$ visible in the interval $[t_1, t_2]$. Then, an equation similar to (2.38) may be constructed, using the landmark set $\{\mathbf{p}_1(t_1), \mathbf{p}_2(t_1), \mathbf{p}_1(t_2)\}$, $\{\mathbf{p}_1(t_1), \mathbf{p}_2(t_1), \mathbf{p}_2(t_2)\}$ or both,

$$\begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_2(t_1) - \mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_2(t_2) - \mathbf{p}_1(t_1)\right] \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \mathbf{0} \quad \text{or} \qquad \begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_2(t_1) - \mathbf{p}_1(t_1)\right] \\ \mathbf{0}_3 & \mathbf{S}\left[\mathbf{p}_1(t_2) - \mathbf{p}_1(t_1)\right] \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \mathbf{0} \tag{2.41}$$

This leads once more to $\mathbf{c}_2 = \mathbf{0}$, which implies $\mathbf{c}_1 = \mathbf{0}$ and therefore contradicts the assumption of $\mathbf{c}$ being a unit vector, or to both sets forming a line, which once more contradicts condition (ii) of the theorem. Finally, let $\mathbf{p}_1(t)$ be the only visible landmark on an interval containing t_1, t_2 and t_3. Using the set $\{\mathbf{p}_1(t_1), \mathbf{p}_1(t_2), \mathbf{p}_1(t_3)\}$ to build another equation in the vein of (2.38) and (2.41), either contradicts condition (iii) or leads to $\mathbf{c}_2 = \mathbf{c}_1 = \mathbf{0}$, hence contradicting the assumption that the $\mathcal{W}(t_0, t_f)$ is singular.

Therefore, if the observability Gramian is not invertible, none of the hypothesis of the theorem can hold, which means that, if at least one of the conditions of Theorem 1 holds, then $\mathcal{W}(t_0, t_f)$ is invertible on $\mathcal{T}$, and, using Lemma 1, it follows that (2.30) is observable. Moreover, as the Lyapunov transformation (2.25) preserves observability, the nonlinear system (2.20), regarded as LTV, is also observable, thus concluding the proof of the theorem. □

Given the sufficient conditions for observability, a Kalman Filter for the nonlinear system (2.19), with globally asymptotically stable error dynamics, can be designed following the classical approach. The following result addresses the equivalence between the state of the nonlinear system (2.20), regarded as LTV, and that of the nominal nonlinear system (2.19), when the non-visible landmarks are not considered.

Theorem 2 (Equivalence of the LTV and the nonlinear systems)**.** *Suppose that the conditions of Theorem 1 hold. Then,*

(i) the initial state of the nonlinear system (2.19)*, discarding the non-visible landmarks, is uniquely determined, and it is the same of the nonlinear system* (2.20)*, regarded as LTV;*

(ii) a state observer with uniformly globally asymptotically stable error dynamics for the LTV system is also a state observer for the underlying nonlinear system, with uniformly globally asymptotically stable error dynamics.

Proof. Consider the transformed system (2.30), whose state, and therefore initial condition, is related with the state of the nonlinear, regarded as LTV, system (2.20). The proof follows then with the transformed system for simplicity of analysis. Let the initial condition for this system be given by

$$\bar{\mathbf{z}}_0 : \mathbf{z}(t_0) = \begin{bmatrix} \bar{\mathbf{v}}^T(t_0) & \bar{\mathbf{b}}_\omega^T(t_0) & \bar{\mathbf{z}}_{p_1}^T(t_0) & \cdots & \bar{\mathbf{z}}_{p_N}^T(t_0) \end{bmatrix}^T, \tag{2.42}$$

where $\bar{\mathbf{z}}_{p_i}(t) = \mathbf{R}_m(t)\mathbf{y}_i(t)$, which follows from $\mathbf{y}(t) = \boldsymbol{\mathcal{C}}(t)\mathbf{z}(t)$. Recall that $\mathbf{z}(t) = \boldsymbol{\phi}(t, t_0)\mathbf{z}(t_0)$, which then yields $\mathbf{y}(t) = \boldsymbol{\mathcal{C}}(t)\boldsymbol{\phi}(t, t_0)\mathbf{z}(t_0)$. As seen before, the output of the system is composed by N_O landmarks, $\mathbf{y}(t) = \begin{bmatrix} \mathbf{y}_1^T(t) & \cdots & \mathbf{y}_{N_O}^T(t) \end{bmatrix}^T$ and its i-th component $\mathbf{y}_i(t) \in \mathbb{R}^3$, $i \in \mathcal{I}_O$, is given by

$$\mathbf{y}_i(t) = \mathbf{R}_m^T(t)\mathbf{R}_m(t_0)\mathbf{y}_i(t_0) - \mathbf{R}_m^T(t) \int_{t_0}^{t} \mathbf{R}_m(\sigma) \left(\bar{\mathbf{v}}(t_0) + \mathbf{S}\left[\mathbf{y}_i(\sigma)\right] \bar{\mathbf{b}}_\omega(t_0) \right) d\sigma, \tag{2.43}$$

where the relation between $\mathbf{y}_i(t_0)$ and $\bar{\mathbf{z}}_{p_i}^T(t_0)$ was used. Left-multiplying both sides of this expression by $\mathbf{R}_m(t)$ gives

$$\mathbf{R}_m(t)\mathbf{y}_i(t) = \mathbf{R}_m(t_0)\mathbf{y}_i(t_0) - \int_{t_0}^{t} \mathbf{R}_m(\sigma) \left(\bar{\mathbf{v}}(t_0) + \mathbf{S}\left[\mathbf{y}_i(\sigma)\right] \bar{\mathbf{b}}_\omega(t_0) \right) d\sigma, \tag{2.44}$$

making the steps to obtain the first derivative of $\mathbf{y}_i(t)$ easier. Thus, the differentiation of (2.44), where

the rotation matrix property $\dot{\mathbf{R}}_m(t) = \mathbf{R}_m(t)\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right]$ is used, yields

$$\mathbf{R}_m(t)\mathbf{S}\left[\boldsymbol{\omega}_m(t)\right]\mathbf{y}_i(t) + \mathbf{R}_m(t)\dot{\mathbf{y}}_i(t) = -\mathbf{R}_m(t)\left(\bar{\mathbf{v}}(t_0) + \mathbf{S}\left[\mathbf{y}_i(t)\right]\bar{\mathbf{b}}_\omega(t_0)\right), \tag{2.45}$$

which may be further simplified by left-multiplying both sides by $\mathbf{R}_m(t)^T$, yielding $\dot{\mathbf{y}}(t) \in \mathbb{R}^3$,

$$\dot{\mathbf{y}}_i(t) = -\bar{\mathbf{v}}(t_0) - \mathbf{S}\left[\mathbf{y}_i(t)\right]\left(\bar{\mathbf{b}}_\omega(t_0) - \boldsymbol{\omega}_m(t)\right). \tag{2.46}$$

Now consider the nonlinear system (2.19). The initial condition of this system is given by

$$\mathbf{x}(t_0) = \begin{bmatrix}\mathbf{v}^T(t_0) & \mathbf{b}_\omega^T(t_0) & \mathbf{p}_1^T(t_0) & \cdots & \mathbf{p}_N^T(t_0)\end{bmatrix}^T, \tag{2.47}$$

where $\mathbf{p}_i(t) = \mathbf{y}_i(t)$ for all $\mathbf{p}_i(t) \in \mathcal{I}_O$. The output of the system is related to the state by

$$\mathbf{y}(t) = \mathbf{C}\int_{t_0}^{t}\dot{\mathbf{x}}(\sigma)d\sigma + \mathbf{C}\mathbf{x}(t_0) = \mathbf{C}\int_{t_0}^{t}\mathbf{A}(\mathbf{x}(\sigma),\sigma)\mathbf{x}(\sigma)d\sigma + \mathbf{C}\mathbf{x}(t_0),$$

which, after a simple computation and substituting $\mathbf{p}_i(t)$ by $\mathbf{y}_i(t)$, yields

$$\mathbf{y}_i(t) = \mathbf{y}_i(t_0) - \int_{t_0}^{t}\left(\mathbf{v}(\sigma) + \mathbf{S}\left[\mathbf{y}_i(\sigma)\right]\left(\mathbf{b}_\omega(\sigma) - \boldsymbol{\omega}_m(\sigma)\right)\right)d\sigma. \tag{2.48}$$

Note that the dynamics of the nonlinear system (2.19) imply that both the linear velocity and the angular velocity bias are constant in the body-fixed frame, thus making $\mathbf{v}(t) = \mathbf{v}(t_0)$ and $\mathbf{b}_\omega(t) = \mathbf{b}_\omega(t_0)$. The first time derivative of (2.48) is given by

$$\dot{\mathbf{y}}_i(t) = -\mathbf{v}(t_0) - \mathbf{S}\left[\mathbf{y}_i(t)\right]\left(\mathbf{b}_\omega(t_0) - \boldsymbol{\omega}(t)\right). \tag{2.49}$$

Comparison of (2.46) with (2.49) yields

$$\mathbf{0} = \left(\bar{\mathbf{v}}(t_0) - \mathbf{v}(t_0)\right) - \mathbf{S}\left[\mathbf{y}_i(t)\right]\left(\bar{\mathbf{b}}_\omega(t_0) - \mathbf{b}_\omega(t_0)\right), \tag{2.50}$$

which is valid for all landmarks in $\mathcal{I}_O$ and for all t in $\mathcal{T}$. In the situation when any of the conditions of Theorem 3 applies, this system yields $\bar{\mathbf{v}}(t_0) = \mathbf{v}(t_0)$ and $\bar{\mathbf{b}}_\omega(t_0) = \mathbf{b}_\omega(t_0)$ by a similar reasoning to that used to prove the sufficiency of those conditions to the observability of the system[2]. As mentioned before, the initial condition $\bar{\mathbf{z}}(t_0)$ is related to that of the nonlinear system (2.20) by the Lyapunov transformation $\mathbf{T}(t)$. Hence, under the conditions of Theorem 1, the initial state of the nonlinear system (2.20), regarded as LTV, and the initial state of the nonlinear system (2.19), discarding the non-visible landmarks, are the same and uniquely defined.

The first part of the theorem, now proved, gives insight for the proof of the second part. An observer designed for a LTV system with globally asymptotically stable error dynamics has an estimation error

[2]Take (2.37), substitute $\mathbf{c}_1$ and $\mathbf{c}_2$ by $\bar{\mathbf{v}}(t_0) - \mathbf{v}(t_0)$ and $\bar{\mathbf{b}}_\omega(t_0) - \mathbf{b}_\omega(t_0)$ respectively, and follow the steps therein.

that converges to zero, implying that the estimates asymptotically tend to the true state. Therefore, if the true state of the nonlinear system and the state of the LTV system are one and the same, as proved, the estimation error of the state of the nonlinear system converges to zero too. □

The previous results establish the ground for the design of a filter with globally asymptotically stable (GAS) error dynamics. Given that a GAS observer for system (2.20) is an observer for the nominal nonlinear system, the design of a globally asymptotically stable observer for the LTV system follows naturally with a linear time-varying Kalman filter. This step requires that the pair $(\mathbf{A}(t,\mathbf{y}(t)),\mathbf{C})$ is uniformly completely observable as declared in [22]. The following proposition is needed before establishing the conditions for this property to be verified.

Proposition 1 ([3, Proposition 4.2, Section 4])**.** *Let $\mathbf{f}(t) : [t_0,t_f] \subset \mathbb{R} \to \mathbb{R}^n$ be a continuous and i-times differentiable function on $\mathcal{T} := [t_0,t_f]$, $T := t_f - t_0 > 0$, and such that*

$$\mathbf{f}(t_0) = \dot{\mathbf{f}}(t_0) = \cdots = \mathbf{f}^{(i-1)}(t_0) = \mathbf{0}.$$

Further assume that $\max_{t\in\mathcal{T}} \left\|\mathbf{f}^{(i-1)}(t)\right\| \leq C$. If there exists a positive constant α and an instant $t_1 \in \mathcal{T}$ such that $\left\|\mathbf{f}^{(i)}(t_1)\right\| \geq \alpha$, then there exists a positive constant β and $0 < \delta \leq T$ such that $\|\mathbf{f}(t_0+\delta)\| \geq \beta$.

Proof. The proof can be found in [3, Appendix]. □

This proposition is used throughout the following theorem, that states the conditions for the uniform complete observability of the pair $(\mathbf{A}(t,\mathbf{y}(t)),\mathbf{C})$.

Theorem 3 (Uniform Complete Observability)**.** *Consider system* (2.20)*. The pair $(\mathbf{A}(t,\mathbf{y}(t)),\mathbf{C})$ is uniformly completely observable if there exists a positive constant $\delta > 0$ such that, for all $t \geq t_0$, it is possible to choose $\{t_1,t_2,t_3\} \in \mathcal{T}_\delta$, $\mathcal{T}_\delta = [t,t+\delta]$, such that at least one of the following conditions holds:*

(i) There are at least three visible landmarks $\mathbf{p}_1(t)$, $\mathbf{p}_2(t)$ and $\mathbf{p}_3(t)$ such that $(\mathbf{p}_1(t_1) - \mathbf{p}_2(t_1)) \times (\mathbf{p}_1(t_1) - \mathbf{p}_3(t_1)) \neq 0$,

(ii) There exist two visible position landmarks at times t_1, t_2 such that at least one of the landmark sets $\{\mathbf{p}_1(t_1),\mathbf{p}_2(t_1),\mathbf{p}_2(t_2)\}$ or the $\{\mathbf{p}_1(t_1),\mathbf{p}_2(t_1),\mathbf{p}_1(t_2)\}$ defines a plane,

(iii) There is a visible time-varying position landmark whose coordinates at three different instants of time $\{t_1,t_2,t_3\}$ define a plane.

Proof. Consider again system (2.30), and recall that it is related by a Lyapunov transformation to system (2.20). The uniform complete observability of the former implies the uniform complete observability of the latter. Hence, the proof will focus on the transformed system.

The uniform complete observability of a system is equivalent to the following statement:

$$\underset{\substack{\delta>0\\ \alpha>0}}{\exists} \quad \underset{t\geq t_0}{\forall} \quad \underset{\substack{\mathbf{c}\in\mathbb{R}^{3N_O}\\ \|\mathbf{c}\|=1}}{\forall} : \quad \mathbf{c}^T\boldsymbol{\mathcal{W}}(t,t+\delta)\mathbf{c} \geq \alpha \tag{2.51}$$

From (2.33) and recalling that $\|\mathbf{f}(\tau,t)\| = \|\mathbf{g}(\tau,t)\|$, one knows that $\mathbf{c}^T\boldsymbol{\mathcal{W}}(t,t+\delta)\mathbf{c} = \int_t^{t+\delta} \|\mathbf{g}(\tau,t)\|^2 d\tau$. Recall the proof of Theorem 1 and the definition of both $\mathbf{g}(\tau,t)$, from (2.35), and $\frac{d}{d\tau}\mathbf{g}(\tau,t)$ from (2.36). Then, it suffices to prove that $\|\mathbf{g}(\tau,t)\| \geq \alpha_1^*$ or, within certain situations specified later, that $\left\|\frac{d}{d\tau}\mathbf{g}(\tau,t)\right\| \geq \alpha_2^*$, with $\tau \in \mathcal{T}_\delta$ and $\alpha_1^*, \alpha_2^* > 0$ for every possible $\mathbf{c}$, provided that the conditions of Proposition 1 are satisfied. It is a simple matter of computation to obtain the required norms,

$$\|\mathbf{g}(\tau,t)\|^2 = \sum_{i=1}^{N_O} \left\| \mathbf{c}_{2+i} - \int_t^\tau \mathbf{R}_m(\sigma_i)\left(\mathbf{c}_1 + \mathbf{S}\left[\mathbf{p}_i(\sigma_i)\right]\mathbf{c}_2\right) d\sigma_i \right\|^2, \tag{2.52}$$

and

$$\left\| \frac{d}{d\tau}\mathbf{g}(\tau,t) \right\|^2 = \sum_{i=1}^{N_O} \left\| \mathbf{c}_1 + \mathbf{S}\left[\mathbf{p}_i(\tau)\right]\mathbf{c}_2 \right\|^2. \tag{2.53}$$

The proof is made by studying all possible cases of $\mathbf{c}$, and showing that (2.51) is true for all of them.

1. In the case where there exists at least one $i \in \{1,\ldots,N_O\}$ such that $\|\mathbf{c}_{2+i}\| \neq 0$ and no restriction is imposed on $\mathbf{c}_1$ and $\mathbf{c}_2$,

$$\|\mathbf{g}(t,t)\|^2 = \sum_{j=1}^{N_O} \|\mathbf{c}_{2+j}\|^2 \geq \|\mathbf{c}_{2+i}\|^2 := \alpha_1^2 > 0.$$

2. Consider now that $\|\mathbf{c}_{2+i}\| = 0$, $\forall i \in \{1,\ldots,N_O\}$. In this situation, $\|\mathbf{g}(t,t)\| = 0$, and it is enough to prove that $\left\|\frac{d}{d\tau}\mathbf{g}(\tau,t)\right\| \geq \alpha_2^*$, with $\tau \in \mathcal{T}_\delta$ and $\alpha_2^* > 0$ for every possible $\mathbf{c}$, as Proposition 1 applies.

   (a) In the case where $\|\mathbf{c}_1\| \neq 0$ and $\|\mathbf{c}_2\| = 0$, it is straightforward to see that the norm of the derivative of $\mathbf{g}(\tau,t)$ is

$$\left\| \frac{d}{d\tau}\mathbf{g}(\tau,t) \right\|^2 = \sum_{i=1}^{N_O} \|\mathbf{c}_1\|^2 = N_O := \alpha_2^2 > 0, \tag{2.54}$$

   where $N_O \in \mathbb{N}$.

   (b) In the case where $\|\mathbf{c}_2\| \neq 0$ and $\|\mathbf{c}_1\| = 0$, $\left\|\frac{d}{d\tau}\mathbf{g}(\tau,t)\right\|^2$ becomes

$$\left\| \frac{d}{d\tau}\mathbf{g}(\tau,t) \right\|^2 = \sum_{i=1}^{N_O} \|\mathbf{S}\left[\mathbf{p}_i(\tau)\right]\mathbf{c}_2)\|^2. \tag{2.55}$$

   Note that both the norms of the landmarks and of $\mathbf{c}_2$ are greater than a given value, which means that, if condition (i) holds,

$$\left\| \frac{d}{d\tau}\mathbf{g}(\tau,t) \right\|^2 \geq \alpha_3^2 > 0. \tag{2.56}$$

(c) In the situation where both $\|\mathbf{c}_1\|$ and $\|\mathbf{c}_2\|$ are non zero it remains to prove that

$$\exists \tau \in \mathcal{T}_\delta : \quad \left\| \begin{bmatrix} \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_1(\tau)\right] \\ \vdots & \vdots \\ \mathbf{I}_3 & \mathbf{S}\left[\mathbf{p}_{N_O}(\tau)\right] \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \right\| > \alpha_2^*. \tag{2.57}$$

Note that, as the landmarks are static in the inertial frame, their relative positions are constant in both frames, which means that a plane defined by a set of landmarks at t_1 will remain formed for all time, without the possibility of degenerating. Therefore if condition (i) holds, then

$$\left\| \frac{d}{d\tau} \mathbf{g}(\tau, t) \right\|^2 \geq \alpha_4^2 > 0. \tag{2.58}$$

(d) Consider now that condition (ii) applies and that $\|\mathbf{c}_1\| \in \mathbb{R}$ and $\|\mathbf{c}_2\| \neq 0$. As $\mathbf{c}_1$ and $\mathbf{c}_2$ are constant, a similar situation is present, meaning that

$$\left\| \frac{d}{d\tau} \mathbf{g}(\tau, t) \right\|^2 = \left\| \mathbf{c}_1 + \mathbf{S}\left[\mathbf{p}_1(\tau)\right] \mathbf{c}_2 \right\|^2 + \left\| \mathbf{c}_1 + \mathbf{S}\left[\mathbf{p}_2(\tau)\right] \mathbf{c}_2 \right\|^2 > \alpha_5^2, \tag{2.59}$$

at $\tau = t_1$, t_2 or both. When there is only one visible landmark that respects the remaining hypothesis of the theorem, it is straightforward to see that

$$\left\| \frac{d}{d\tau} \mathbf{g}(\tau, t) \right\|^2 = \left\| \mathbf{c}_1 + \mathbf{S}\left[\mathbf{p}_1(\tau)\right] \mathbf{c}_2 \right\|^2 > \alpha_6^2, \tag{2.60}$$

at $\tau = t_1$, t_2, t_3 or any combination of the three.

All possible cases are now enumerated, hence the sufficiency of the conditions (i), (ii), and (iii) of Theorem 3 for the pair $(\boldsymbol{\mathcal{A}}(t, \mathbf{y}(t)), \boldsymbol{\mathcal{C}}(t))$ is proved. Because (2.25) is a Lyapunov transformation, uniform complete observability of the pair $(\boldsymbol{\mathcal{A}}(t, \mathbf{y}(t)), \boldsymbol{\mathcal{C}}(t))$ implies that of the pair $(\mathbf{A}(t, \mathbf{y}(t)), \mathbf{C}(t))$, and thus the proof of the theorem is concluded. □

The results shown in this section lead towards the design of a state observer, such as the linear time-varying Kalman filter, with globally asymptotically stable error dynamics for the nominal nonlinear system (2.19). The discrete-time implementation of such a filter is discussed in the following section.

2.3 SLAM Filter Design

This section addresses the design of a sensor-based 3-D SLAM filter. A discrete Kalman filter is designed, considering the sample-based/digital characteristics of both sensors needed for this work: an IMU (or more precisely a triad of rate-gyros) and a RGB-D camera (or other tridimensional relative position sensor). Hence, it is important to obtain the discrete-time version of the dynamic system under analysis.

2.3.1 Discrete dynamics

Denoting the synchronized sampling period of both sensors as T_s, the discrete time steps can be expressed as $t_k = kT_s + t_0$, where $k \in \mathbb{N}_0$ and t_0 is the initial time. Thus, the discretized system is characterized by the state $\mathbf{x}_k := \mathbf{x}(t_k)$, the dynamics matrix $\mathbf{A}_k(\mathbf{y}_k) := \mathbf{A}(t_k, y(t_k))$ and the output matrix $\mathbf{C}_k := \mathbf{C}(t_k)$. Finally, the Euler discretization of the system dynamics (2.20), including system disturbance and measurement noise, yields

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{F}_k(\mathbf{y}_k)\mathbf{x}_k + \boldsymbol{\xi}_k \\ \mathbf{y}_{k+1} = \mathbf{H}_{k+1}\mathbf{x}_{k+1} + \boldsymbol{\theta}_{k+1} \end{cases}, \tag{2.61}$$

where $\mathbf{F}_k(\mathbf{y}_k) := \mathbf{I}_{n_x} + T_s\mathbf{A}_k(\mathbf{y}_k)$ and $\mathbf{H}_{k+1} := \mathbf{C}_{k+1}$. The disturbance vector $\boldsymbol{\xi}_k$ and the measurement noise vector $\boldsymbol{\theta}_k$ are both zero-mean, discrete white Gaussian noise, with $\langle \boldsymbol{\xi}_k \boldsymbol{\xi}_k^T \rangle = \boldsymbol{\Xi}_k$ and $\langle \boldsymbol{\theta}_k \boldsymbol{\theta}_k^T \rangle = \boldsymbol{\Theta}_k$.

2.3.2 Prediction Step

System (2.61) does not include the unobserved landmarks, which must be propagated in open-loop using the nonlinear equations in (2.19). Thus, with the full state vector, $\mathbf{x}_{F_k} := \begin{bmatrix} \mathbf{x}_k^T & \mathbf{x}_{U_k}^T \end{bmatrix}$, where $\mathbf{x}_{U_k} := \mathbf{x}_U(t_k)$, the prediction step of the Kalman filter is given by

$$\begin{cases} \hat{\mathbf{x}}_{F_{k+1|k}} = \mathbf{F}_{F_{k|k}}(\mathbf{y}_k, \hat{\mathbf{x}}_{U_{k|k}})\hat{\mathbf{x}}_{F_{k|k}} \\ \boldsymbol{\Sigma}_{F_{k+1|k}} = \mathbf{F}_{F_{k|k}}\boldsymbol{\Sigma}_{F_{k+1|k}}\mathbf{F}_{F_{k|k}}^T + \boldsymbol{\Xi}_{F_k} \end{cases}, \tag{2.62}$$

where $\mathbf{F}_{F_{k|k}}(\mathbf{y}_k, \hat{\mathbf{x}}_{U_{k|k}}) = \mathbf{I}_{n_x} + T_s\mathbf{A}_{F_{k|k}}(\mathbf{y}_k, \hat{\mathbf{x}}_{U_{k|k}})$ and $\boldsymbol{\Xi}_{F_k} = \mathrm{diag}(\boldsymbol{\Xi}_k, \boldsymbol{\Xi}_{U_k})$, with $\mathbf{A}_{F_{k|k}}(\mathbf{y}_k, \hat{\mathbf{x}}_{U_{k|k}}) := \mathbf{A}_F(t_k, \mathbf{y}_k, \hat{\mathbf{x}}_{U_{k|k}})$. This prediction step uses the measurements of the rate-gyros, propagating the state every time a reading is available.

2.3.3 Update Step

The update step is divided in two different stages, landmark association and the Kalman filter update equations. This step occurs every time 2-D colour and depth images are available from the *Microsoft Kinect™*. Then, an implementation of SURF [23] detects features in the 2-D picture of the environment that corresponds to the array with the depth measurements provided. The resulting features are then matched to a pointcloud built with the depth image. This matching returns a set of observed tridimensional landmarks in Cartesian coordinates. The SLAM filter does not know a priori if a landmark from this set is in the current map or if it is the first time it is seen. This is when data association takes place, associating the measured data with the known, existing, landmarks. Wrong associations may have a very negative effect on the estimation, so this is a field where a lot of research effort has been put, yielding various algorithms from the community. The algorithm used, the Joint Compatibility Branch and Bound [24], performs a depth-first search only expanding nodes when the joint associations are

jointly compatible in a probabilistic sense. Note that both the landmark detection and association algorithms may be substituted by others, as they are independent from the filtering technique described in this thesis. The procedures here crudely introduced regarding these algorithms are described with more detail in Chapter 4. The association algorithm provides the innovation vector and its covariance matrix, and also redefines the new sets of visible and non-visible landmarks,

$$\begin{cases} \boldsymbol{\nu}_{k+1} = \mathbf{y}_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1|k} \\ \boldsymbol{\Sigma}_{\mu_{k+1}} = \mathbf{H}_{k+1}\boldsymbol{\Sigma}_{k+1|k}\mathbf{H}_{k+1}^T + \boldsymbol{\Theta}_{k+1} \end{cases}, \tag{2.63}$$

The update equations are standard, and given by

$$\begin{cases} \mathbf{K}_{k+1} = \boldsymbol{\Sigma}_{k+1|k}\mathbf{H}_{k+1}^T\boldsymbol{\Sigma}_{\mu_{k+1}}^{-1} \\ \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\boldsymbol{\nu}_{k+1} \\ \boldsymbol{\Sigma}_{k+1|k+1} = \boldsymbol{\Sigma}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\boldsymbol{\Sigma}_{k+1|k} \end{cases}, \tag{2.64}$$

see [25] and [26].

2.3.4 Loop closing

The loop closing problem is very important in any SLAM algorithm, as it enables the recognition of an already sighted place, allowing the reduction of the uncertainty associated with the landmarks. Note that as the vehicle moves, landmarks will become unobserved, and their uncertainty grows indefinitely. The loop closing algorithm consists on using only a subset of the state landmarks in the association algorithm, namely the more recent or current ones $\mathcal{I}_{cur}$, separating the full state into three subsets, the current, the old, $\mathcal{I}_{old}$, and the ones in between: $\mathcal{I}_{gap}$. This allows the duplication of landmarks when an area is revisited. Then, periodically, the algorithm tries to associate landmarks in $\mathcal{I}_{cur}$ and $\mathcal{I}_{old}$ using an adapted version of the association algorithm. If the number of jointly compatible associations passes a certain predefined threshold, a loop closure takes place.

The loop closure is incorporated in the filter by means of a noise free measurement, an association between landmark $i \in \mathcal{I}_{cur}$ and landmark $j \in \mathcal{I}_{old}$, which yields the output vector $\mathbf{z}_{ij_k} := \hat{\mathbf{m}}_{i_{k|k}} - \hat{\mathbf{m}}_{j_{k|k}} = \mathbf{H}_{ij_k}\hat{\mathbf{x}}_{k|k}$, the measurement $\mathbf{y}_{ij_k} := \mathbf{0}$ and the covariance matrix $\boldsymbol{\Theta}_{ij_k} := \mathbf{0}$. Therefore, the new innovation vector and covariance matrix are given by

$$\begin{cases} \boldsymbol{\nu}_{ij_k} = -\mathbf{H}_{ij_k}\hat{\mathbf{x}}_{k|k} \\ \boldsymbol{\Sigma}_{\mu_{ij_k}} = \mathbf{H}_{ij_k}\boldsymbol{\Sigma}_{k|k}\mathbf{H}_{ij_k}^T. \end{cases} \tag{2.65}$$

Finally, the innovation equations above can be used in a new update step, using equations (2.64). This step constrains the two landmarks to be equal and fully correlated, allowing the removal of one of them from the filter state.

3

Tridimensional Inertial Map and Trajectory Estimation

THE framework of the SLAM filter detailed in Chapter 2 is completely independent of the inertial frame, as every input and state are expressed in the body-fixed frame. Therefore, the localization of the vehicle is trivial, corresponding to the origin of the referred frame. Nevertheless, the usual SLAM algorithms perform the mapping and localization in an inertial frame, and many applications require the inertial map and the trajectory of the vehicle. The sensor-based map is readily available, and, if an inertial estimate is provided, the pose of the vehicle can be estimated by the comparison of these two maps. The problem of computing the transformation that maps the two sets of points is usually called the Procrustes Problem. Its generalization for rotation, translation and scaling has been the subject of works in areas such as computer vision applications, and can be traced back to [2] and [27]. This chapter presents an Online Tridimensional Inertial Map and Trajectory Estimate algorithm, henceforth denoted as the *esTIMATE* algorithm, that solves the extended Procrustes problem and provides a measure of the uncertainty associated with its outputs. The chapter is organized as follows. Section 3.1 introduces the problem and provides a closed-form optimal solution. The proposed algorithm is detailed in Section 3.2. In Section 3.3, the characterization of the uncertainty involved is performed using perturbation theory, and in Section 3.4 final remarks regarding the algorithm are presented.

3.1 Definition and Solution of the Optimization Problem

This section addresses the problem of obtaining an estimate of the inertial map and the pose of the vehicle, by formulating an optimization problem with a solution that corresponds to an estimate of the transformation between the body-fixed frame $\{B\}$ and the inertial frame $\{I\}$, yielding the *esTIMATE* algorithm. An error function is defined and then used to construct a functional for the optimization problem. This problem is shown to have a closed-form solution that is detailed next.

Consider the existence of two landmark sets, $\mathcal{I}_{I_k}$ and $\mathcal{I}_{B_k}$, which contain, respectively, the landmarks expressed in the inertial frame and those in the body-fixed frame. The last set is provided as the output of the SLAM filter that is proposed in Chapter 2. Each landmark ${}^I\hat{\mathbf{p}}_{i_k} \in \mathcal{I}_{I_k}$ corresponds to a landmark $\hat{\mathbf{p}}_{i_k} \in \mathcal{I}_{B_k}$, with $i \in \{1, \ldots, N_M\}$, and that correspondence is expressed by

$$ {}^I\hat{\mathbf{p}}_{i_k} = {}^I\hat{\mathbf{p}}_k + \hat{\mathbf{R}}_k \hat{\mathbf{p}}_{i_k}, \tag{3.1} $$

where the pair $\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right)$, namely the orientation and position of the vehicle in frame $\{I\}$, fully defines the transformation from the body-fixed frame $\{B\}$ to the inertial frame, as it represents the estimated rotation and translation from $\{B\}$ to $\{I\}$. Given the relation between the two sets, it is possible to define the error function

$$ {}^I\mathbf{e}_{i_k} = {}^I\hat{\mathbf{p}}_{i_k} - \hat{\mathbf{R}}_k \hat{\mathbf{p}}_{i_k} - {}^I\hat{\mathbf{p}}_k, \tag{3.2} $$

that represents the error between the inertial landmark estimate $i \in \mathcal{I}_{I_k}$ and its sensor-based homologous, rotated and translated with the estimated transformation. As both sets are known, it remains to obtain the pair $\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right)$. That is the purpose of the optimization problem

$$ \left(\mathbf{R}_k^*, {}^I\mathbf{p}_k^*\right) = \arg \min_{\substack{\hat{\mathbf{R}}_k \in \mathrm{SO}(3) \\ {}^I\hat{\mathbf{p}}_k \in \mathbb{R}^3}} G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right), \tag{3.3} $$

where the functional $G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right)$ is given by

$$ G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right) = \frac{1}{N_T} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left\| {}^I\hat{\mathbf{p}}_{i_k} - \hat{\mathbf{R}}_k \hat{\mathbf{p}}_{i_k} - {}^I\hat{\mathbf{p}}_k \right\|^2 \tag{3.4} $$

and $\sigma_{i_k}^2 > 0$, $i \in \{1, \ldots, N_T\}$, account for the intrinsic uncertainty of each landmark pair. Note that N_T may be different from N_M, as the landmarks used in the algorithm may be only subsets of $\mathcal{I}_{I_k}$ and $\mathcal{I}_{B_k}$. This implementation aspect will be explained later in this chapter. It is a simple matter of computation to show that this cost function may also be described in the matrix form

$$ G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right) = \frac{1}{N_T} \left\| \left(\mathbf{Y}_k - \hat{\mathbf{R}}_k \mathbf{X}_k - {}^I\hat{\mathbf{p}}_k \mathbf{1}^T\right) \boldsymbol{\Sigma}_{I_{e_k}}^{-1/2} \right\|^2, \tag{3.5} $$

where

- $\mathbf{Y}_k = \begin{bmatrix} {}^I\hat{\mathbf{p}}_{1_k} & \cdots & {}^I\hat{\mathbf{p}}_{N_{T_k}} \end{bmatrix}$ and $\mathbf{X}_k = \begin{bmatrix} \hat{\mathbf{p}}_{1_k} & \cdots & \hat{\mathbf{p}}_{N_{T_k}} \end{bmatrix}$, $\mathbf{Y}_k, \mathbf{X}_k \in \mathbb{R}^{3 \times N_T}$, are the concatena-

tion of the landmark vectors expressed in the inertial and body-fixed frames, respectively,

- $\mathbf{1} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T \in \mathbb{R}^{N_T}$ is a vector of ones,
- and the weight matrix $\mathbf{\Sigma}_{I_{e_k}}$ is a diagonal matrix whose entries are the weights $\sigma^2_{1_k}, \dots, \sigma^2_{N_T k}$ that model the landmark uncertainty.

The elements of $\mathbf{\Sigma}_{I_{e_k}}$ are conservatively defined as

$$\begin{aligned} \sigma^2_{i_k} &= \lambda_{max}\left(\mathbf{\Sigma}_{{}^I\mathbf{p}_{i_k}}\right) + \lambda_{max}\left(\mathbf{\Sigma}_{\mathbf{p}_{i_k}}\right) \\ &\geq \lambda_{max}\left(\mathbf{\Sigma}_{{}^I\mathbf{p}_{i_k}} + \mathbf{R}_k \mathbf{\Sigma}_{\mathbf{p}_{i_k}} \mathbf{R}_k^T\right) \end{aligned} \tag{3.6}$$

because the true $\mathbf{\Sigma}_{{}^I\mathbf{e}_k}$ is not known. This weight matrix allows the use of the information regarding the different degrees of uncertainty of each landmark pair in the inertial map and trajectory estimation.

The objective of this section is to find a closed-form solution for the optimization problem (3.3). That objective is accomplished by rewriting the associated functional as a matrix expression and rearranging the terms involved. The derivation here described is based on the work exposed in [2], where a similar problem is posed, although the functional used does not include any weight. Before advancing with the derivation, several statistical quantities must be defined.

Definition 4 (Statistical properties of landmark sets)**.** *The weighted statistical properties of the landmark sets $\mathcal{I}_{B_T}$ and $\mathcal{I}_{I_T}$, respectively the sensor-based and inertial subsets used in esTIMATE, each with N_T landmarks, comprise the following first and second order moments:*

$$\boldsymbol{\mu}_{B_k} := \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \hat{\mathbf{p}}_{i_k} \tag{3.7}$$

$$\sigma^2_{B_k} := \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \left\| \sigma_{i_k}^{-2} \left(\hat{\mathbf{p}}_{i_k} - \boldsymbol{\mu}_{B_k}\right) \right\|^2 \tag{3.8}$$

$${}^I\boldsymbol{\mu}_{I_k} := \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} {}^I\hat{\mathbf{p}}_{i_k} \tag{3.9}$$

$$\sigma^2_{I_k} := \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \left\| \sigma_{i_k}^{-2} \left({}^I\hat{\mathbf{p}}_{i_k} - {}^I\boldsymbol{\mu}_{I_k}\right) \right\|^2 \tag{3.10}$$

and

$$\mathbf{\Sigma}_{IB_k} := \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \left({}^I\hat{\mathbf{p}}_{i_k} - {}^I\boldsymbol{\mu}_{I_k}\right) \left(\hat{\mathbf{p}}_{i_k} - \boldsymbol{\mu}_{B_k}\right)^T \tag{3.11}$$

where $N_{W_k} := \sum_{i=1}^{N_T} \sigma_{i_k}^{-2}$, the centroids $\boldsymbol{\mu}_{B_k}, {}^I\boldsymbol{\mu}_{I_k} \in \mathbb{R}^3$, the variances $\sigma^2_{B_k}, \sigma^2_{I_k} \in \mathbb{R}$, and the covariance $\mathbf{\Sigma}_{IB_k} \in \mathbb{R}^{3\times 3}$.

As with the functional (3.4), the quantities presented in Definition 4 may also be expressed in matrix form. For that purpose, consider the symmetric weight matrix $\mathbf{W}_k \in \mathbb{R}^{N_T \times N_T}$,

$$\mathbf{W}_k := \mathbf{\Sigma}^{-1}_{{}^I\mathbf{e}_k} - \frac{1}{N_{W_k}} \mathbf{\Sigma}^{-1}_{{}^I\mathbf{e}_k} \mathbf{1}\mathbf{1}^T \mathbf{\Sigma}^{-1}_{{}^I\mathbf{e}_k}. \tag{3.12}$$

Using $\mathbf{W}_k$ and recalling the definition of $\mathbf{Y}_k$ and $\mathbf{X}_k$, the new expressions for the centroids ${}^I\boldsymbol{\mu}_{I_k}$ and $\boldsymbol{\mu}_{B_k}$ of the landmark sets $\mathcal{I}_I$ and $\mathcal{I}_B$ and respective second moments can be rewritten as

$$\boldsymbol{\mu}_{B_k} = \frac{1}{N_{W_k}} \mathbf{X}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1} \tag{3.13}$$

$$\sigma_{B_k}^2 = \frac{1}{N_{W_k}} \left\| \mathbf{X}_k \mathbf{W}_k \right\|^2 \tag{3.14}$$

$${}^I\boldsymbol{\mu}_{I_k} = \frac{1}{N_{W_k}} \mathbf{Y}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1} \tag{3.15}$$

$$\sigma_{I_k}^2 = \frac{1}{N_{W_k}} \left\| \mathbf{Y}_k \mathbf{W}_k \right\|^2 \tag{3.16}$$

and

$$\boldsymbol{\Sigma}_{IB_k} = \frac{1}{N_{W_k}} \mathbf{Y}_k \mathbf{W}_k \mathbf{X}_k^T, \tag{3.17}$$

where N_{W_k} may also be written as $N_{W_k} = \mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1}$. The derivation of the closed-form solution may proceed now following steps similar to the ones in [2]. Consider the definition of $\mathbf{W}_k$ in (3.12). Left-multiplication separately by $\mathbf{X}_k$ and $\mathbf{Y}_k$ and right-multiplication of each result by $\boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}$ yields

$$\begin{cases} \mathbf{X}_k \mathbf{W}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k} = \mathbf{X}_k - \dfrac{1}{N_{W_k}} \mathbf{X}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1}\mathbf{1}^T \\ \mathbf{Y}_k \mathbf{W}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k} = \mathbf{Y}_k - \dfrac{1}{N_{W_k}} \mathbf{Y}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1}\mathbf{1}^T \end{cases}. \tag{3.18}$$

Isolating $\mathbf{X}_k$ and $\mathbf{Y}_k$ in the respective equations in (3.18), and substituting the result into the cost function (3.5) yields

$$G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right) = \frac{1}{N} \left\| \mathbf{Y}_k \mathbf{W}_k' - \hat{\mathbf{R}}_k \mathbf{X}_k \mathbf{W}_k' - \mathbf{t}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2} \right\|^2,$$

where $\mathbf{W}_k' = \mathbf{W}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{1/2}$ and

$$\mathbf{t} = {}^I\hat{\mathbf{p}}_k - \frac{1}{N_{W_k}} \left(\mathbf{Y}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1} - \hat{\mathbf{R}}_k \mathbf{X}_k \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1} \right). \tag{3.19}$$

Recall the definition of the matrix Frobenius norm, $\|\mathbf{A}\|^2 = \operatorname{tr}\left(\mathbf{A}\mathbf{A}^T\right)$. The expansion of the norm that defines the functional gives

$$\begin{aligned} G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right) = \frac{1}{N_T} \left\| \mathbf{Y}_k \mathbf{W}_k' - \hat{\mathbf{R}}_k \mathbf{X}_k \mathbf{W}_k' \right\|^2 + \frac{1}{N_T} \left\| \mathbf{t}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2} \right\|^2 \\ - \frac{1}{N_T} \operatorname{tr}\left(\mathbf{W'}_k^T \left(\mathbf{Y}_k^T - \mathbf{X}_k^T \hat{\mathbf{R}}_k{}^T \right) \mathbf{t}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2} \right). \end{aligned} \tag{3.20}$$

The last term of the previous expression can be easily shown to be null. Recover the definition of $\mathbf{W}_k'$, and substitute it in (3.20). Then, isolating the last term yields

$$\operatorname{tr}\left(\boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{1/2} \mathbf{W}_k^T \left(\mathbf{Y}_k^T - \mathbf{X}_k^T \hat{\mathbf{R}}_k{}^T \right) \mathbf{t}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2} \right).$$

It is a simple matter of computation to show that this is the same as

$$\operatorname{tr}\left(\mathbf{1}^T \mathbf{W}_k^T \left(\mathbf{Y}_k^T - \mathbf{X}_k^T \hat{\mathbf{R}}_k{}^T \right) \mathbf{t} \right),$$

which, using the definition of $\mathbf{W}_k$ in (3.12), becomes

$$\operatorname{tr}\left(\left(\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} - \frac{1}{N_{W_k}} \mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \right) \left(\mathbf{Y}_k^T - \mathbf{X}_k^T \hat{\mathbf{R}}_k{}^T \right) \mathbf{t} \right).$$

Finally, and noting that $\mathbf{N}_W = \mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1}$, this yields

$$\operatorname{tr}\left(\left(\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} - \frac{N_{W_k}}{N_{W_k}} \mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1}\right)\left(\mathbf{Y}_k^T - \mathbf{X}_k^T \hat{\mathbf{R}}_k{}^T\right)\mathbf{t}\right) = 0.$$

With this simple step, the functional to be minimized is now composed of a sum of two matrix norms, and each one may be minimized separately. Ideally, both terms would be made zero, but the first can only be zero if the correspondence between sets is noiseless. The second norm can be further simplified by noting that $\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2} \in \mathbb{R}^{1\times N}$ and that $\|\mathbf{a}\mathbf{b}^T\|^2 = \operatorname{tr}\left(\mathbf{a}\mathbf{b}^T\left(\mathbf{a}\mathbf{b}^T\right)^T\right) = \operatorname{tr}\left(\mathbf{a}\mathbf{b}^T\mathbf{b}\mathbf{a}^T\right) = \|\mathbf{b}\|^2\|\mathbf{a}\|^2$.

$$\left\|\mathbf{t}\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2}\right\|^2 = \|\mathbf{t}\|^2 \left\|\mathbf{1}^T \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2}\right\|^2 = 0$$
$$\Leftrightarrow$$
$$\|\mathbf{t}\|^2 = 0$$
$$\Leftrightarrow$$
$$\mathbf{t} = \mathbf{0}$$

Recall that $\boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1/2}$ is diagonal and that its elements are all positive, and thus the step of removing its norm from the expression is valid. Given that $\mathbf{t} = \mathbf{0}$, the optimal translation vector may now be found from the definition of $\mathbf{t}$ in (3.19),

$${}^I\mathbf{p}_k^* = \frac{1}{N_{W_k}} \left(\mathbf{Y}_k - \mathbf{R}_k^* \mathbf{X}_k\right) \boldsymbol{\Sigma}_{{}^I\mathbf{e}_k}^{-1} \mathbf{1} = {}^I\boldsymbol{\mu}_{I_k} - \mathbf{R}_k^* \boldsymbol{\mu}_{B_k}. \tag{3.21}$$

Taking a closer look at (3.21) as the solution for the optimization problem (3.3), it is possible to see that the optimal translation is the vector that translates the centroid of the body-fixed landmarks expressed in the inertial frame to the centroid of the inertial landmarks, which makes perfect sense.

Finally, it remains to find the optimal rotation from the body-fixed frame onto the inertial frame. As the second norm in (3.20) is optimally zero, the new functional is then

$$G\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right) = \frac{1}{N_T} \left\|\mathbf{Y}_k \mathbf{W}_k' - \hat{\mathbf{R}}_k \mathbf{X}_k \mathbf{W}_k'\right\|^2. \tag{3.22}$$

The following result addresses the problem of finding the optimal transformation between two point sets with the same centroid. Being centered, this transformation takes the form of the rotation that maps the two sets. Note that both $\mathbf{Y}_k \mathbf{W}_k'$ and $\hat{\mathbf{R}}_k \mathbf{X}_k \mathbf{W}_k'$ have their centroid in the origin of the inertial frame.

Lemma 2 (Orthogonal Procrustes problem [2, Lemma, Section II]). *Let $\mathbf{Y}'$ and $\mathbf{X}'$ be $m \times n$ matrices, and $\mathbf{R} \in SO(m)$ an $m \times m$ rotation matrix, and $\mathbf{U}\mathbf{D}\mathbf{V}^T$ a singular value decomposition of $\mathbf{Y}'\mathbf{X}'^T$, with $\mathbf{U}\mathbf{U}^T = \mathbf{V}\mathbf{V}^T = \mathbf{I}$ and $\mathbf{D} = \operatorname{diag}(d_i)$, $d_1 \geq \cdots \geq d_m \geq 0$. Then the minimum value of $\|\mathbf{Y}' - \mathbf{R}\mathbf{X}'\|^2$ with respect to $\mathbf{R}$ is*

$$\min_{\mathbf{R}} \|\mathbf{Y}' - \mathbf{R}\mathbf{X}'\|^2 = \|\mathbf{Y}\|^2 + \|\mathbf{X}'\|^2 - 2\operatorname{tr}(\mathbf{D}\mathbf{D}') \tag{3.23}$$

where

$$\mathbf{D}' = \begin{cases} \mathbf{I} & \text{if } \det(\mathbf{Y}'\mathbf{X}'^T) \geq 0 \\ \operatorname{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(\mathbf{Y}'\mathbf{X}'^T) \leq 0. \end{cases} \tag{3.24}$$

When $\operatorname{rank}(\mathbf{Y}'\mathbf{X}'^T) \geq m-1$*, the optimum rotation matrix* $\mathbf{R}$ *which achieves the above minimum value is uniquely determined,*

$$\mathbf{R}^* = \mathbf{U}\mathbf{D}'\mathbf{V}^T, \tag{3.25}$$

where $\mathbf{D}'$ *must be chosen as*

$$\mathbf{D}' = \begin{cases} \mathbf{I} & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = 1 \\ \operatorname{diag}(1, 1, \cdots, 1, -1) & \text{if } \det(\mathbf{U})\det(\mathbf{V}) = -1, \end{cases} \tag{3.26}$$

when $\det(\mathbf{Y}'\mathbf{X}'^T) = 0$*, i.e.,* $\operatorname{rank}(\mathbf{Y}'\mathbf{X}'^T) = m-1$

Proof. The proof of this lemma can be found in [2, Section II]. □

Lemma 2 applies directly to the problem of minimizing the error function (3.22), if it is established that $\mathbf{Y}' := \mathbf{Y}_k\mathbf{W}'_k$ and $\mathbf{X}' := \mathbf{X}_k\mathbf{W}'_k$. Thus, and using the referred lemma, the optimal rotation matrix is given by

$$\mathbf{R}_k^* = \mathbf{U}^*\mathbf{V}^{*T}, \tag{3.27}$$

where $\mathbf{U}^* := \mathbf{U}\operatorname{diag}(1, 1, |\mathbf{U}|)$ and $\mathbf{V}^* := \mathbf{V}\operatorname{diag}(1, 1, |\mathbf{V}|)$. Note that

$$\mathbf{U}^*\mathbf{V}^{*T} = \mathbf{U}\operatorname{diag}(1, 1, |\mathbf{U}|)\operatorname{diag}(1, 1, |\mathbf{V}|)\mathbf{V}^T,$$

and that $\operatorname{diag}(1, 1, |\mathbf{U}|)\operatorname{diag}(1, 1, |\mathbf{V}|)$ corresponds to $\mathbf{D}'$, as defined in (3.26). Furthermore, the matrix $\mathbf{Y}'\mathbf{X}'^T$, whose singular value decomposition yields $\mathbf{U}$ and $\mathbf{V}$, is given by

$$\mathbf{Y}'\mathbf{X}'^T = \mathbf{Y}_k\mathbf{W}'_k\mathbf{W}'^T_k\mathbf{X}_k^T = \mathbf{Y}_k\mathbf{W}_k\mathbf{\Sigma}_{^I\mathbf{e}_k}\mathbf{W}_k\mathbf{X}_k^T = \mathbf{Y}_k\mathbf{W}_k\mathbf{X}_k^T = N_{W_k}\mathbf{\Sigma}_{IB_k}, \tag{3.28}$$

noting that $\mathbf{W}_k\mathbf{\Sigma}_{^I\mathbf{e}_k}\mathbf{W}_k$ is trivially given by $\mathbf{W}_k$. Finally, $\mathbf{U}$ and $\mathbf{V}$ are obtained using

$$\mathbf{U}\mathbf{D}\mathbf{V}^T = \operatorname{svd}(\mathbf{B}_k^T), \tag{3.29}$$

with $\mathbf{B}_k := \mathbf{X}_k\mathbf{W}_k\mathbf{Y}_k^T$.

3.2 The *esTIMATE* algorithm

The correspondence in (3.1), as the landmarks in $\mathcal{I}_I$ are static, is also valid between ${}^I\hat{\mathbf{p}}_{i_{k+1}}$ and $\hat{\mathbf{p}}_{i_k}$, $i \in \{1, \ldots, N_M\}$. This step is of the utmost importance in the design of the algorithm, because it is only possible to compute the pair $\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right)$ having, a priori, both landmark sets. However, the computation of ${}^I\hat{\mathbf{p}}_{i_k}$ requires the transformation between frames to be available. This algebraic loop is

averted by using the update equation

$$ {}^{I}\hat{\mathbf{p}}_{i_{k+1}} = {}^{I}\hat{\mathbf{p}}_{k} + \hat{\mathbf{R}}_{k}\hat{\mathbf{p}}_{i_{k}}, \tag{3.30} $$

following the steps in Fig. 3.1. The set $\mathcal{I}_{B_k}$, which comes from the Sensor-based SLAM filter developed

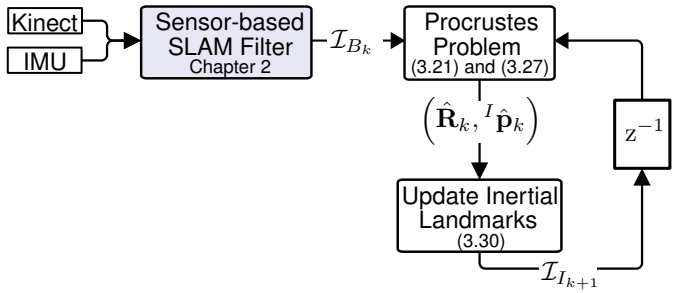


Figure 3.1: The *esTIMATE* algorithm.

in Chapter 2 is compared with the set $\mathcal{I}_{I_k}$ computed in the previous iteration of the *esTIMATE* algorithm, thus yielding the estimated transformation between frames, using (3.21) and (3.27). The inertial map is then computed for all the landmarks in $\mathcal{I}_{B_k}$. Note that the solution of the Procrustes problem uses only a subset containing the most recently visible landmarks in $\mathcal{I}_{B_k}$, provided that the dimension of the resulting set $\mathcal{I}_{B_{T_k}}$ is greater, if possible, than a predefined threshold. Reducing the number of landmarks makes the algorithm computationally more efficient, and the threshold is imposed to guarantee the numerical robustness and statistical consistency of the algorithm.

As to the uncertainty of the estimated pair $\left(\mathbf{R}_k^*, {}^{I}\mathbf{p}_k^*\right)$ and of the inertial map, it is the subject of the next section.

3.3 Uncertainty Description of the Inertial Estimates

The work exposed in this chapter, including the estimates for the vehicle pose, given by (3.21) and (3.27), and the update equation (3.30), allows the real time computation of the vehicle trajectory and of the inertial map. That, along with the SLAM Kalman filter previously described, enables the implementation of an integrated SLAM algorithm, that provides the vehicle pose and velocity in both reference frames, as well as the environment map. However, the tridimensional inertial map and trajectory estimation algorithm here described, apart from having uncertainties involved, demands the knowledge of the uncertainty of both the inertial and sensor-based landmark estimates. The latter is directly provided by the Kalman filter, but the former is yet to be described. These statistical properties have been the subject of study in works such as [28], [29] and [30]. In these works, perturbation theory was employed. However, some rather limiting assumptions were taken, namely, the absence of weighting in the functional $G\left(\hat{\mathbf{R}}_k, {}^{I}\hat{\mathbf{p}}_k\right)$, the use of small rotations, and the same covariance for each landmark. The scope of this section is to provide approximate uncertainty descriptions of the estimates provided by this algorithm. That is done by building on the referred work and considering arbitrary rotations and translations, individual weights, and a total covariance matrix for the whole inertial

map. A similar approach for the bidimensional case is hinted in [19].

The error models of the known sensor-based and inertial variables are defined as follows

$$\hat{\mathbf{p}}_{i_k} = \hat{\mathbf{p}}_{i_k}^{(0)} + \epsilon\, \hat{\mathbf{p}}_{i_k}^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.31a}$$

$${}^I\hat{\mathbf{p}}_{i_k} = {}^I\hat{\mathbf{p}}_{i_k}^{(0)} + \epsilon\, {}^I\hat{\mathbf{p}}_{i_k}^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.31b}$$

$${}^I\hat{\mathbf{p}}_{k} = {}^I\hat{\mathbf{p}}_{k}^{(0)} + \epsilon\, {}^I\hat{\mathbf{p}}_{k}^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.31c}$$

$${}^I\hat{\mathbf{p}}_{i_{k+1}} = {}^I\hat{\mathbf{p}}_{i_{k+1}}^{(0)} + \epsilon\, {}^I\hat{\mathbf{p}}_{i_{k+1}}^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.31d}$$

where

- ϵ is the smallness parameter and the notation $\mathcal{O}(\epsilon^n)$ stands for all the terms of order n or superior,
- the zero order terms are deterministic, i.e, $\langle \mathbf{a}^{(0)} \rangle = \mathbf{a}^{(0)}$,
- and the first order terms, $\mathbf{a}_i^{(1)}$, are assumed to be Gaussian distributed with zero mean and covariance matrices defined by $\mathbf{\Sigma}_{a_{ij_k}} := \langle \mathbf{a}_i^{(1)} {\mathbf{a}_j^{(1)}}^T \rangle$.

The rotation matrix from $\{B\}$ to $\{I\}$ is assumed to have the special structure

$$\hat{\mathbf{R}}_k = \exp\left(\mathbf{S}\left[\mathbf{\Omega}_k\right]\right) \hat{\mathbf{R}}_k^{(0)} = \left[\mathbf{I} + \epsilon\, \mathbf{S}\left[\mathbf{\Omega}_k\right] + \mathcal{O}(\epsilon^2)\right] \hat{\mathbf{R}}_k^{(0)}, \tag{3.32}$$

where $\mathbf{\Omega}_k \in \mathbb{R}^3$ denotes the rotation error and $\mathbf{R}_k^{(0)}$ the true rotation matrix. With all the error models defined, the next step is to compute the expressions that define ${}^I\hat{\mathbf{p}}_{i_{k+1}}^{(0)}$, ${}^I\hat{\mathbf{p}}_{i_{k+1}}^{(1)}$, ${}^I\hat{\mathbf{p}}_{k}^{(0)}$, ${}^I\hat{\mathbf{p}}_{k}^{(1)}$, and $\mathbf{\Omega}_k$ as well as their expected values and covariance matrices.

3.3.1 Rotation uncertainty

The rotation matrix obtained by the optimization process described before belongs to the special orthogonal group $\mathrm{SO}(3)$[1], which yields the two equations $\mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}$ and $|\mathbf{R}_k| = 1$. It is necessary to prove then that $\hat{\mathbf{R}}_k^{(0)}$ also belongs to $\mathrm{SO}(3)$, a necessity for it to be considered the true rotation. Consider the expansion of the two equations that describe the orthogonal group restrictions, where the higher order terms were neglected:

$$\begin{cases} \left[\hat{\mathbf{R}}_k^{(0)} + \epsilon\, \mathbf{S}\left[\mathbf{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)}\right]^T \left[\hat{\mathbf{R}}_k^{(0)} + \epsilon\, \mathbf{S}\left[\mathbf{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)}\right] = \mathbf{I} \\ \det\left(\hat{\mathbf{R}}_k^{(0)} + \epsilon\, \mathbf{S}\left[\mathbf{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)}\right) = 1 \end{cases}$$

$$\Leftrightarrow$$

$$\begin{cases} {\hat{\mathbf{R}}_k^{(0)}}^T \hat{\mathbf{R}}_k^{(0)} + \epsilon\, \left({\hat{\mathbf{R}}_k^{(0)}}^T \mathbf{S}\left[\mathbf{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)} + {\hat{\mathbf{R}}_k^{(0)}}^T \mathbf{S}^T\left[\mathbf{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)}\right) + O(\epsilon^2) = \mathbf{I} \\ \det\left(\hat{\mathbf{R}}_k^{(0)}\right) + \epsilon\, \det\left(\mathbf{S}\left[\mathbf{\Omega}_k\right]\right) \det\left(\hat{\mathbf{R}}_k^{(0)}\right) = 1 \end{cases}$$

[1]The optimization problem treated in [2], and from which spawns Lemma 2, does indeed include the constraints of the rotation group.

Recalling that $\mathbf{S}\left[\mathbf{\Omega}_k\right] = -\mathbf{S}^T\left[\mathbf{\Omega}_k\right]$ is skew-symmetric and that the determinant of any odd-order skew-symmetric matrix is zero, this system of equations is reduced to

$$\begin{cases} \hat{\mathbf{R}}_k^{(0)^T}\hat{\mathbf{R}}_k^{(0)} = \mathbf{I} \\ \det\left(\hat{\mathbf{R}}_k^{(0)}\right) = 1 \end{cases},$$

thus leading to $\hat{\mathbf{R}}_k^{(0)} \in \mathrm{SO}(3)$, which confirms that this is the true rotation. Having found the properties of the deterministic term in $\mathbf{R}_k$, it remains to compute the associated uncertainty, which is assumed in (3.32) to be related to the rotation error $\mathbf{\Omega}_k$. Thus, the next steps describe the computation of this rotation error and its statistical properties, starting with some properties associated with the closed-form solution of the optimization problem (3.3). Consider the matrix that is used to compute the estimated rotation $\mathbf{R}_k^*$, $\mathbf{B}_k$. This matrix can be described in terms of its error model, using that of matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$,

$$\mathbf{X}_k = \mathbf{X}_k^{(0)} + \epsilon\,\mathbf{X}_k^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.33}$$

and

$$\mathbf{Y}_k = \mathbf{Y}_k^{(0)} + \epsilon\,\mathbf{Y}_k^{(1)} + \mathcal{O}(\epsilon^2) \tag{3.34}$$

respectively, which are a generalization of (3.31a) and (3.31b). Discarding the higher order terms in the previous expressions, it is simple to obtain the error model for $\mathbf{B}_k$

$$\mathbf{B}_k = \mathbf{B}_k^{(0)} + \epsilon\,\mathbf{B}_k^{(1)}, \tag{3.35}$$

with $\mathbf{B}_k^{(0)} = \mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{Y}_k^{(0)^T}$, and $\mathbf{B}_k^{(1)} = \mathbf{X}_k^{(1)}\mathbf{W}_k\mathbf{Y}_k^{(0)^T} + \mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{Y}_k^{(1)^T}$. From the proof of Lemma 2 found in [2, Section II], it is known that the matrix $\mathbf{B}_k\mathbf{R}_k$ is symmetrical, and thus, using the error model (3.32) and (3.35) the following expression may be derived

$$\mathrm{skew}\left(\mathbf{B}_k\mathbf{R}_k\right) = \mathrm{skew}\left(\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}\right) + \epsilon\,\mathrm{skew}\left(\mathbf{B}_k^{(1)}\hat{\mathbf{R}}_k^{(0)} + \mathbf{B}_k^{(0)}\mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}_k^{(0)}\right) = 0, \tag{3.36}$$

where the definition of the skew operator is used, $\mathrm{skew}(\mathbf{A}) = \frac{1}{2}\left(\mathbf{A} - \mathbf{A}^T\right)$. This expression is to be employed in the determination of the rotation error, enabling the computation of the uncertainty of the estimated $\hat{\mathbf{R}}_k$. Consider then the fundamental relation between the true values of a landmark in both frames and the vehicle pose

$${}^I\mathbf{p}_{i_k}^{(0)} = \hat{\mathbf{R}}_k^{(0)}\mathbf{p}_{i_k}^{(0)} + {}^I\mathbf{p}_k^{(0)}, \tag{3.37}$$

which may be extended to apply to the full conglomerated matrices $\mathbf{Y}_k^{(0)}$ and $\mathbf{X}_k^{(0)}$,

$$\mathbf{Y}_k^{(0)} = \hat{\mathbf{R}}_k^{(0)}\mathbf{X}_k^{(0)} + {}^I\mathbf{p}_k^{(0)}\mathbf{1}^T. \tag{3.38}$$

This can be used in the computation of $\operatorname{skew}\left(\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}\right)$ by substituting (3.38) in $\mathbf{B}_k^{(0)}$, and expanding the skew operator, which yields

$$\begin{aligned}2\operatorname{skew}\left(\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}\right) =&\mathbf{X}_k^{(0)}\mathbf{W}_k{\mathbf{X}_k^{(0)}}^T\hat{\mathbf{R}}_k^{(0)T}\hat{\mathbf{R}}_k^{(0)} + \mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{1}^I{\mathbf{p}_k^{(0)}}^T\hat{\mathbf{R}}_k^{(0)}\\ &-\hat{\mathbf{R}}_k^{(0)T}\hat{\mathbf{R}}_k^{(0)}\mathbf{X}_k^{(0)}\mathbf{W}_k{\mathbf{X}_k^{(0)}}^T - \hat{\mathbf{R}}_k^{(0)T}{}^I\mathbf{p}_k^{(0)}\mathbf{1}^T\mathbf{W}_k{\mathbf{X}_k^{(0)}}^T\end{aligned}$$

As $\hat{\mathbf{R}}_k^{(0)T}\hat{\mathbf{R}}_k^{(0)} = \mathbf{I}_3$, the terms with $\mathbf{X}_k^{(0)}\mathbf{W}_k{\mathbf{X}_k^{(0)}}^T$ cancel, yielding

$$\operatorname{skew}\left(\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}\right) = \operatorname{skew}\left(\mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{1}^I{\mathbf{p}_k^{(0)}}^T\hat{\mathbf{R}}_k^{(0)}\right). \tag{3.39}$$

It can be seen that the argument of the skew operator on the right-hand side is zero. For that purpose, consider its expression as a sum of terms,

$$\mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{1}^I{\mathbf{p}_k^{(0)}}^T = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_k^{(0)T} - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_k^{(0)T}\right].$$

Isolating $\sigma_{i_k}^{-2}\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_k^{(0)T}$, and noting that $N_{W_k} = \sum_{j=1}^{N_T}\sigma_{j_k}^{-2}$, yields

$$\mathbf{X}_k^{(0)}\mathbf{W}_k\mathbf{1}^I{\mathbf{p}_k^{(0)}}^T = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_k^{(0)T}\left(1 - \frac{\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}}{\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}}\right) = 0. \tag{3.40}$$

thus making $\operatorname{skew}\left(\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}\right) = 0$, as intended. It is now shown that both $\mathbf{B}_k\mathbf{R}_k$ and $\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}$ are symmetric matrices. This makes it possible to, from (3.36), build the following equality,

$$\operatorname{skew}\left(\mathbf{B}_k^{(0)}\mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}_k^{(0)}\right) = -\operatorname{skew}\left(\mathbf{B}_k^{(1)}\hat{\mathbf{R}}_k^{(0)}\right). \tag{3.41}$$

The final objective of this step, as well as of the following, is to compute $\mathbf{\Omega}_k$. For that purpose, the last relation will be rearranged in order to isolate $\mathbf{S}\left[\mathbf{\Omega}_k\right]$. Consider the expansion of (3.41)

$$\mathbf{B}_k^{(0)}\mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}_k^{(0)} - \hat{\mathbf{R}}_k^{(0)T}\mathbf{S}^T\left[\mathbf{\Omega}_k\right]{\mathbf{B}_k^{(0)}}^T = \hat{\mathbf{R}}_k^{(0)T}{\mathbf{B}_k^{(0)}}^T - \mathbf{B}_k^{(1)}\hat{\mathbf{R}}_k^{(0)},$$

which, left multiplied by $\hat{\mathbf{R}}_k^{(0)}$ and right multiplied by $\hat{\mathbf{R}}_k^{(0)T}$, yields

$$\hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(0)}\mathbf{S}\left[\mathbf{\Omega}_k\right] + \mathbf{S}\left[\mathbf{\Omega}_k\right]{\mathbf{B}_k^{(0)}}^T\hat{\mathbf{R}}_k^{(0)T} = -\left(\hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(1)} - {\mathbf{B}_k^{(1)}}^T\hat{\mathbf{R}}_k^{(0)T}\right).$$

For simplicity of notation, this may be expressed as

$$\mathbf{A}_k\mathbf{S}\left[\mathbf{\Omega}_k\right] + \mathbf{S}\left[\mathbf{\Omega}_k\right]\mathbf{A}_k^T = -2\operatorname{skew}\left(\mathbf{C}_k\right), \tag{3.42}$$

where $\mathbf{A}_k := \hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(0)}$ and $\mathbf{C}_k := \hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(1)} = [c_{ij}]$. Computing this expression element by element, and then rearranging the result allows to extract the underlying linear matrix equation

$$\boldsymbol{\mathcal{A}}_k\boldsymbol{\Omega}_k = \mathbf{b}_k, \tag{3.43}$$

where the matrix $\boldsymbol{\mathcal{A}}_k \in \mathbb{R}^{3\times3}$ is defined as $\boldsymbol{\mathcal{A}}_k := \left[\mathrm{tr}(\mathbf{A}_k)\,\mathbf{I}_3 - \mathbf{A}_k^T\right]$, and the vector $\mathbf{b}_k \in \mathbb{R}^3$ is defined as $\mathbf{b}_k := \begin{bmatrix} c_{23}-c_{32} & c_{31}-c_{13} & c_{12}-c_{21}\end{bmatrix}^T$, $c_{ij} \in \mathbb{R}$ being the element on the i-th row and j-th column of $\mathbf{C}_k$. Recall that $\mathbf{B}_k^{(0)}\hat{\mathbf{R}}_k^{(0)}$ is a symmetric matrix. It is easy to see that $\hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(0)}$ is also symmetric, i.e., $\mathbf{A}_k = \mathbf{A}_k^T$, which allows to write

$$\boldsymbol{\mathcal{A}}_k = \left[\mathrm{tr}(\mathbf{A}_k)\,\mathbf{I}_3 - \mathbf{A}_k\right]. \tag{3.44}$$

Note that from the linear equation now derived it is straightforward to obtain $\boldsymbol{\Omega}_k$, as long as $\boldsymbol{\mathcal{A}}_k$ is invertible. The next step in finding $\boldsymbol{\Omega}_k$ is then to unveil the conditions in which $\boldsymbol{\mathcal{A}}_k$ is invertible, if any. For that purpose, rewriting it as a sum of terms involving the landmarks (both inertial and sensor-based) provides a better insight on its properties. Consider the matrices $\mathbf{B}_k^{(0)}$ and $\mathbf{A}_k$ expressed as summations,

$$\mathbf{B}_k^{(0)} = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\hat{\mathbf{p}}_{i_k}^{(0)\,I}\hat{\mathbf{p}}_{i_k}^{(0)^T} - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{p}}_{i_k}^{(0)\,I}\hat{\mathbf{p}}_{j_k}^{(0)^T}\right], \tag{3.45}$$

and

$$\mathbf{A}_k = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\left[\hat{\mathbf{p}}_{i_k}^{(0)\,I}\hat{\mathbf{p}}_{i_k}^{(0)^T} - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{p}}_{i_k}^{(0)\,I}\hat{\mathbf{p}}_{j_k}^{(0)^T}\right], \tag{3.46}$$

respectively. Using (3.37), this last expression may be expanded to

$$\mathbf{A}_k = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)} + {}^I\hat{\mathbf{p}}_k^{(0)}\right)^T - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)} + {}^I\hat{\mathbf{p}}_k^{(0)}\right)^T\right].$$

Careful observation of this expression shows that the term ${}^I\hat{\mathbf{p}}_k^{(0)^T}$, which doesn't depend on i nor j, appears on both members of the first summation and thus can be isolated, yielding

$$\begin{aligned}\mathbf{A}_k = &\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)^T - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right)^T\right]\\ &+\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)\,I}\hat{\mathbf{p}}_k^{(0)^T}\left[1 - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\right].\end{aligned}$$

Recall that $N_{W_k} = \sum_{j=1}^{N_T}\sigma_{j_k}^{-2}$. Hence, the second summation is zero and $\mathbf{A}_k$ is given by

$$\mathbf{A}_k = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)^T - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right)^T\right] \tag{3.47}$$

Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be a square matrix composed by the multiplication of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, i.e., $\mathbf{M} = \mathbf{a}\mathbf{b}^T$. Then the trace of matrix $\mathbf{M}$ is given by $\mathrm{tr}(\mathbf{M}) = \mathbf{a}^T\mathbf{b}$. Applying this principle, the computation of $\boldsymbol{\mathcal{A}}_k$ using (3.47) is straightforward, yielding

$$\begin{aligned}\boldsymbol{\mathcal{A}}_k = &\sum_{i=1}^{N_T} \sigma_{i_k}^{-2}\left[\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)^T \hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\,\mathbf{I}_3 - \hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)^T\right] \\ &- \frac{1}{N_{W_k}}\sum_{i,j=1}^{N_T}\sigma_{i_k}^{-2}\sigma_{j_k}^{-2}\left[\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)^T \hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\,\mathbf{I}_3 - \hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right)^T\right].\end{aligned}$$

This rather long expression can be easily simplified by means of the identity $\mathbf{a}^T\mathbf{b}\mathbf{I}_3 - \mathbf{a}\mathbf{b}^T = \mathbf{S}^T[\mathbf{b}]\,\mathbf{S}[\mathbf{a}]$. This yields

$$\boldsymbol{\mathcal{A}}_k = \sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)} - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right]\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]. \tag{3.48}$$

This form of expressing $\boldsymbol{\mathcal{A}}_k$ allows a better insight on its properties than the original (3.44), as the next steps show. To find the conditions in which $\boldsymbol{\mathcal{A}}_k$ is invertible, consider

$$\mathbf{u}^T\boldsymbol{\mathcal{A}}_k\mathbf{u} = 0, \quad \mathbf{u} \in \mathbb{R}^3. \tag{3.49}$$

If $\boldsymbol{\mathcal{A}}_k$ is non-singular, the equation has only the trivial solution $\mathbf{u} = \mathbf{0}$. Thus, it suffices to find the general solutions, which shall be the situations to avert when trying to compute $\boldsymbol{\Omega}_k$. Rearranging (3.49), it is simple to obtain

$$\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)} - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right]\mathbf{u}\right]^T\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\mathbf{u}\right] = 0,$$

which may be separated using the distributive property of the cross product, yielding

$$\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\mathbf{u}\right]^T\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\mathbf{u}\right] - \frac{1}{N_{W_k}}\sum_{i,j=1}^{N_T}\sigma_{i_k}^{-2}\sigma_{j_k}^{-2}\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right]\mathbf{u}\right]^T\left[\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\mathbf{u}\right] = 0.$$

Recall that $\mathbf{S}[\mathbf{a}]\,\mathbf{b} = \mathbf{a}\times\mathbf{b}$ and that $\mathbf{a}^T\mathbf{a} = \mathbf{a}\cdot\mathbf{a} = \|\mathbf{a}\|^2$. This allows to write

$$\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left\|\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)\times\mathbf{u}\right\|^2 - \frac{1}{N_{W_k}}\sum_{i=1}^{N_T}\sum_{j=1}^{N_T}\sigma_{i_k}^{-2}\sigma_{j_k}^{-2}\left[\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right)\times\mathbf{u}\right]\cdot\left[\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)\times\mathbf{u}\right] = 0,$$

which may be further simplified isolating the two summations on the right, giving

$$\sum_{i=1}^{N_T}\left\|\sigma_{i_k}^{-1}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)\times\mathbf{u}\right\|^2 - \frac{1}{N_{W_k}}\left[\sum_{i=1}^{N_T}\sigma_{i_k}^{-2}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right)\times\mathbf{u}\right]\cdot\left[\sum_{j=1}^{N_T}\sigma_{i_k}^{-2}\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right)\times\mathbf{u}\right] = 0.$$

The final step is to recognize that both members involve norms. On the left, there is a sum of norms, on the right the norm of a sum is present. Then, the final expression is obtained, yielding

$$\sum_{i=1}^{N_T} \left\| \sigma_{i_k}^{-1} \left(\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \right) \times \mathbf{u} \right\|^2 - \frac{1}{N_{W_k}} \left\| \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left(\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \right) \times \mathbf{u} \right\|^2 = 0. \tag{3.50}$$

The following technical lemma is of great use in the analysis of this expression.

Lemma 3 (Weighted triangle inequality)**.** *Let V be an Euclidean normed vector space of dimension n, $\mathbf{a}_i \in V$, and $k_i \in \mathbb{R}$, with $i = 1, \ldots, N$. Then,*

$$\sum_{j=1}^{N} k_j^2 \sum_{i=1}^{N} \|k_i \mathbf{a}_i\|^2 \geq \left\| \sum_{i=1}^{N} k_i^2 \mathbf{a}_i \right\|^2, \tag{3.51}$$

where the equality only applies if $\mathbf{a}_i = \mathbf{a}_j \; \forall i, j = 1, \ldots, N$.

Proof. Consider the case $N = 2$. The triangle inequality, in which equality is only valid if $\mathbf{a}_1 = \alpha \, \mathbf{a}_2$, for any $\alpha > 0$, or $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{0}$, states that

$$\left\| k_1^2 \mathbf{a}_1 + k_2^2 \mathbf{a}_2 \right\|^2 \leq k_1^4 \|\mathbf{a}_1\|^2 + k_2^4 \|\mathbf{a}_2\|^2 + 2 k_1^2 k_2^2 \|\mathbf{a}_1\| \|\mathbf{a}_2\|.$$

Its expansion to $N > 2$ is trivially[2] given by

$$\left\| \sum_{i=1}^{N} k_i^2 \mathbf{a}_i \right\|^2 \leq \sum_{i=1}^{N} \sum_{j=1}^{N} k_i^2 k_j^2 \|\mathbf{a}_i\| \|\mathbf{a}_j\|. \tag{3.52}$$

Consider Young's inequality, which states that

$$ab \leq a^p/p + b^q/q, \tag{3.53}$$

with the equality valid only if $a^p = b^q$. Then, using (3.53) in (3.52) yields

$$\begin{aligned} \left\| \sum_{i=1}^{N} k_i^2 \mathbf{a}_i \right\|^2 &\leq \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} k_i^2 k_j^2 (\|\mathbf{a}_i\|^2 + \|\mathbf{a}_j\|^2) \\ &\leq \sum_{i=1}^{N} \sum_{j=1}^{N} k_i^2 k_j^2 \|\mathbf{a}_i\|^2 \\ &\leq \sum_{j=1}^{N} k_j^2 \sum_{i=1}^{N} k_i^2 \|\mathbf{a}_i\|^2, \end{aligned}$$

which concludes the proof of the inequality part of the lemma. The two inequalities used establish the conditions of equality, with Young's inequality being the most restrict:

- The equality in (3.52) is only true if $\mathbf{a}_i = \alpha_i \mathbf{a}_1$ for all $i = 1, \ldots, N$ and any $\alpha_i > 0$;

[2] Consider every two $\mathbf{a}_i$ as conglomerated in another $\mathbf{a}_j$ repeatedly until only two vectors exist, and then expand the right-hand side of the inequality.

- Young's inequality becomes an equality if $\|\mathbf{a}_i\| = \|\mathbf{a}_j\|$ for all $i, j = 1, \ldots, N$.

The conjunction of these two conditions implies that for the equality in (3.51) to apply, all the involved $\mathbf{a}_i$ must be one and the same. □

Let $k_i := \sigma_{i_k}^{-1}$, $\mathbf{a}_i := \left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right) \times \mathbf{u}$ and $\sum_{j=1}^{N} k_j^2 := \sum_{j=1}^{N_T} \sigma_{j_k}^{-2} = N_{W_k}$. Then, it is trivial to see that the present lemma applies, showing that the left-hand member of (3.50) is always greater or equal to zero. Thus, it is possible to conclude that the only solutions for equation (3.50) are

- the trivial solutions $\mathbf{u} = \mathbf{0}$ or $\hat{\mathbf{p}}_{1_k}^{(0)} = \cdots = \hat{\mathbf{p}}_{N_T k}^{(0)} = \mathbf{0}$;
- $\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right) \times \mathbf{u} = \mathbf{0}$, for every possible landmark, i.e., $\mathbf{u}$ parallel to all $\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}$. That implies that all $\hat{\mathbf{p}}_{i_k}^{(0)}$ are collinear, which translates into $\hat{\mathbf{p}}_{i_k}^{(0)} = \alpha_i \hat{\mathbf{p}}_{j_k}^{(0)}$, $\forall i, j \in [1, N_T]$ and $\alpha_i \in \mathbb{R}$;
- and $\left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right) \times \mathbf{u} = \left(\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right) \times \mathbf{u} \neq \mathbf{0}$, $\forall i, j \in [1, N_T]$, which in turn means that $\hat{\mathbf{p}}_{i_k}^{(0)} = \hat{\mathbf{p}}_{j_k}^{(0)}$, $\forall i, j \in [1, N_T]$.

Hence, if at least two landmarks are non collinear, the only solution of (3.49) is $\mathbf{u} = 0$, provided that Assumption 1 is true. This implies that matrix $\boldsymbol{\mathcal{A}}_k$ is invertible, and

$$\boldsymbol{\Omega}_k = \boldsymbol{\mathcal{A}}_k^{-1}\mathbf{b}_k. \tag{3.54}$$

With $\boldsymbol{\Omega}_k$ determined, the next step is to compute its statistical properties, namely its expected value and covariance matrix. The expected value of the rotation error is

$$\langle\boldsymbol{\Omega}_k\rangle = \boldsymbol{\mathcal{A}}_k^{-1}\langle\mathbf{b}_k\rangle = \boldsymbol{\mathcal{A}}_k^{-1}\langle\begin{bmatrix} c_{23} - c_{32} & c_{31} - c_{13} & c_{12} - c_{21}\end{bmatrix}^T\rangle. \tag{3.55}$$

Note that the components of $\mathbf{b}_k$ are linear functions of $\mathbf{C}_k^T - \mathbf{C}_k$ and thus their expected values are connected. Again, for better insight on its properties, rewriting $\mathbf{C}_k = \hat{\mathbf{R}}_k^{(0)}\mathbf{B}_k^{(1)}$ as a summation of terms involving the landmarks is appropriate. Following the same reasoning of (3.45) and (3.46), this yields

$$\mathbf{C}_k = \hat{\mathbf{R}}_k^{(0)} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left[\left(\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_{i_k}^{(1)T} + \hat{\mathbf{p}}_{i_k}^{(1)I}\hat{\mathbf{p}}_{i_k}^{(0)T}\right) - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\left(\hat{\mathbf{p}}_{i_k}^{(0)I}\hat{\mathbf{p}}_{j_k}^{(1)T} + \hat{\mathbf{p}}_{i_k}^{(1)I}\hat{\mathbf{p}}_{j_k}^{(0)T}\right)\right]. \tag{3.56}$$

Recall that both $\langle{}^I\hat{\mathbf{p}}_{i_k}^{(1)}\rangle$ and $\langle\hat{\mathbf{p}}_{i_k}^{(1)}\rangle$ are zero. Then, it is straightforward to see that $\mathbf{C}_k$ and consequently $\mathbf{C}_k^T - \mathbf{C}_k$ have zero mean by using that information in

$$\begin{aligned}\langle\mathbf{C}_k\rangle = \hat{\mathbf{R}}_k^{(0)} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} &\left[\left(\hat{\mathbf{p}}_{i_k}^{(0)}\langle{}^I\hat{\mathbf{p}}_{i_k}^{(1)T}\rangle + \langle\hat{\mathbf{p}}_{i_k}^{(1)}\rangle{}^I\hat{\mathbf{p}}_{i_k}^{(0)T}\right) - \frac{1}{N_{W_k}}\sum_{j=1}^{N_T}\sigma_{j_k}^{-2}\left(\hat{\mathbf{p}}_{i_k}^{(0)}\langle{}^I\hat{\mathbf{p}}_{j_k}^{(1)T}\rangle + \langle\hat{\mathbf{p}}_{i_k}^{(1)}\rangle{}^I\hat{\mathbf{p}}_{j_k}^{(0)T}\right)\right] \\ &= 0.\end{aligned}$$

Hence, if $\langle\mathbf{C}_k\rangle = 0$ then obviously all its elements are zero. That implies that $\langle\mathbf{b}_k\rangle = 0$ which proves that the rotation error has zero mean. Recall now the definition of a covariance matrix given in section

. As $\mathbf{\Omega}_k$ is a zero mean quantity, its covariance matrix is simply given by

$$\mathbf{\Sigma}_{\Omega_k} = \langle \mathbf{\Omega}_k \mathbf{\Omega}_k^T \rangle = \boldsymbol{\mathcal{A}}_k^{-1} \langle \mathbf{b}_k \mathbf{b}_k^T \rangle \boldsymbol{\mathcal{A}}_k^{-1^T}. \tag{3.57}$$

The uncertain component of this covariance matrix, $\langle \mathbf{b}_k \mathbf{b}_k^T \rangle$, is seen to have a special structure when computed elementwise,

$$\langle \mathbf{b}_k \mathbf{b}_k^T \rangle = \langle \begin{bmatrix} (c_{23}-c_{32})^2 & (c_{31}-c_{13})(c_{23}-c_{32}) & (c_{12}-c_{21})(c_{23}-c_{32}) \\ (c_{23}-c_{32})(c_{31}-c_{13}) & (c_{31}-c_{13})^2 & (c_{12}-c_{21})(c_{31}-c_{13}) \\ (c_{23}-c_{32})(c_{12}-c_{21}) & (c_{31}-c_{13})(c_{12}-c_{21}) & (c_{12}-c_{21})^2 \end{bmatrix} \rangle.$$

It is a simple matter of tedious computation to show that this is the same as

$$\langle \mathbf{b}_k \mathbf{b}_k^T \rangle = \langle (\mathbf{C}_k - \mathbf{C}_k^T)^2 \rangle - \frac{1}{2} \operatorname{tr}\langle (\mathbf{C}_k - \mathbf{C}_k^T)^2 \rangle \, \mathbf{I}_3, \tag{3.58}$$

and, therefore, the key element left to determine in order to obtain $\mathbf{\Sigma}_{\Omega_k}$ is

$$\langle (\mathbf{C}_k - \mathbf{C}_k^T)^2 \rangle = \langle \mathbf{C}_k^2 \rangle - \langle \mathbf{C}_k \mathbf{C}_k^T \rangle - \langle \mathbf{C}_k^T \mathbf{C}_k \rangle - \langle \mathbf{C}_k^2 \rangle^T. \tag{3.59}$$

Using (3.56), the three components of $\langle (\mathbf{C}_k - \mathbf{C}_k^T)^2 \rangle$ are given by

$$\langle \mathbf{C}_k \mathbf{C}_k \rangle = \langle \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)} \mathbf{a}_{i_k}^{(1)^T} \mathbf{r}_{j_k}^{(0)} \mathbf{a}_{j_k}^{(1)^T} + \mathbf{r}_{i_k}^{(0)} \mathbf{a}_{i_k}^{(1)^T} \mathbf{r}_{j_k}^{(1)} \mathbf{a}_{j_k}^{(0)^T} + \mathbf{r}_{i_k}^{(1)} \mathbf{a}_{i_k}^{(0)^T} \mathbf{r}_{j_k}^{(0)} \mathbf{a}_{j_k}^{(1)^T} + \mathbf{r}_{i_k}^{(1)} \mathbf{a}_{i_k}^{(0)^T} \mathbf{r}_{j_k}^{(1)} \mathbf{a}_{j_k}^{(0)^T} \right] \rangle,$$

$$\langle \mathbf{C}_k \mathbf{C}_k^T \rangle = \langle \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)} \mathbf{a}_{i_k}^{(1)^T} \mathbf{a}_{j_k}^{(1)} \mathbf{r}_{j_k}^{(0)^T} + \mathbf{r}_{i_k}^{(0)} \mathbf{a}_{i_k}^{(1)^T} \mathbf{a}_{j_k}^{(0)} \mathbf{r}_{j_k}^{(1)^T} + \mathbf{r}_{i_k}^{(1)} \mathbf{a}_{i_k}^{(0)^T} \mathbf{a}_{j_k}^{(1)} \mathbf{r}_{j_k}^{(0)^T} + \mathbf{r}_{i_k}^{(1)} \mathbf{a}_{i_k}^{(0)^T} \mathbf{a}_{j_k}^{(0)} \mathbf{r}_{j_k}^{(1)^T} \right] \rangle,$$

and

$$\langle \mathbf{C}_k^T \mathbf{C}_k \rangle = \langle \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{a}_{i_k}^{(1)} \mathbf{r}_{i_k}^{(0)^T} \mathbf{r}_{j_k}^{(0)} \mathbf{a}_{j_k}^{(1)^T} + \mathbf{a}_{i_k}^{(1)} \mathbf{r}_{i_k}^{(0)^T} \mathbf{r}_{j_k}^{(1)} \mathbf{a}_{j_k}^{(0)^T} + \mathbf{a}_{i_k}^{(0)} \mathbf{r}_{i_k}^{(1)^T} \mathbf{r}_{j_k}^{(0)} \mathbf{a}_{j_k}^{(1)^T} + \mathbf{a}_{i_k}^{(0)} \mathbf{r}_{i_k}^{(1)^T} \mathbf{r}_{j_k}^{(1)} \mathbf{a}_{j_k}^{(0)^T} \right] \rangle,$$

where, for the sake of clarity, $\mathbf{C}_k$ is rewritten as $\mathbf{C}_k = \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)} \mathbf{a}_{i_k}^{(1)^T} + \mathbf{r}_{i_k}^{(1)} \mathbf{a}_{i_k}^{(0)^T} \right]$, with $\mathbf{r}_{i_k}^{(.)} \in \mathbb{R}^3$ being the deterministic or uncertain part, depending on whether the superscript is respectively (0) or (1), of a body-fixed landmark rotated to the inertial frame, $\mathbf{r}_{i_k}^{(.)} := \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(.)}$ and $\mathbf{a}_{i_k}^{(.)} \in \mathbb{R}^3$ the deterministic or uncertain part, with the same relation to the superscripts, of an inertial landmark *decentralized*, i.e., to which is subtracted the coordinates of the landmarks centroid: $\mathbf{a}_{i_k}^{(.)} := {}^I\hat{\mathbf{p}}_{i_k}^{(.)} - \frac{1}{N_{W_k}} \sum_{j=1}^{N_T} \sigma_j^{-2} {}^I\hat{\mathbf{p}}_{j_k}^{(.)}$. In all these expressions $i, j \in \{1, \ldots, N_T\}$. It is assumed that the body landmarks

$\mathbf{r}_{i_k}^{(1)}$ are independent from the inertial landmarks $\mathbf{a}_{j_k}^{(1)}$ as the latter were calculated using information from the previous instant $k-1$. Therefore, these cumbersome expressions are reduced to

$$\langle \mathbf{C}_k \mathbf{C}_k \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)} \mathbf{r}_{j_k}^{(0)^T} \langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle + \langle \mathbf{r}_{i_k}^{(1)} \mathbf{r}_{j_k}^{(1)^T} \rangle \mathbf{a}_{i_k}^{(0)} \mathbf{a}_{j_k}^{(0)^T} \right]$$

$$\langle \mathbf{C}_k \mathbf{C}_k^T \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)} \mathbf{r}_{j_k}^{(0)^T} \langle \mathbf{a}_{i_k}^{(1)^T} \mathbf{a}_{j_k}^{(1)} \rangle + \langle \mathbf{r}_{i_k}^{(1)} \mathbf{r}_{j_k}^{(1)^T} \rangle \mathbf{a}_{i_k}^{(0)^T} \mathbf{a}_{j_k}^{(0)} \right]$$

$$\langle \mathbf{C}_k^T \mathbf{C}_k \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\mathbf{r}_{i_k}^{(0)^T} \mathbf{r}_{j_k}^{(0)} \langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle + \langle \mathbf{r}_{i_k}^{(1)^T} \mathbf{r}_{j_k}^{(1)} \rangle \mathbf{a}_{i_k}^{(0)} \mathbf{a}_{j_k}^{(0)^T} \right],$$

where the identity $\mathbf{a}_1 \mathbf{b}_1^T \mathbf{a}_2 \mathbf{b}_2^T = \mathbf{a}_1 \mathbf{a}_2^T \mathbf{b}_1 \mathbf{b}_2^T$ is used. Further computations, where the symbol $\mathbf{r}_{i_k}^{(.)}$ is dropped, lead to the final expressions,

$$\langle \mathbf{C}_k \mathbf{C}_k \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)^T} \hat{\mathbf{R}}_k^{(0)^T} \langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle + \hat{\mathbf{R}}_k^{(0)} \mathbf{\Sigma}_{p_{i_{j_k}}} \hat{\mathbf{R}}_k^{(0)^T} \mathbf{a}_{i_k}^{(0)} \mathbf{a}_{j_k}^{(0)^T} \right] \tag{3.60a}$$

$$\langle \mathbf{C}_k \mathbf{C}_k^T \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)^T} \hat{\mathbf{R}}_k^{(0)^T} \operatorname{tr} \langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle + \hat{\mathbf{R}}_k^{(0)} \mathbf{\Sigma}_{p_{i_{j_k}}} \hat{\mathbf{R}}_k^{(0)^T} \mathbf{a}_{i_k}^{(0)^T} \mathbf{a}_{j_k}^{(0)} \right] \tag{3.60b}$$

$$\langle \mathbf{C}_k^T \mathbf{C}_k \rangle = \sum_{i=1}^{N_T} \sum_{j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\hat{\mathbf{p}}_{i_k}^{(0)^T} \hat{\mathbf{p}}_{j_k}^{(0)} \langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle + \operatorname{tr}(\mathbf{\Sigma}_{p_{i_{j_k}}}) \mathbf{a}_{i_k}^{(0)} \mathbf{a}_{j_k}^{(0)^T} \right], \tag{3.60c}$$

and

$$\langle \mathbf{a}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)^T} \rangle = \mathbf{\Sigma}_{I p_{i_{j_k}}} - \frac{1}{N_{W_k}} \sum_{r=1}^{N_T} \sigma_{r_k}^{-2} \left(\mathbf{\Sigma}_{I p_{i_{r_k}}} + \mathbf{\Sigma}_{I p_{r_{j_k}}} \right) + \frac{1}{N_{W_k}^2} \sum_{r=1}^{N_T} \sum_{s=1}^{N_T} \sigma_{r_k}^{-2} \sigma_{s_k}^{-2} \mathbf{\Sigma}_{I p_{r_{s_k}}}. \tag{3.61}$$

Note that $\mathbf{\Sigma}_{p_{i_{j_k}}} = \langle \hat{\mathbf{p}}_{i_k}^{(1)} \hat{\mathbf{p}}_{j_k}^{(1)^T} \rangle$ is part of the covariance matrix provided by the Kalman filter,

$$\mathbf{\Sigma}_{F_{k|k}} = \begin{bmatrix} \mathbf{\Sigma}_{x_V} & \mathbf{\Sigma}_{x_V x_{M_k}} \\ \mathbf{\Sigma}_{x_M x_{V_k}} & \mathbf{\Sigma}_{x_{M_k}} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_{x_V} & \mathbf{\Sigma}_{x_V p_{1_k}} & \cdots & \mathbf{\Sigma}_{x_V p_{N_{M_k}}} \\ \mathbf{\Sigma}_{p_1 x_{V_k}} & \mathbf{\Sigma}_{p_{1_k}} & \cdots & \mathbf{\Sigma}_{p_1 N_{M_k}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Sigma}_{p_{N_M} x_{V_k}} & \mathbf{\Sigma}_{p_{N_M 1_k}} & \cdots & \mathbf{\Sigma}_{p_{1_k}} \end{bmatrix}, \tag{3.62}$$

and that $\mathbf{\Sigma}_{I p_{i_{j_k}}}$ is part of the yet to be determined covariance of the set of inertial landmarks, $\mathbf{\Sigma}_{I x_M}$, denoting the cross-covariance between landmark i and landmark j, $i, j \in \mathcal{I}_{I_k}$. This finishes the derivations of the uncertainty related with the computation of the rotation between frames. The work proceeds by determining the uncertainty of the inertial map and trajectory estimates, starting by completing the characterization of the transformation between the sensor-based and inertial frames, that is, the uncertainty of the translation vector, ${}^I\hat{\mathbf{p}}_k$.

3.3.2 Translation uncertainty

The translation between frames is, as explained before, given by the position of the vehicle in a given instant. Therefore, to determine its uncertainty, it is necessary to recall the vehicle position estimate error model presented in (3.31c), as well as its definition in (3.21). The expansion of the latter using the former yields

$$ {}^{I}\mathbf{p}_k = \frac{1}{N_{W_k}} \sum_{i=1}^{N} \sigma_{i_k}^{-2} \left[\left({}^{I}\hat{\mathbf{p}}_{i_k}^{(0)} + \epsilon\, {}^{I}\hat{\mathbf{p}}_{i_k}^{(1)} \right) - \hat{\mathbf{R}}_k^{(0)} \left(\mathbf{I}_3 + \epsilon \mathbf{S}\left[\boldsymbol{\omega}_k\right] \right) \left(\hat{\mathbf{p}}_{i_k}^{(0)} + \epsilon \hat{\mathbf{p}}_{i_k}^{(1)} \right) \right], $$

from which, once more neglecting higher order terms, it is possible to conclude that the deterministic and uncertain parcels are given by

$$ {}^{I}\hat{\mathbf{p}}_k^{(0)} = \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left({}^{I}\hat{\mathbf{p}}_{i_k}^{(0)} - \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \right) \tag{3.63} $$

and

$$ {}^{I}\hat{\mathbf{p}}_k^{(1)} = \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left({}^{I}\hat{\mathbf{p}}_{i_k}^{(1)} - \mathbf{S}\left[\boldsymbol{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} - \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(1)} \right), \tag{3.64} $$

respectively. It is easy to confirm that ${}^{I}\hat{\mathbf{p}}_k^{(1)}$ has zero mean, noting that all other quantities have zero mean,

$$ \langle {}^{I}\hat{\mathbf{p}}_k^{(1)} \rangle = \frac{1}{N_{W_k}} \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \left(\langle {}^{I}\hat{\mathbf{p}}_{i_k}^{(1)} \rangle - \mathbf{S}\left[\langle \boldsymbol{\Omega}_k \rangle\right] \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} - \hat{\mathbf{R}}_k^{(0)} \langle \hat{\mathbf{p}}_{i_k}^{(1)} \rangle \right) = 0. $$

Being so, the covariance matrix of the position estimate $\boldsymbol{\Sigma}_{{}^{I}p_k}$ is simply, $\boldsymbol{\Sigma}_{{}^{I}p_k} = \langle {}^{I}\hat{\mathbf{p}}_k^{(1)} {}^{I}\hat{\mathbf{p}}_k^{(1)^T} \rangle$, which, expanding ${}^{I}\hat{\mathbf{p}}_k^{(1)}$ according to (3.64), yields

$$ \begin{aligned} \boldsymbol{\Sigma}_{{}^{I}p_k} = \frac{1}{N_{W_k}^2} \sum_{i,j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \Big[& \boldsymbol{\Sigma}_{{}^{I}p_{ij_k}} + \hat{\mathbf{R}}_k^{(0)} \boldsymbol{\Sigma}_{p_{ij_k}} \hat{\mathbf{R}}_k^{(0)^T} + \langle \mathbf{S}\left[\boldsymbol{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)^T} \hat{\mathbf{R}}_k^{(0)^T} \mathbf{S}^T\left[\boldsymbol{\Omega}_k\right] \rangle \\ & - \langle {}^{I}\hat{\mathbf{p}}_{i_k}^{(1)} \hat{\mathbf{p}}_{j_k}^{(0)^T} \hat{\mathbf{R}}_k^{(0)^T} \mathbf{S}^T\left[\boldsymbol{\Omega}_k\right] \rangle - \langle \mathbf{S}\left[\boldsymbol{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} {}^{I}\hat{\mathbf{p}}_{j_k}^{(1)^T} \rangle \\ & - \langle \mathbf{S}\left[\boldsymbol{\Omega}_k\right] \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \hat{\mathbf{p}}_{j_k}^{(1)^T} \rangle \hat{\mathbf{R}}_k^{(0)^T} + \hat{\mathbf{R}}_k^{(0)} \langle \hat{\mathbf{p}}_{i_k}^{(1)} \hat{\mathbf{p}}_{j_k}^{(0)^T} \hat{\mathbf{R}}_k^{(0)^T} \mathbf{S}^T\left[\boldsymbol{\Omega}_k\right] \rangle \Big], \end{aligned} $$

where all the cross terms between inertial and sensor-based landmarks were omitted, as these are zero. Consider the cross product property $\mathbf{S}\left[\mathbf{a}\right]\mathbf{b} = -\mathbf{S}\left[\mathbf{b}\right]\mathbf{a}$, which allows to extract the $\boldsymbol{\Omega}_k$ from the skew-symmetric matrix. After some computations, an approximate version of the covariance matrix may be obtained,

$$ \boldsymbol{\Sigma}_{{}^{I}p_k} \approx \frac{1}{N_{W_k}^2} \sum_{i,j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\boldsymbol{\Sigma}_{{}^{I}p_{ij_k}} + \hat{\mathbf{R}}_k^{(0)} \boldsymbol{\Sigma}_{p_{ij_k}} \hat{\mathbf{R}}_k^{(0)^T} + \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \right] \boldsymbol{\Sigma}_{\Omega_k} \mathbf{S}^T \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)} \right] \right], \tag{3.65} $$

where the cross covariance terms between the rotation error and the landmarks (both inertial and sensor-based), i.e., the terms with combinations of $\boldsymbol{\Omega}_k$ and $\hat{\mathbf{p}}_{i_k}^{(1)}$, and $\boldsymbol{\Omega}_k$ and $\hat{\mathbf{p}}_{i_k}^{(1)}$, were neglected.

3.3.3 Inertial map uncertainty

The final step in the process of studying the uncertainty description of the algorithm is computing $\mathbf{\Sigma}_{^I p_{i_j k+1}}$, for all $i, j \in \mathcal{I}_{I_{k+1}}$. The inertial map estimation depends on the translation and rotation estimates, as well as on the sensor-based map of the SLAM filter proposed in Chapter 2. Then, it is predictable that the associated uncertainty will include terms related to the uncertainty of each one of these components, as well as cross covariance terms. Recall that the inertial map estimate is calculated with the update equation in (3.30). Using this expression and the error model in (3.31d), it is possible to determine the variables of the error model, given by, respectively

$$^I\hat{\mathbf{p}}^{(0)}_{i_{k+1}} = {}^I\hat{\mathbf{p}}^{(0)}_{k} + \hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k} \tag{3.66}$$

and

$$^I\hat{\mathbf{p}}^{(1)}_{i_{k+1}} = {}^I\hat{\mathbf{p}}^{(1)}_{k} + \mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k} + \hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(1)}_{i_k}. \tag{3.67}$$

Once again, it is quite simple to confirm that $^I\hat{\mathbf{p}}^{(1)}_{i_{k+1}}$ has zero mean,

$$\langle {}^I\hat{\mathbf{p}}^{(1)}_{i_{k+1}}\rangle = \langle {}^I\hat{\mathbf{p}}^{(1)}_{k}\rangle + \mathbf{S}\left[\langle\mathbf{\Omega}_k\rangle\right]\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k} + \hat{\mathbf{R}}^{(0)}_{k}\langle\hat{\mathbf{p}}^{(1)}_{i_k}\rangle = 0,$$

and the covariance matrix of the position estimate is, approximately,

$$\begin{aligned}
\mathbf{\Sigma}_{^I p_{i_j k+1}} &= \langle {}^I\hat{\mathbf{p}}^{(1)}_{i_{k+1}}\, {}^I\hat{\mathbf{p}}^{(1)^T}_{j_{k+1}}\rangle \\
&\approx \mathbf{\Sigma}_{^I p_k} + \hat{\mathbf{R}}^{(0)}_{k}\mathbf{\Sigma}_{p_{i_j k}}\hat{\mathbf{R}}^{(0)^T}_{k} + \langle\mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k}\hat{\mathbf{p}}^{(0)^T}_{j_k}\hat{\mathbf{R}}^{(0)^T}_{k}\mathbf{S}^T\left[\mathbf{\Omega}_k\right]\rangle \\
&+ \langle\mathbf{S}\left[\mathbf{\Omega}_k\right]\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k}{}^I\hat{\mathbf{p}}^{(1)^T}_{k}\rangle + \langle {}^I\hat{\mathbf{p}}^{(1)}_{k}\hat{\mathbf{p}}^{(0)^T}_{j_k}\hat{\mathbf{R}}^{(0)^T}_{k}\mathbf{S}^T\left[\mathbf{\Omega}_k\right]\rangle \\
&+ \hat{\mathbf{R}}^{(0)}_{k}\langle\hat{\mathbf{p}}^{(1)}_{i_k}{}^I\hat{\mathbf{p}}^{(1)^T}_{k}\rangle + \langle {}^I\hat{\mathbf{p}}^{(1)}_{k}\hat{\mathbf{p}}^{(1)^T}_{j_k}\rangle\hat{\mathbf{R}}^{(0)^T}_{k},
\end{aligned}$$

where all the cross terms between inertial and sensor-based landmarks were omitted, as these are zero and, as before, the cross covariance terms between the rotation error and the landmarks were neglected. After some computation, and also neglecting the cross covariance terms between the vehicle position and the landmarks (both inertial and sensor-based), an approximate version of the covariance matrix may be obtained:

$$\begin{aligned}
\mathbf{\Sigma}_{^I p_{i_j k+1}} \approx \mathbf{\Sigma}_{^I p_k} + \hat{\mathbf{R}}^{(0)}_{k}\mathbf{\Sigma}_{p_{i_j k}}\hat{\mathbf{R}}^{(0)^T}_{k} + \mathbf{S}\left[\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k}\right]\mathbf{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{j_k}\right] \\
+ \mathbf{S}^T\left[\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{i_k}\right]\mathbf{\Sigma}^T_{^I p_k\Omega_k} + \mathbf{\Sigma}_{^I p_k\Omega_k}\mathbf{S}\left[\hat{\mathbf{R}}^{(0)}_{k}\hat{\mathbf{p}}^{(0)}_{j_k}\right].
\end{aligned} \tag{3.68}$$

This confirms the assumption that the uncertainty of the inertial map depends directly on the covariances of the translation, rotation, and sensor-based map estimates. Note that $\mathbf{\Sigma}_{^I p_k\Omega_k} := \langle {}^I\hat{\mathbf{p}}^{(1)}_{k}\mathbf{\Omega}^T_k\rangle$ is the cross covariance between the translation and rotation estimates, given by

$$\mathbf{\Sigma}_{^I p_k\Omega_k} = \langle {}^I\hat{\mathbf{p}}^{(1)}_{k}\mathbf{b}^T_k\rangle\mathcal{A}^{-1^T}_{k} = \langle\begin{bmatrix} {}^I\hat{\mathbf{p}}^{(1)}_{k}(c_{23}-c_{32}) & {}^I\hat{\mathbf{p}}^{(1)}_{k}(c_{31}-c_{13}) & {}^I\hat{\mathbf{p}}^{(1)}_{k}(c_{12}-c_{21})\end{bmatrix}\rangle\mathcal{A}^{-1^T}_{k}.$$

The computation of this cross covariance is achieved by treating each of its columns separately. From (3.56) it is straightforward to see that

$$c_{lm} = \sum_{i=1}^{N_T} \sigma_{i_k}^{-2} \mathbf{1}_l^T \hat{\mathbf{R}}_k^{(0)} \left[\hat{\mathbf{p}}_{i_k}^{(0)} \mathbf{1}_m^T \mathbf{a}_{i_k}^{(1)} + \hat{\mathbf{p}}_{i_k}^{(1)} \mathbf{1}_m^T \mathbf{a}_{i_k}^{(0)} \right],$$

where the vector $\mathbf{1}_m = \begin{bmatrix} \mathbf{0}_{1\times(m-1)} & 1 & \mathbf{0}_{1\times(3-m)} \end{bmatrix}^T \in \mathbb{R}^3$ is a vector whose only non-zero entry is the m-th one. Using (3.64), it is possible to compute approximately $\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle$, which composes each of the columns of $\langle {}^I\hat{\mathbf{p}}_k^{(1)} \mathbf{b}_k^T \rangle$:

$$\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle \approx \frac{1}{N_{W_k}} \sum_{i,j=1}^{N_M} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \left[\langle {}^I\hat{\mathbf{p}}_{i_k}^{(1)} \mathbf{1}_m^T \mathbf{a}_{j_k}^{(1)} \rangle \mathbf{1}_l^T \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)} - \hat{\mathbf{R}}_k^{(0)} \langle \hat{\mathbf{p}}_{i_k}^{(1)} \mathbf{1}_l^T \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(1)} \rangle \mathbf{1}_m^T \mathbf{a}_{j_k}^{(0)} \right],$$

where, as before, the cross covariance terms between landmarks are assumed as zero, and the ones between landmarks and the rotation error were neglected. Rearranging the uncertain terms above and recalling that $\mathbf{a}_{i_k}^{(.)} = {}^I\hat{\mathbf{p}}_{i_k}^{(.)} - \frac{1}{N_{W_k}} \sum_{j=1}^{N_T} {}^I\hat{\mathbf{p}}_{j_k}^{(.)}$, finally yields

$$\begin{aligned} \langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle \approx \frac{1}{N_{W_k}} \sum_{i,j=1}^{N_T} \sigma_{i_k}^{-2} \sigma_{j_k}^{-2} \Big[& \boldsymbol{\Sigma}_{{}^I p_{i_{j_k}}} \mathbf{1}_m \mathbf{1}_l^T \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)} - \frac{1}{N_{W_k}} \sum_{r=1}^{N_T} \sigma_{r_k}^{-2} \boldsymbol{\Sigma}_{{}^I p_{i_{r_k}}} \mathbf{1}_m \mathbf{1}_l^T \hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)} \\ & - \hat{\mathbf{R}}_k^{(0)} \boldsymbol{\Sigma}_{p_{i_{j_k}}} \hat{\mathbf{R}}_k^{(0)} \mathbf{1}_l \mathbf{1}_m^T \mathbf{a}_{j_k}^{(0)} \Big], \end{aligned} \tag{3.69}$$

thus concluding the characterization of the estimates provided by the *esTIMATE* algorithm.

3.4 Final Remarks

Uncertainty update It is important to notice that in this procedure, an inertial landmark is only updated if the associated uncertainty decreases in that iteration. Thus, in each iteration, the candidate inertial landmarks covariance matrix is computed, and then the trace of each $\boldsymbol{\Sigma}_{{}^I p_{i_{k+1}}}$ is compared to its previous value, $\mathrm{tr}(\boldsymbol{\Sigma}_{{}^I p_{i_k}})$. If the uncertainty is raised, then the old covariance is kept and $\boldsymbol{\Sigma}_{{}^I p_{i_{j_{k+1}}}} = \mathbf{0}$ for all $j \neq i$.

Initial conditions The initial estimate is also a relevant point. The usual approach in any localization algorithm is to assume that the vehicle is deterministically at the origin and aligned with the inertial frame. Hence, when the algorithm is started, the position and attitude of the vehicle are fully known, thus allowing the computation of the inertial estimate at $k = 0$ with

$${}^I\hat{\mathbf{p}}_{i_0} = \mathbf{R}_0 \hat{\mathbf{p}}_{i_0} + {}^I\mathbf{p}_0. \tag{3.70}$$

The only non-deterministic quantity here is the sensor-based landmark estimate, and, thus the uncertain part of ${}^I\hat{\mathbf{p}}_{i_0}$ is simply

$${}^I\hat{\mathbf{p}}_{i_0}^{(1)} = \mathbf{R}_0 \hat{\mathbf{p}}_{i_0}^{(1)}. \tag{3.71}$$

The initial covariance matrix that describes the uncertainty associated with the initial estimate is then given by

$$\mathbf{\Sigma}_{I_{p_{i_{j_0}}}} = \langle {}^{I}\hat{\mathbf{p}}_{i_0}^{(1)} {}^{I}\hat{\mathbf{p}}_{j_0}^{(1)^T} \rangle = \mathbf{R}_0 \mathbf{\Sigma}_{p_{i_{j_0}}} \mathbf{R}_0^T. \tag{3.72}$$

This concludes the theoretical description of the *esTIMATE* algorithm, and consequently this chapter. The conceptual description of the algorithm will be presented in the following chapter, as well as its utilization in association with the SLAM filter presented in Chapter 2.

4

Algorithm Implementation

THE complete algorithm proposed in this dissertation involves two different main algorithms that complement each other, the SLAM Kalman filter on one side, and the Inertial Map and Trajectory Estimation algorithm on the other. As mentioned previously, but not fully explained before, the two algorithms are to be used in conjunction with each other: the sensor-based map generated by the SLAM filter, as well as its associated uncertainty, is used as input to *esTIMATE*, which then provides an estimate of the inertial map and of the vehicle pose at the present time.

The previous chapters described the theory behind each main component of the integrated SLAM algorithm proposed. However, it remains to detail the practical aspects of the algorithm as a whole, especially in what concerns the particular implementation that was performed in order to experimentally validate the theoretical results. That is the scope of this chapter: to report the implementation of the complete algorithm comprising all its sections, from feature detection to the loop closure procedure. Several practical aspects of the implementation are also addressed.

This chapter is organized as follows. In Section 4.1, the complete algorithm is presented in full detail, and the composing parts are explained systematically, in opposition with the theoretical description of Chapters 2 and 3. Section 4.2 describes the process of obtaining the landmark measurements. In Section 4.3 the landmark association process is detailed, by explaining the concept of data association and presenting the algorithm employed. The loop closure proceeding is treated in Section 4.4. Finally, Section 4.5 addresses some practical issues regarding the implementation of the algorithms.

4.1 The Algorithm

Simultaneous localization and mapping is intrinsically a nonlinear problem, and, as such, subject to many interpretations, approaches, and solutions. However, the majority of the most popular algorithms tackle the problem where it is most obvious: the inertial frame. One of the reasons for this is that this provides the most human-readable outputs, the map and position in a static world. This raises the need to represent the pose of the vehicle in the state - for state space formulations, or in the probability distribution that describes the belief of the algorithm - for particle filter implementations. Nevertheless, the sensors, moving with the vehicle, provide vehicle-centered measurements that need to be transformed to the inertial frame in these formulations. Working directly on a sensor-based framework enables to leave the inertial pose representation out of the filtering process and avoids the transformation of the measurements. However, there is still need for an estimate of the inertial pose of the vehicle, as well as the map of the environment. The algorithm proposed in this thesis solves the SLAM problem in the sensor-based frame, on one hand, and provides the needed inertial map and vehicle pose on the other. This is accomplished by separating the problem in two parts. The diagram

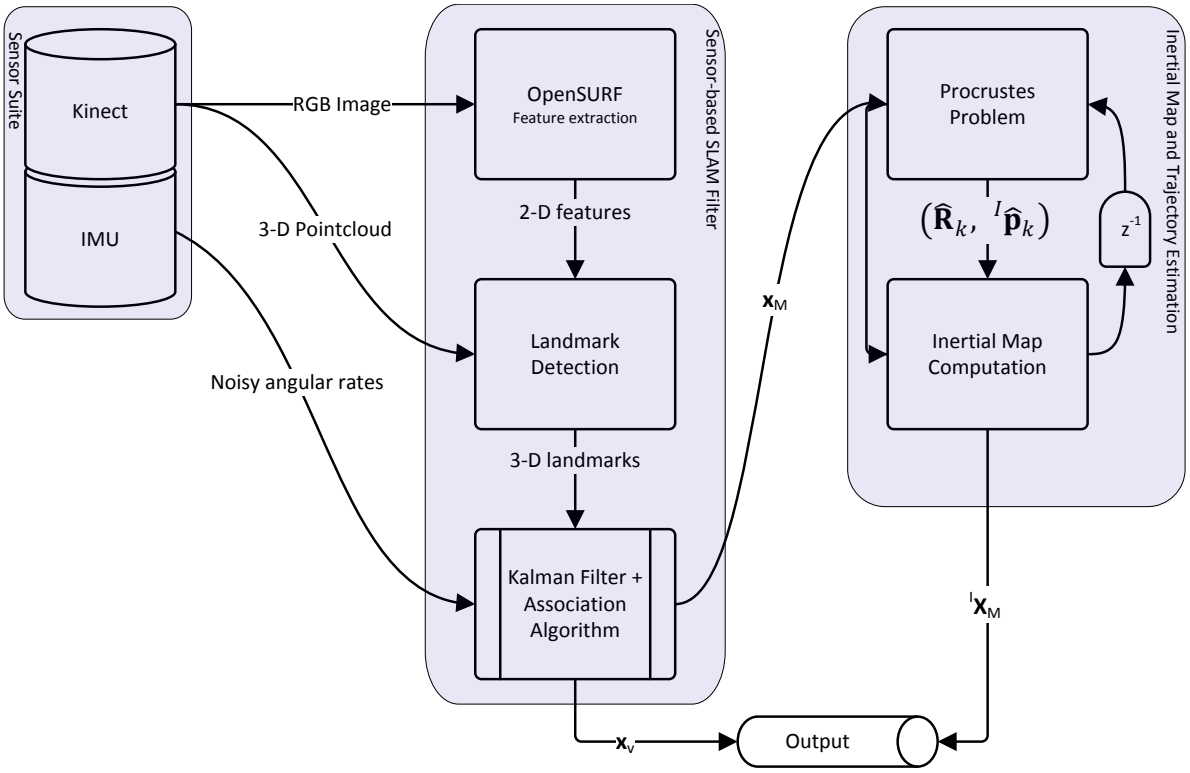


Figure 4.1: The flow of information in the complete SLAM algorithm: the SLAM filter and the *esTIMATE* algorithm.

in Fig. 4.1 shows the flow of information through all the stages of the algorithm, starting in the sensor suite composed by a *Microsoft Kinect™* and an Inertial Measurement Unit, which provides inputs to the SLAM filter, which in turn outputs estimates for the velocity of the vehicle and the angular velocity bias, as well as the estimated sensor-based landmark set - the input to the Inertial Map and Trajectory

Estimation. Each part of the complete algorithm, composed by the SLAM filter and the Inertial Map and Trajectory Estimation, comprises the following steps. The Sensor-based SLAM Filter, see Chapter 2, encompasses the landmark detection algorithm, that is detailed in Section 4.2, the data association algorithm, described in Section 4.3, and the SLAM filter itself (recall Section 2.3 for more details). The Tridimensional Inertial Map and Trajectory Estimation algorithm, whose design was treated in Chapter 3, is constituted by two main parts: the orthogonal Procrustes problem that estimates the vehicle pose, and the Inertial map computation. As its name suggests, all the variables involved in the Sensor-based SLAM Filter are expressed in sensor-based coordinates, and each of the processes directly concerned is detailed below, starting with the pseudo-code of the SLAM filter, in algorithm 1. The Feature detection, Landmark extraction and Data association processes are described in the following sections.

Algorithm 1 Sensor-based SLAM filter

Input: angular velocity measurements, set of observed landmarks
Output: landmark map and vehicle state vector, with respective covariances
- Predict step using (2.62).
- **Data association**: Associate observed landmarks with recently observed landmarks (last observation in less than t_{rec} seconds) using the association algorithm described in the Section 4.3.
- Update step using (2.64).
if enough time has passed ($t > t_{lc}$) and the old landmarks set is big enough **then**
  - **Loop Closing**: Associate old landmarks (last observation in more than t_{old} seconds) with recent ones using the association algorithm described in the Section 4.3.
  if size of association is greater than defined threshold m_{lc} **then**
    - Confirm association, using (2.65), and remove duplicates.
  end if
end if
if enough time has passed and there are landmarks too old (last observation in more than t_{ta} seconds) **then**
  - Throw away really old landmarks: state maintenance.
end if

Given that the product of the Kalman filter is the body-fixed map (the pose expressed in this frame is simply the origin and orientation of the frame), the Inertial Map and Trajectory Estimation algorithm is very important, because it offers the usual product of a localization algorithm, the position of the vehicle in the world. The steps taken are exposed in algorithm 2.

Algorithm 2 *esTIMATE* Algorithm

Input: Inertial landmark set $\mathcal{I}_{I_k}$ and respective covariance matrix $\mathbf{\Sigma}_{^I p_{i_k}}$, body-fixed landmark set $\mathcal{I}_B$ and its covariance matrix $\mathbf{\Sigma}_{p_{i_k}}$
Output: Updated inertial landmark set $\mathcal{I}_{I_{k+1}}$, rotation matrix $\mathbf{R}_k^*$, vehicle position ${}^I\mathbf{p}_k^*$, and respective covariances
if the simulation just started **then**
  - Initialize the inertial set with the true (known) rotation and translation of the observed landmarks
end if
- Estimate present rotation using (3.27) and the estimated landmarks which were present in the last iteration (they already have an inertial counterpart).
- Estimate present translation using (3.21).
- Rotate and translate all present landmarks using (3.30).
- Compute the uncertainty properties of the inertial rotation and translation estimates as well as of the inertial landmarks using the derivations in subsection 3.3.

As referred, the following sections serve the purpose of introducing the reader to the auxiliary algorithms that are required in this work but that were not the focus of the development of this work.

4.2 Feature Detection and Landmark Extraction

The *Microsoft Kinect*^{TM}camera provides both RGB and depth images of the environment. These images are then combined to generate a coloured tridimensional pointcloud. In order for this pointcloud to be usable in an algorithm such as the one presented here, some kind of feature detector must be used so that only a small number of interest points is chosen. This section describes the process of detecting features on a flat image and extracting 3-D landmarks from the pointcloud.

The main idea behind is to use the renowned SURF algorithm [23] to detect features on the 2-D RGB image. The obtained bidimensional points are expressed in *pixel* coordinates, which then are easily matched to the pointcloud, resulting in a set of 3-D landmarks.

SURF, Speeded Up Robust Features, is a scale and rotation invariant interest point detector and descriptor, which, and quoting [23], "*approximates or even outperforms previously proposed schemes with respect to repeatability, distinctiveness, and robustness, yet can be computed and compared must faster*". The algorithm was designed as a tool to search for discrete image correspondences, divided in three stages: interest point detection - corners, blobs and t-junctions, feature description and image matching. However, for the purpose of extracting landmarks from the environment, only the interest point detection is important.

The algorithm will not be explained here, as its implementation was used as a *black box* - the important facts to keep in mind are the invariance and repeatability permitted by SURF. The version used was a MATLAB implementation of the OpenSURF library (see [31] for more details) which can be found in [32]. A simple example of the results of this algorithm is shown in Fig. 4.2. The same scene was captured from two different angles and the resulting pictures were analysed through SURF to extract features that were then matched in both pictures as the figure depicts. Although the input images are similar, it is possible to see the correct correspondence between the features, thus hinting at the repeatability and invariance of the algorithm.

Figure 4.2: An example of the repeatability of features in two different images of the same scenery.

4.3 Data Association

The data association in a simultaneous localization and mapping framework is far from trivial, as there is no way to know exactly which observed landmark corresponds to a mapped landmark or if it does not correspond to anything at all. The problem here is that there is uncertainty associated with both the noisy observed landmark set and the estimated map, which asks for a probabilistic answer.

The simplest way of associating landmarks would be to pick an observed landmark, compute its distance to all the mapped landmarks and pick the closest landmark as the correct association, provided that the distance was below a defined threshold. Then, proceed to the next observed landmark until all the observed landmarks had been associated or marked as new. This strategy poses a problem: in a probabilistic setting, a landmark (both observed and mapped) is represented by its mean and covariance, and by computing the euclidean distances, the covariances, and hence the uncertainties, are simply ignored.

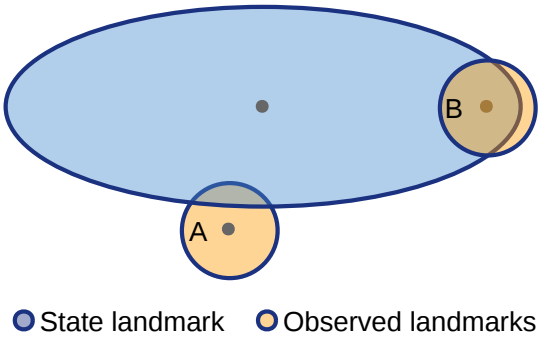


Figure 4.3: Schematic of the situation of associating two landmarks to an existing one.

Figure 4.3 shows a bidimensional situation where not taking into account the uncertainty leads to a wrong association. Here, the ellipses represent the uncertainty ellipses and the dots in their center represent the mean. It is easily seen that the shortest distance between means is that of landmark *A*

with the state landmark. However, the intersection of the uncertainty ellipses is much greater in the case of landmark *B*, which would mean that the probability of landmark *B* being the same as the state landmark is greater. This shows that using the euclidean distance is of course a very naive approach, although it serves as an introduction for the notion of the *Mahalanobis* distance.

Definition 5 (*Mahalanobis* distance [24])**.** *The Mahalanobis distance, also denoted as Normalized Innovation Squared (NIS), is given by*

$$D^2_{ij_k} := (\mathbf{y}_{i_k} - \hat{\mathbf{p}}_{j_k})^T (\mathbf{\Xi}_{j_k} + \mathbf{\Theta}_{i_k})^{-1} (\mathbf{y}_{i_k} - \hat{\mathbf{p}}_{j_k}), \tag{4.1}$$

where $\mathbf{y}_{i_k}$ represents the i-th observed landmark at time k, $\hat{\mathbf{p}}_{j_k}$ is the j-th estimated state landmark at the same instant. Recall that $\mathbf{\Theta}_k$ and $\mathbf{\Xi}_k$ are the measurement noise and state disturbance covariances, respectively.

The Mahalanobis distance is known to follow a χ^2 distribution with 3 degrees of freedom, as the vectors are tridimensional.

The *Mahalanobis* distance can be understood as a probabilistic distance between two landmarks, related to the intersection of their uncertainty ellipsoids, in the form of a distance weighted by the joint uncertainty covariances, and thus, the best association, i.e., the one with the maximum likelihood, is the one with the lowest *normalized innovation squared*.

While describing a possible, yet naive, way of associating landmarks, the existence of a decision threshold was mentioned. Using a probabilistic definition for the distance, such as the *Mahalanobis* distance, it is possible to perform the following statistical test, to decide if two landmarks are individually compatible, i.e., if they may be statistically associated:

$$D^2_{ij_k} < \gamma_\alpha, \tag{4.2}$$

where $\gamma_\alpha := F^{-1}_{\chi^2_3}(\alpha)$ respects the relation $\mathrm{P}(\Gamma < \gamma_\alpha) = \alpha$ with $\Gamma \sim \chi^2_3$. α is usually 0.95, and $1 - \alpha$ represents the percentage of correct associations permitted to be discarded.

Despite the solid statistical foundation of this strategy, it presents some limitations, mainly because there is usually a whole batch of measurements which joint information is not taken into account when associating landmarks using only an individual association process, such as the Nearest Neighbor [33] plainly described in this section. This makes pertinent the definition of the joint *Mahalanobis* distance of a whole association hypothesis, in which two sets of landmarks are associated in batch.

Definition 6 (Joint *Mahalanobis* distance [24])**.** *The Joint Mahalanobis distance is given by*

$$D^2_{\mathcal{H}_k} := (\mathbf{y}_k - \hat{\mathbf{p}}_{\mathcal{H}_k})^T \left(\mathbf{H}_{\mathcal{H}_k} \mathbf{\Xi}_k \mathbf{H}^T_{\mathcal{H}_k} + \mathbf{\Theta}_k\right)^{-1} (\mathbf{y}_k - \hat{\mathbf{p}}_{\mathcal{H}_k}), \tag{4.3}$$

where $\mathcal{H}_k$ is the association hypothesis at time k, $\mathbf{y}_k$ is the full measurement vector, $\hat{\mathbf{p}}_{\mathcal{H}_k}$ is the hypothetically associated landmark vector mapped to the full state vector through $\mathbf{H}_{\mathcal{H}_k}$, $\hat{\mathbf{p}}_{\mathcal{H}_k} = \mathbf{H}_{\mathcal{H}_k} \hat{\mathbf{x}}_{F_k}$. This quantity is known to follow a χ^2 distribution with as many levels of freedom as the dimension of

the landmark vectors to be associated, $3 \times \text{size}(\mathcal{H}_k)$.

In this case, the statistical test takes the form of

$$D^2_{\mathcal{H}_k} < \gamma_{\alpha_{3\times\text{size}(\mathcal{H}_k)}}, \tag{4.4}$$

where $\gamma_{\alpha_{3\times\text{size}(\mathcal{H}_k)}} := F^{-1}_{\chi^2_{3\times\text{size}(\mathcal{H}_k)}}(\alpha)$ respects the relation $\text{P}(\Gamma < \gamma_\alpha) = \alpha$ with $\Gamma \sim \chi^2_{3\times\text{size}(\mathcal{H}_k)}$. α is usually 0.95, and $1-\alpha$ again represents the percentage of correct associations permitted to be discarded.

The naive algorithm described in the beginning of this section is what is classified as a Depth First Search, in the field of search algorithms. Its extension to the use of joint compatibility yields the association algorithm that was adopted therein, the *Joint Compatibility Branch and Bound* (see [34], [24] and [33, Chap. 3] for further information), which is detailed in Algorithm 3.

Algorithm 3 Joint Compatibility Branch and Bound - JCBB

Input: set of state landmarks in which to search $\mathbf{cSet}$, observed set $\mathbf{oSet}$, state and observation covariance matrices $\mathbf{P}$ and $\mathbf{R}$, best association hypothesis to date bH, current working hypothesis cH and current tree depth
Output: association hypothesis
if current depth $>$ tree depth **then**
  if $\text{size}(cH) > \text{size}(bH)$ or $\text{size}(cH) = \text{size}(bH)$ and cH better than bH **then**
    $bH = cH$
  end if
else
  for i in all the landmarks not associated in cH **do**
    - Calculate *Mahalanobis* distance, D^2_{ij} using (4.1) between the current observed landmark j (determined by the depth of the tree) and state landmark i.
    if $D_{ij} < F^{-1}_{\chi^2_3}(0.95)$ **then**
      Calculate the joint *Mahalanobis* distance, D_{cH} using (4.3).
      if $D_{cH} < F^{-1}_{\chi^2_{3(\text{size}(cH)+1)}}(0.95)$ **then**
        - Augment the current hypothesis by introducing the (i,j) association.
        - Recursively run JCBB with the new hypothesis.
      end if
    end if
  end for
  if $\text{size}(cH)+$tree depth$-$current depth$> \text{size}(bH)$ **then**
    - Do not associate current observed landmark j.
    - Recursively run JCBB with the new hypothesis.
  end if
end if

This algorithm performs a depth first search in the search tree, where each level represents an observed landmark and each node constitutes the assignable state landmarks, or the null assignment as Fig. 4.4 depicts.

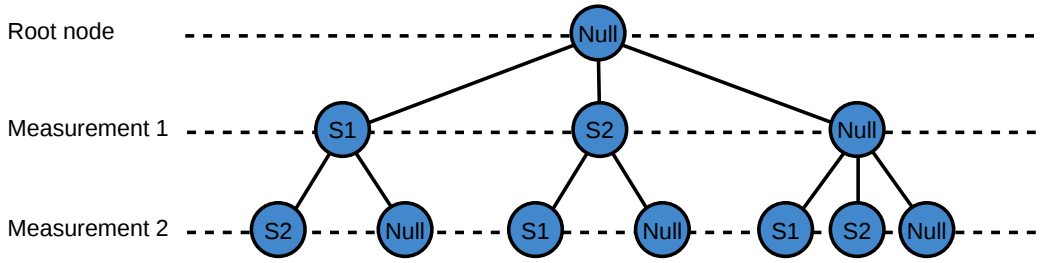


Figure 4.4: Example of a search tree with two measurements and two state landmarks.

When searching, each node corresponds to a $D_{\mathcal{H}_k}$ and the first in each level to be expanded is the one with the lowest joint *Mahalanobis* distance. Note that a node is only expanded if the individual assignment is compatible.

As stated before, although the association algorithm is not within the scope of this work, it is of key importance to the well-functioning of the proposed SLAM filter. Furthermore, it intervenes in two steps of value in the whole algorithm, associating observed landmarks with state landmarks and in the loop closing procedure, associating old landmarks with more recent ones, as mentioned in Subsection 2.3.4. This will be further explained in the next section.

4.4 Loop Closure

In the long term, SLAM may become computationally burdensome, especially in feature-rich environments. Note that, in state-space SLAM, the features are included in the filter state, and, as their uncertainty is coupled, this makes the size of the data grow with $\mathcal{O}(n^2)$. Moreover, the update step in Subsection 2.3.3 involves operations with matrices that can become very large with a small map. For example, consider a map with 40 landmarks, which would be normal for a small room of 16 m^2. The associated covariance would have 120 rows and columns, i.e., 14400 elements. The association of measurements to a map of this size is an even greater problem, as the search space has $N_m \times (N_M + 1)$ possible associations, N_m being the number of measurements and N_M the number of mapped landmarks. Also, in a long loop, as time goes by, the uncertainty and error of the older landmarks grows larger, making it harder to obtain a consistent map. That can lead to wrong associations, thus degrading the performance of any SLAM algorithm. For that reason, the algorithm proposed divides the complete landmark set in three different sets. Building a map using several local maps is not a novel idea, the works in [11] and [12] proposed dividing the problem into several locally independent maps, and then joining them to obtain a consistent map. However, in the approach followed in this dissertation, the sets are divided into subsets according to the last time they were visible. Furthermore, the subsets are not independent, as the state of the SLAM filter is not shortened to contain only a single subset. In fact, the division in subsets is only for the purpose of making landmark association less computationally demanding. The set $\mathcal{I}_B$ is divided into the three non-overlapping sets $\mathcal{I}_{rec}$, containing the landmarks whose last visualization occurred in less than t_{rec} seconds, $\mathcal{I}_{old}$, which contains the

landmarks whose last visualization occurred in more than t_{old} seconds, and $\mathcal{I}_{gap}$, encompassing the remaining landmarks. With this division, it is possible to efficiently associate the observed landmarks with landmarks in $\mathcal{I}_{rec}$, thus reducing the search space. A potential problem is the duplication of landmarks when seen after being placed in $\mathcal{I}_{gap}$ or $\mathcal{I}_{old}$. That is where the basic loop closure procedure that was devised takes place. A loop closing algorithm aims at solving the problem of deciding whether or not a vehicle has returned to a previously visited area after exploring unknown environment. The proposed solution for the loop closing problem, similar to the one used in [18], builds on the idea of dividing the map in subsets. In fact, without the aforementioned duplication of landmarks in the sets $\mathcal{I}_{rec}$ and $\mathcal{I}_{old}$, this procedure would not work at all. That is due to the main idea behind the algorithm: association of recently visible landmarks with older, possibly duplicated, landmarks. This is done by performing a periodic search for duplicates in the sets of recent and old landmarks, thus finding previously visualized features. The data association algorithm 3 is employed to find jointly compatible associations between landmarks in the two sets, thus triggering a loop closure if a minimum number of associations is found. The insertion of the algorithm in the SLAM filter was shown in Algorithm 1.

Similar to landmark association, the loop closing problem is not the main focus of this work, but rather a demonstration that the filter is able to provide consistent estimates that will enable any loop closing technique to perform in a good fashion, in real world scenarios, in particular if more efficient and advanced algorithms are employed.

4.5 Implementation Issues

So far, only a conceptual, though thorough, description of the algorithm was made. However, the implementation of an algorithm of this dimension involves many technical and practical decisions.

Given that the main idea behind this work was to prove that the SLAM problem could be solved not discarding any of its intrinsic nonlinearities and still have global convergence results, absent from the famous EKF solutions, for example, the focus was not on the implementation itself, but on the theoretical results. That said, the implementation, at this stage, was thought for offline purposes that could, however, lead to an optimized real-time implementation of the algorithm. For that purpose, and due to the flexibility of the language, the algorithm was implemented in MATLAB[1].

As MATLAB performs optimized matrix operations, the process of programming the algorithm tried to take that into account to save operation time. One of the main focus was the conversion of loops into matricial operations. The stage where most effort was allocated was the computation of the uncertainty of the estimates of the Inertial Map and Trajectory Estimation algorithm, as it is the most computationally demanding part of the algorithm – see Appendix A for a full time complexity analysis. Recall that (3.57), (3.60), (3.65), (3.68), and (3.69) involve sums which, at first sight, must be computed by looping through all the landmarks in the set $\mathcal{I}_T$. Nevertheless, equivalent matrix operations were found to avoid the use of loops in the programming. A description of the correspondence between each sum and the matrix operation can be found in Appendix B. As a measure of the difference between

[1] `http://www.mathworks.com/products/matlab`

using loops for the calculations or the proposed method, consider the following figure that depicts the time taken in each iteration of the algorithm to compute the uncertainty of the inertial estimates for each methodology. It is clear that although still somewhat lengthy, the matrix method is considerably better when more landmarks are involved.

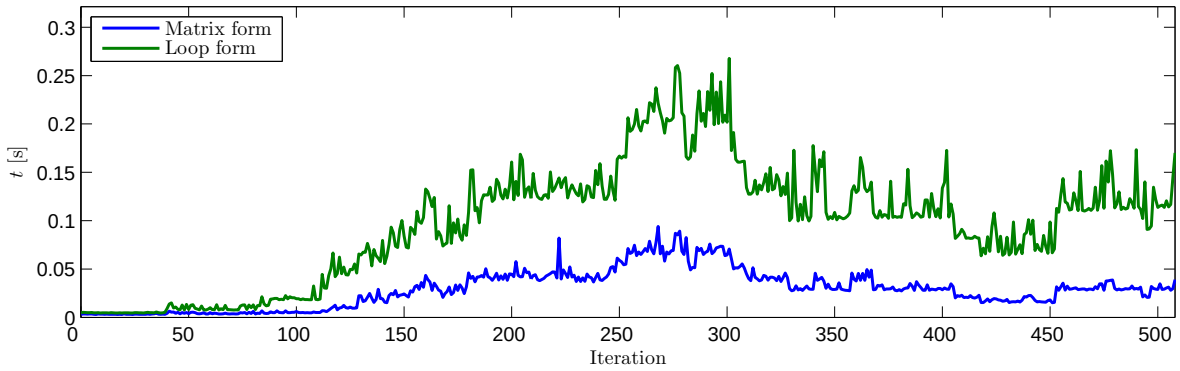


Figure 4.5: Time taken to compute uncertainty of the *esTIMATE* algorithm while using loops or matrix operations.

5

Simulation

THE first step after the development of any scientific work is validation through experiment. Ideally, the validation would be performed in a real platform obtaining real data. However, before proceeding to real world scenarios, and spending resources in preparing realistic experiments, it is important to validate the performance of the algorithm in a simulated setting. The validation of an algorithm for positioning, mapping or both has various important steps, namely, the creation of a landmark map, based on which the landmark observations may be synthesized, the definition of a motion model, to generate the simulated inputs to the filter and to move the vehicle, the design of the trajectory of the vehicle, and a model for the sensors. This chapter explains the process of creating the inertial map, the motion model, the vehicle path, and the sensor models. Finally, the complete algorithm is tested in simulation, and the results are also presented.

The chapter is organized as follows. Section 5.1 presents the design of an inertial map and a trajectory for the vehicle. In Section 5.2 a motion model is presented. Section 5.3 details the sensor models and Section 5.4 the algorithm parameters. The chapter culminates in Section 5.5, which presents the simulation results that validate the SLAM algorithm.

5.1 Inertial Map and Trajectory

The simulated world consists of a corridor enclosed over itself in a square shape, as Fig. 5.1 shows. The landmarks, represented as blue stars in the figure, are randomly uniformly distributed in each run of the simulation, covering only the corridors. The line in red is the trajectory of the vehicle in the same environment.

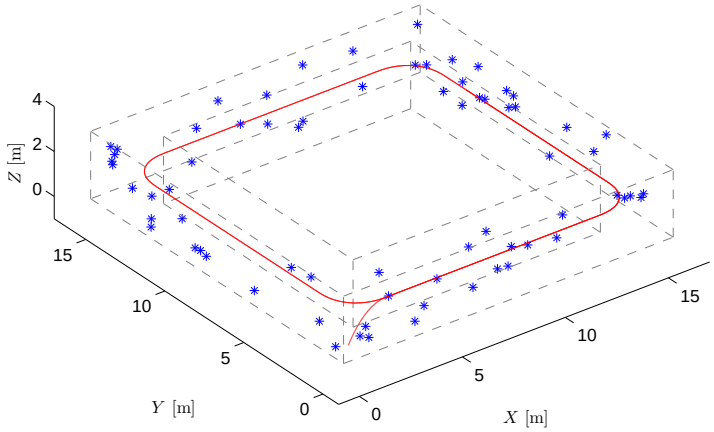


Figure 5.1: Example of a simulated map and the trajectory followed by the vehicle.

5.1.1 Trajectory design

The path was designed in order to allow the vehicle to cycle through the four corridors that compose the map. The vehicle starts at rest for some time and then takes-off, until it reaches constant speed at the desired altitude. The velocity is maintained while in a corridor, changing direction when the vehicle enters a corner. Figure 5.2 depicts the chosen trajectory in bidimensional views.

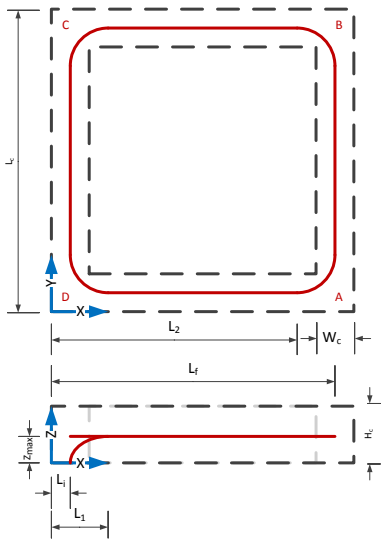


Figure 5.2: Bidimensional path followed by the vehicle during simulation.

Given that the trajectory of the vehicle is divided in various phases, it was mathematically defined as such, in stages:

- Take-off
- Steady-flight through a corridor
- Making a turn at a corner

The three inertial coordinates are described by the following 5 degree polynomials. The coefficients depend on the flight phase, and will be computed next.

$$\begin{cases} x(t) = a_x t^5 + b_x t^4 + c_x t^3 + d_x t^2 + e_x t + f_x \\ y(t) = a_y t^5 + b_y t^4 + c_y t^3 + d_y t^2 + e_y t + f_y \\ z(t) = a_z t^5 + b_z t^4 + c_z t^3 + d_z t^2 + e_z t + f_z \end{cases} \tag{5.1}$$

Note that the linear acceleration of the vehicle is given by ${}^I\mathbf{a}_k = \begin{bmatrix} \ddot{x}(t = kT_s) & \ddot{y}(t = kT_s) & \ddot{z}(t = kT_s) \end{bmatrix}^T$.

The corridors As mentioned before, the flight through a corridor is in a steady state, and therefore the acceleration is zero.

The corners Flying through the corners is made at constant altitude, and varying x and y (both expressed in the inertial frame). Hence, the equation in z is merely $z(t) = f_z = Z_{max}$. There are 6 boundary conditions for both x and y, the constants depending on the corner:

$$\begin{array}{lll} x(t_i) = X_i & \dot{x}(t_i) = U_i & \ddot{x}(t_i) = 0 \\ x(t_f) = X_f & \dot{x}(t_f) = U_f & \ddot{x}(t_f) = 0 \\ y(t_i) = Y_i & \dot{y}(t_i) = V_i & \ddot{y}(t_i) = 0 \\ y(t_f) = Y_f & \dot{y}(t_f) = V_f & \ddot{y}(t_f) = 0 \end{array} \tag{5.2}$$

where t_i and t_f are the time instants when the vehicle enters and leaves the corner, respectively. The following table and corresponding explanation figure present the values of each constant according to the corner. Thus, the polynomial that describe the position in the corner must be of 5th order as (5.1)

Corner	X_i	X_f	Y_i	Y_f	U_i	U_f	V_i	V_f
A	L_2	L_f	L_i	L_1	v_∞	0	0	v_∞
B	L_f	L_2	L_2	L_f	0	$-v_\infty$	v_∞	0
C	L_1	L_i	L_f	L_2	$-v_\infty$	0	0	$-v_\infty$
D	L_i	L_1	L_1	L_i	0	v_∞	$-v_\infty$	0

Table 5.1: Correspondence of boundary conditions for (5.2).

shows. In this case, the coefficients $a_{x,y} - f_{x,y}$ were derived from the boundary conditions expressed by (5.2) and are given by the following expressions, where the y-coefficients are equal to the x ones substituting the pair $(X_{(.)}, U_{(.)})$ for $(Y_{(.)}, V_{(.)})$.

- $a = \frac{3(-t_f U_f + t_i U_f - t_f U_i + t_i U_i + 2X_f - 2X_i)}{(t_f - t_i)^5}$
- $b = \frac{7t_f^2 U_f + t_f t_i U_f - 8t_i^2 U_f + 8t_f^2 U_i - t_f t_i U_i - 7t_i^2 U_i - 15t_f X_f - 15t_i X_f + 15t_f X_i + 15t_i X_i}{(t_f - t_i)^5}$
- $c = \frac{2(2t_f^3 U_f + 8t_f^2 t_i U_f - 7t_f t_i^2 U_f - 3t_i^3 U_f + 3t_f^3 U_i + 7t_f^2 t_i U_i - 8t_f t_i^2 U_i - 2t_i^3 U_i - 5t_f^2 X_f - 20t_f t_i X_f - 5t_i^2 X_f + 5t_f^2 X_i + 20t_f t_i X_i + 5t_i^2 X_i)}{(t_f - t_i)^5}$
- $d = \frac{6t_f t_i(-2t_f^2 U_f - t_f t_i U_f + 3t_i^2 U_f - 3t_f^2 U_i + t_f t_i U_i + 2t_i^2 U_i + 5t_f X_f + 5t_i X_f - 5t_f X_i - 5t_i X_i)}{(t_f - t_i)^5}$
- $e = \frac{-12t_f^3 t_i^2 U_f + 8t_f^2 t_i^3 U_f + 5t_f t_i^4 U_f - t_i^5 U_f + t_f^5 U_i - 5t_f^4 t_i U_i - 8t_f^3 t_i^2 U_i + 12t_f^2 t_i^3 U_i + 30t_f^2 t_i^2 X_f - 30t_f^2 t_i^2 X_i}{(t_f - t_i)^5}$
- $f = \frac{4t_f^3 t_i^3 U_f - 5t_f^2 t_i^4 U_f + t_f t_i^5 U_f - t_f^5 t_i U_i + 5t_f^4 t_i^2 U_i - 4t_f^3 t_i^3 U_i - 10t_f^2 t_i^3 X_f + 5t_f t_i^4 X_f - t_i^5 X_f + t_f^5 X_i - 5t_f^4 t_i X_i + 10t_f^3 t_i^2 X_i}{(t_f - t_i)^5}$

Take-off Take-off is very similar to the corners, the difference being that y is constant, $y(t) = f_y = L_i$, and z varies from 0 to the flight altitude Z_{max}. The boundary conditions are

$$\begin{aligned} x(t_i) &= X_i \quad \dot{x}(t_i) = U_i \quad \ddot{x}(t_i) = 0 \\ x(t_f) &= X_f \quad \dot{x}(t_f) = U_f \quad \ddot{x}(t_f) = 0 \\ z(t_i) &= Z_i \quad \dot{z}(t_i) = W_i \quad \ddot{z}(t_i) = 0 \\ z(t_f) &= Z_f \quad \dot{z}(t_f) = W_f \quad \ddot{z}(t_f) = 0 \end{aligned} \tag{5.3}$$

where the correspondence is now expressed in table 5.2. Note that the coefficients $a_{x,z} - f_{x,z}$ are once

X_i	X_f	Z_i	Z_f	U_i	U_f	W_i	W_f
L_i	L_1	0	Z_{max}	0	v_∞	0	0

Table 5.2: Correspondence of constants boundary conditions for (5.3).

more given by the ones presented for the corners, by substitution of the pair $X_{(.)}, U_{(.)}$ by $Z_{(.)}, W_{(.)}$ for the z coefficients. One important fact in the definition of the trajectory is that it must serve as an input to the to-be-defined motion model in some way. As it shall be seen, the path is enforced by computing the input forces by the means of determining the inertial linear acceleration, and applying it to the motion model.

5.2 Motion model

The simulation implementation raises the need of creating a motion model, which can provide the angular velocity readings and move the vehicle in the simulated world. Although the main idea behind the simulation process is to make as similar as possible to a real world process, the factors that are really important to mimic are the ones fed to the proposed algorithm: the measured landmarks and the angular rates. That being said, it was still considered relevant to use a motion model accordant to the kind of vehicle intended, a quadrotor. For that reason, the following derivations take into account the set of forces and moments presented in Fig. 5.3. Note that, as the algorithm works in discrete time, it was found best to derive a discrete version of the motion model.

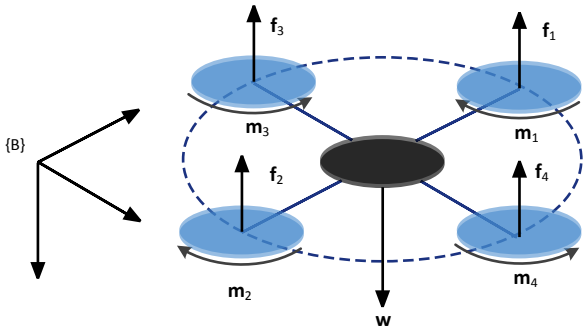


Figure 5.3: Simplified model of the forces and moments acting on a quadrotor.

5.2.1 Kinematic model

The kinematic model used in the simulation is described by

$$\begin{cases} \mathbf{R}_{k+1} = \mathbf{R}_k \left(\mathbf{I}_3 + T_s \mathbf{S}\left[\boldsymbol{\omega}_k\right]\right) \\ \boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + T_s \boldsymbol{\alpha}_k \\ {}^I\mathbf{p}_{k+1} = {}^I\mathbf{p}_k + T_s {}^I\mathbf{v}_k \\ {}^I\mathbf{v}_{k+1} = {}^I\mathbf{v}_k + T_s {}^I\mathbf{a}_k, \end{cases} \tag{5.4}$$

where ${}^I\mathbf{a}_k$ is the linear acceleration of the vehicle expressed on the inertial frame, and comes from the differentiation of the chosen path, that was already described, and $\boldsymbol{\alpha}_k$ is the angular acceleration also expressed on the same frame. These equations, apart from the discretization, are purely kinematic and therefore standard. The fact that the vehicle is a quadrotor is only used in the dynamic part of the model. In this kinematic model, the rotation matrix is detailed later, thus yielding $\boldsymbol{\omega}_k$. The linear acceleration is obtained through double differentiation of the vehicle's inertial coordinates given by its predefined trajectory. The model is initialized with the following initial conditions

$$\mathbf{R}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad {}^I\mathbf{p}_0 = \begin{bmatrix} L_i & L_i & 0 \end{bmatrix}^T, \quad \boldsymbol{\omega}_0 = \mathbf{0}, \quad \text{and} \quad {}^I\mathbf{v}_0 = \mathbf{0},$$

where L_i was defined in the last section.

5.2.2 Dynamic model

The dynamic model below is used to obtain the inertial counterpart of the force exerted by the controls, which is then useful to compute the rotation matrix, as in a quadrotor the force is always perpendicular to the platform, and thus collinear with the z body axis (see Fig. 5.3). The dynamical model is derived using the second Newton law of motion, that relates the acceleration of the vehicle with the total forces applied, and with the conservation of angular momentum, which then relates the variation of

the angular momentum with the applied torques [35],

$$\begin{cases} \sum \mathbf{f}_k = m\,{}^I\mathbf{a}_k = {}^I\mathbf{f}_k - {}^I\mathbf{w} \\ \sum \mathbf{m}_k = \mathbf{J}\boldsymbol{\alpha_k} + \mathbf{S}\left[\boldsymbol{\omega}_k\right]\mathbf{J}\boldsymbol{\omega}_k \end{cases}. \tag{5.5}$$

The quantity ${}^I\mathbf{w}$ is the weight of the vehicle expressed in the inertial frame, ${}^I\mathbf{w} := \begin{bmatrix} 0 & 0 & mg \end{bmatrix}^T$, where m is the mass of the vehicle and g is the acceleration of gravity, $g := 9.81\ \mathrm{m/s}$. It is clear that it maintains the same value and direction at all times. The sum of the force exerted by the controls, ${}^I\mathbf{f}_k$ is given by ${}^I\mathbf{f}_k := \mathbf{R}_k\left(\mathbf{f}_{1_k} + \mathbf{f}_{2_k} + \mathbf{f}_{3_k} + \mathbf{f}_{4_k}\right)$ where the $\mathbf{f}_{i_k}$s are the ones in Fig. 5.3. In the second expression, $\mathbf{J}$ is the moment of inertia on the body-fixed axes - meaning that is constant and independent of the vehicle attitude. It is important to notice that the second equation is not necessary to the motion model, as the angular velocity is determined by the evolution of the rotation matrix, and because for the model to work there is no need to particularize the control forces (only their sum is needed) which removes the need for the knowledge of the total moments.

5.2.3 Rotation matrix construction

The a priori knowledge of the trajectory and of the form of the vehicle allows the computation of the rotation matrix that represents the attitude of the vehicle in each iteration. For example, examine Fig. 5.3 once more. It is clear that the forces produced by the motors are all aligned with the z axis, and the forces are calculated using the linear acceleration, which, in turn is available through differentiation of the trajectory. Thus, ${}^I\mathbf{k}_{B_k}$ is simply the unit vector opposite to the sum of the control forces. Another quantity that is intrinsically related with the way the body-fixed frame is defined is the linear velocity $\mathbf{v}_k$: the velocity is always in the xz plane, which means that its cross-product with the $\mathbf{k}$ versor is aligned with the y axis. Finally, to obtain a right-handed frame the x axis versor is obtained by the cross product between $\mathbf{j}_B$ and $\mathbf{k}_B$. The following expressions summarize the procedure.

$$\begin{cases} {}^I\mathbf{i}_{Bk} = {}^I\mathbf{j}_{Bk} \times {}^I\mathbf{k}_{Bk} \\ {}^I\mathbf{j}_{Bk} = \dfrac{{}^I\mathbf{k}_{Bk} \times {}^I\mathbf{v}_k}{|{}^I\mathbf{v}_k|} \\ {}^I\mathbf{k}_{Bk} = -\dfrac{{}^I\mathbf{f}_k}{|{}^I\mathbf{f}_k|} \end{cases} \tag{5.6}$$

The rotation matrix is, recalling (2.1), given by

$$\mathbf{R}_k = \begin{bmatrix} {}^I\mathbf{i}_B & {}^I\mathbf{j}_B & {}^I\mathbf{k}_B \end{bmatrix}. \tag{5.7}$$

5.3 Sensor model

The sensor suite employed in this work is composed simply by a triad of orthogonally mounted rate-gyros and an RGB-D camera. For simulation purposes, there is only need to obtain the angular rates

and the landmark observations. This section details the process of synthesizing these measurements.

The vision sensor to be used is a RGB-D camera, such as *Microsoft Kinect™*. Its field of view is depicted in Fig. 5.4[1]. In simulation, the real map of the world is rotated and translated to the body, and

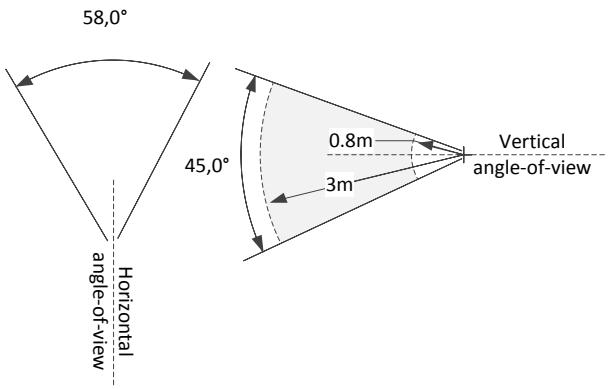


Figure 5.4: The field of view of the *Microsoft Kinect™*.

the landmarks within the field of view are maintained and additive Gaussian white noise is included, in accordance with the following steps.

Algorithm 4 Obtain simulated landmark observations

Input: true inertial map, true pose of the vehicle $\left(\mathbf{R}_k, {}^I\mathbf{p}_k\right)$
Output: set of visible landmarks expressed in the body-fixed frame: $\mathcal{I}_O$
- Convert the inertial map to the body-fixed frame: $\mathbf{p}_{i_k} = \mathbf{R}_k^{T\,I}\mathbf{p}_i - {}^I\mathbf{p}_k$
- Compute the norm of each body-fixed landmark: $\sqrt{\mathbf{p}_i^T\mathbf{p}_i}$
- Find which landmarks are within viewing range (as the landmarks are body-centric, their norm corresponds to their distance to the vehicle)

if No landmarks are within viewing range **then**
  exit Nothing is observed
end if
- Calculate the elevation and azimuth of each landmark using $az = \arctan_2(p_{i_k}^y, p_{i_k}^x)$ and $el = \arctan_2\left(-p_{i_k}^z, \sqrt{{p_{i_k}^x}^2 + {p_{i_k}^y}^2}\right)$
- Find which of the landmarks within range are also within angle of view

if No landmarks are within angle of view **then**
  exit Nothing is observed
end if
if any of the chosen landmarks are collinear **then**
  - Choose the closest to the vehicle
end if

For the angular velocity measurements, the true angular velocity is taken from the motion model and also added Gaussian white noise, as well as the randomly generated constant bias.

5.4 Simulation parameters

The SLAM filter, as any Kalman filter, has tunable parameters, namely the model disturbance noise covariance, the measurement noise covariance and the filter initial conditions. Furthermore, the algorithm here proposed divides the estimated map in subsets of landmarks depending on the last time

[1] Field of view information taken from the OpenKinect community at `http://openkinect.org/wiki/Imaging_Information`.

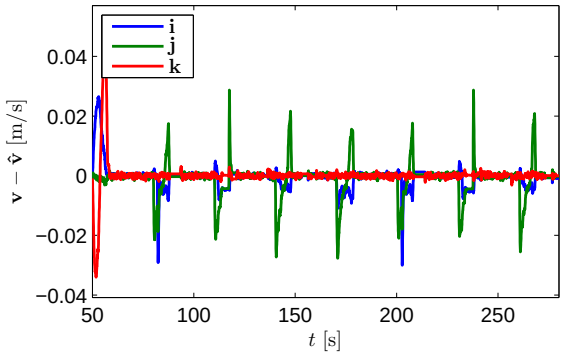
they were observed, raising the need for time thresholds. The loop closure and state maintenance procedures also have tuning knobs - the first time they occur and the number of landmarks needed to trigger an action. The output noise covariance is $\mathbf{\Theta}_k = \sigma_\theta^2 \mathbf{I}_3$, $\sigma_\theta = 0.032$ m, the state disturbance covariance is given by $\mathbf{\Xi}_k = T_s \operatorname{diag}\left(\sigma_v^2 \mathbf{I}_3, \sigma_{b_\omega}^2 \mathbf{I}_3, \sigma_{p_1}^2 \mathbf{I}_3, \ldots, \sigma_{p_N}^2 \mathbf{I}_3\right)$, with $\sigma_v = 0.05$ m/s, $\sigma_{b_\omega} = 10^{-5}$ rad/s and $\sigma_{p_i} = 10^{-4}$ m. The initial state covariance was set to $\mathbf{\Sigma}_0 = \operatorname{diag}\left(\sigma_{v_0}^2 \mathbf{I}_3, \sigma_{b_{\omega_0}}^2 \mathbf{I}_3\right)$ with $\sigma_{v_0} = 0.011$ m/s and $\boldsymbol{\sigma}_{b_{\omega_0}} = 0.022$ rad/s. Note that at $k = 0$ there are no landmarks in the state, but any new landmark is initialized with $\mathbf{\Sigma}_{p_{i_0}} = \sigma_{p_{i_0}}^2 \mathbf{I}_3$, $\sigma_{p_{i_0}} = 0.017$ m. Finally the initial estimates of velocity and angular bias are set to zero. As to the SLAM-specific parameters, the recent landmark set is composed by landmarks seen in the last 15 s, and the old by landmarks not seen in more than 100 s. The first loop closure is tried at 100 s and it is triggered if at least 6 landmarks are associated. Finally, any landmark not visible for more than 200 s is discarded.

5.5 Results

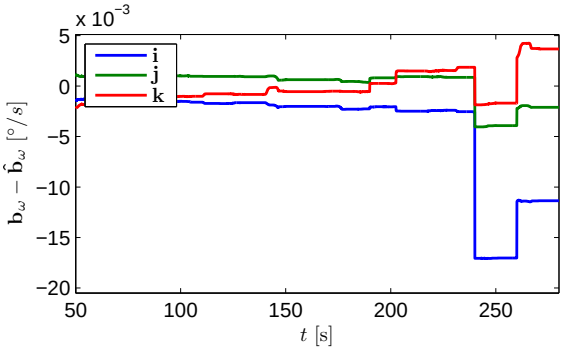
The simulated environment consists of 70 landmarks spread throughout a 16m×16m×3m map, including a closed 2 m wide corridor in the outer borders of the map. Recall that the trajectory is simply a loop at half-height, with the vehicle starting on the floor. In this section, the results of a typical run will be presented. The simulation starts with the vehicle stopped for 50 seconds, to let the filter converge, which then takes off and circles through the corridors for around 280 s at an average speed of 0.45 m/s. The zero-mean noise added to the angular velocity measurements is normal-distributed with a standard deviation of $\sigma_{\omega_m} = 3 \times 10^{-2}$ °/s at each coordinate and the noise included in the landmark observations is also zero-mean Gaussian white noise with a standard deviation of $\sigma_y = 10^{-3}$ m. The section is divided firstly in results of the Sensor-based Filter, followed by the *esTIMATE* algorithm results. Finally, the consistency and convergence of the complete algorithm is demonstrated in Subsection 5.5.3.

5.5.1 Sensor-based SLAM Filter

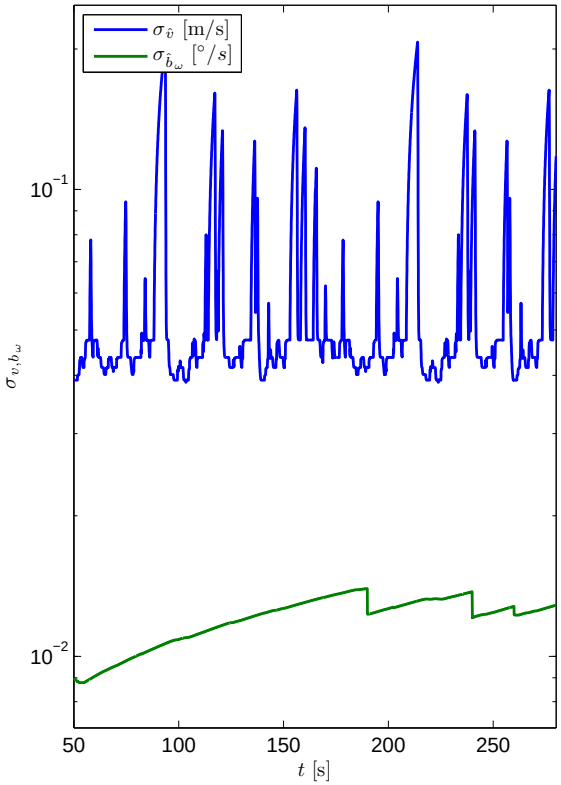
The SLAM filter performance can be evaluated through Fig. 5.5 and Fig. 5.6. The former shows the estimation error of the vehicle variables, the linear velocity on Fig. 5.5(a) and the measurement bias on Fig. 5.5(b), and the standard deviation of these two estimates, which is presented in Fig. 5.5(c). Note that $\sigma_{(.)}^2 = \operatorname{tr}(\mathbf{\Sigma}_{(.)})$. The latter depicts the evolution of the estimation of 10 sensor-based landmarks, with Fig. 5.6(b) showing the standard deviation of each landmark growing to about 1 m, when a loop closure is triggered at $t = 190$ s, and the uncertainty diminishes considerably. The same can be said about the estimation error, presented in Fig. 5.6(a), that is maintained at a very low level while the landmarks are observed ($t \in [50, 80]$ s), then diverges just to be reduced drastically after the loop closing event.



(a) The velocity estimation error.

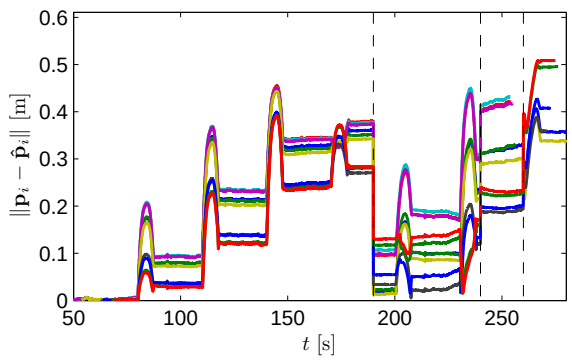


(b) The angular measurement bias estimation error.

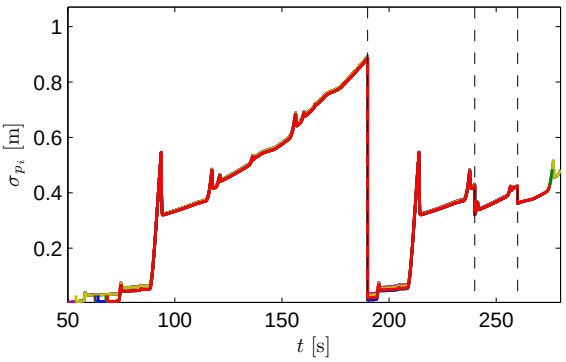


(c) The standard deviation of the vehicle state.

Figure 5.5: Time evolution of the vehicle related variables.



(a) The estimation error.



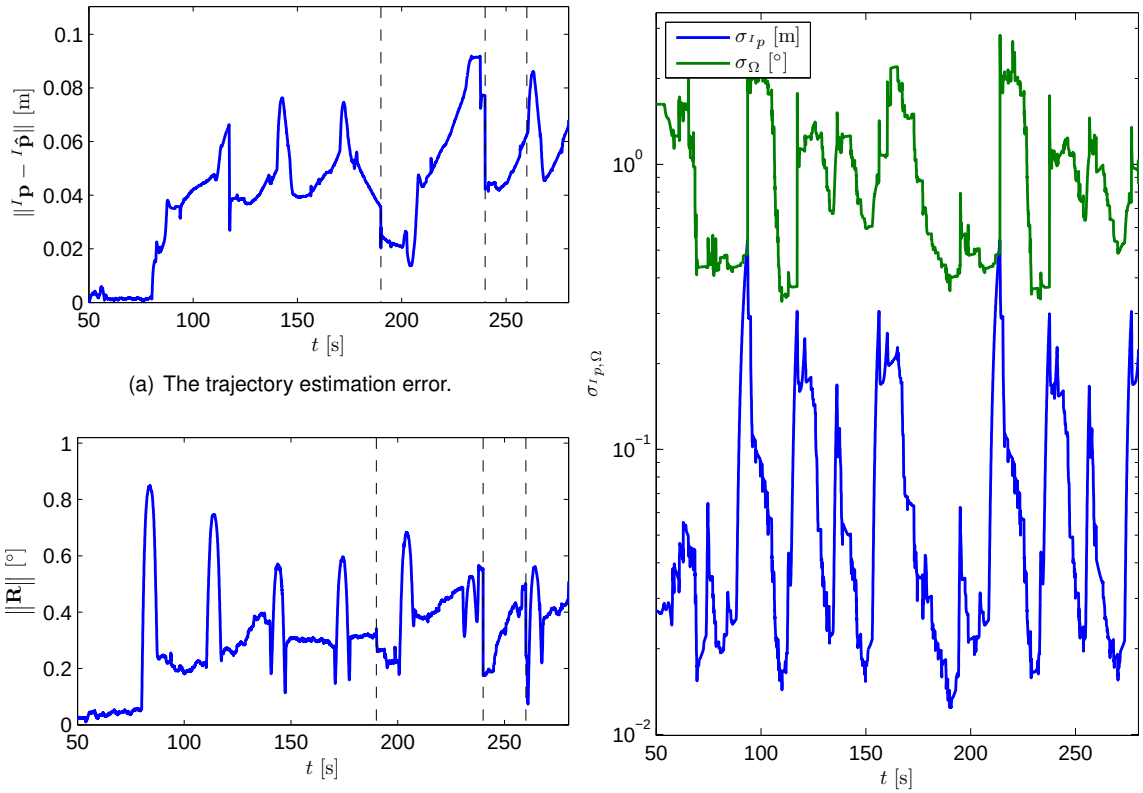
(b) The standard deviation.

Figure 5.6: Estimation error and standard deviation of 10 sensor-based landmark estimates. Vertical dashed lines indicate loop closing events.

5.5.2 *esTIMATE* algorithm

The performance of the inertial map and trajectory estimation can be assessed by observation of Fig. 5.7 and Fig. 5.8. The estimation error of the vehicle trajectory, presented in 5.7(a), is kept in a small absolute value of under 10 centimeters. In Fig. 5.7(b), the estimation error of the rotation matrix is expressed by the quantity $\left\|\tilde{\mathbf{R}}_k\right\| = \arccos\left(\frac{1}{2}\left(\mathrm{tr}\left(\hat{\mathbf{R}}_k{}^T\mathbf{R}_k\right) - 1\right)\right)$ that can be seen as the distance between the true and estimated rotation matrices in $\mathrm{SO}(3)$. There, it is possible to observe that this rotation error is also very small, always below 1°. Note that the periodic variations that can be found

in most of these error figures coincide with the passage of the vehicle through the corners of the map, where rapid changes in the angular velocity occur. Figure 5.8 depicts the estimation error (Fig. 5.8(a)) and standard deviation (Fig. 5.8(b)) of the same 10 landmarks picked out for Fig. 5.6, this time in the inertial frame. Note that the uncertainty never increases, as mentioned in Subsection 3.4.

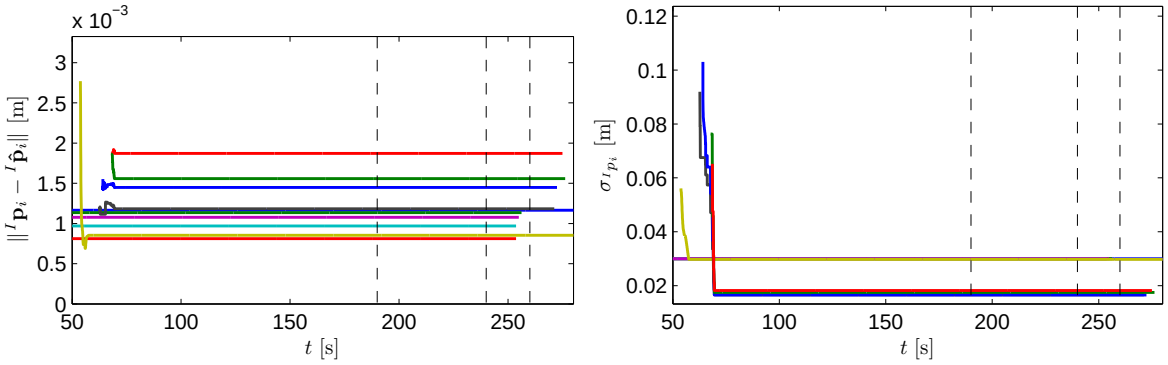


(a) The trajectory estimation error.

(b) The orientation estimation error.

(c) The standard deviation of the vehicle state.

Figure 5.7: The estimation error and standard deviation of the pose estimate. Vertical dashed lines indicate loop closing events.



(a) The estimation error.

(b) The standard deviation.

Figure 5.8: Time evolution of the 10 inertial landmark estimates corresponding to those of Fig. 5.6.

5.5.3 Performance demonstration

The statistics regarding the number of landmarks can be found in Fig. 5.9: in blue, the number of landmarks in the filter state, in green the number of landmarks used for inertial estimation (only a recent set), and in red the number of observed landmarks . The stems represent loop closure trials and events, and the dashed line the minimum number of landmark associations for loop closing.

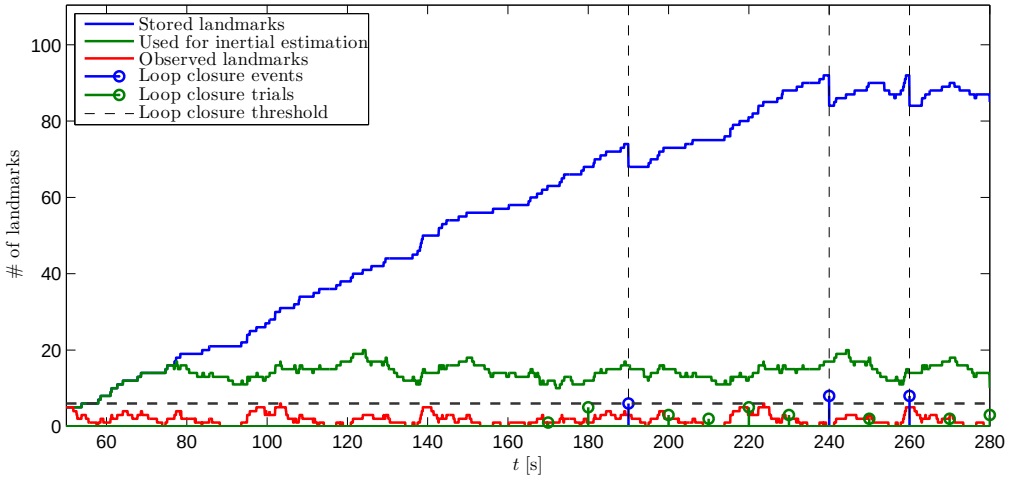
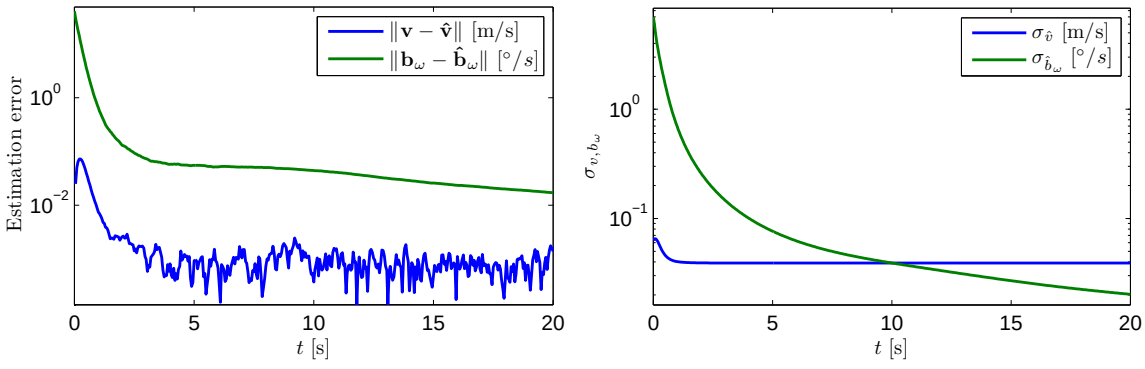
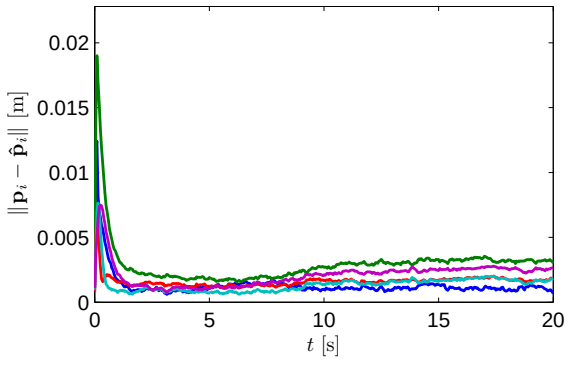


Figure 5.9: Evolution of the number of landmarks used in the Kalman filter (in blue), in the inertial map and trajectory estimation (in green) and observed. Loop closure trials, threshold and events also present.

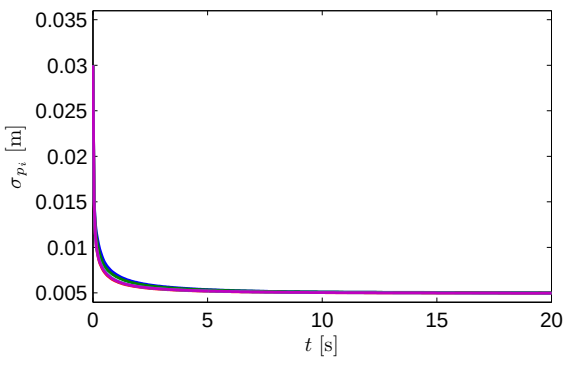


(a) The estimation error of the vehicle variables.

(b) The standard deviation of the vehicle variables.



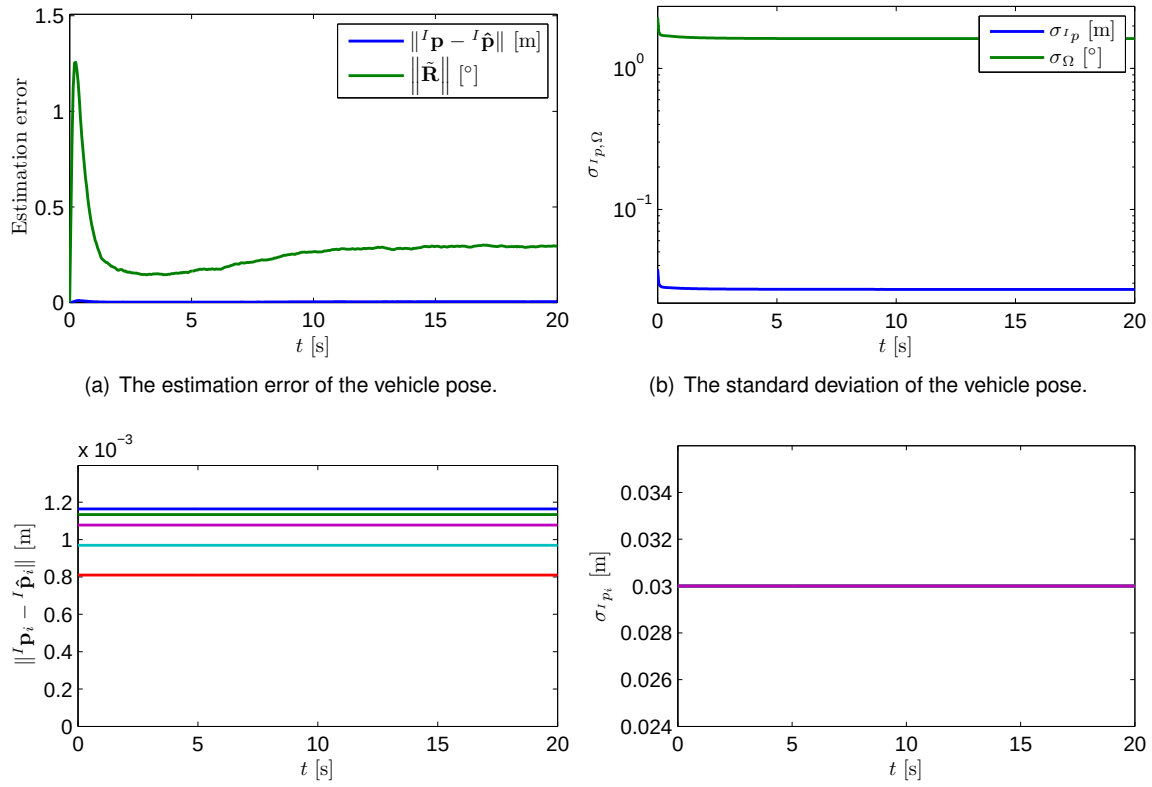
(c) The estimation error of the first landmarks.



(d) The standard deviation of the first landmarks.

Figure 5.10: Convergence of the Sensor-based SLAM filter.

For demonstration of the convergence of the algorithm when the observability conditions are satisfied, Fig. 5.10 is included. In these first 20 seconds of simulation, the vehicle is immobilized on the ground, and 5 different landmarks are visible. In Fig. 5.10(a) and Fig. 5.10(c), respectively the estimation error of the vehicle variables and of the first landmarks can be seen to converge, as does the respective standard deviation in Fig. 5.10(b) and Fig. 5.10(d).



(a) The estimation error of the vehicle pose.

(b) The standard deviation of the vehicle pose.

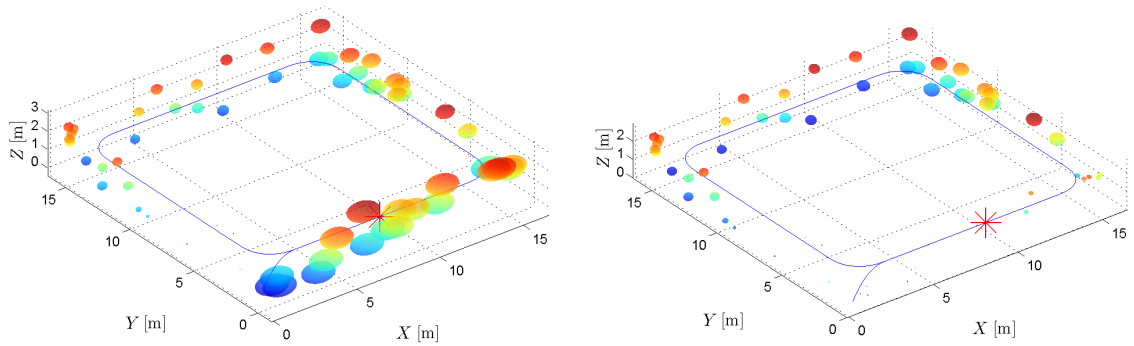
(c) The estimation error of the first landmarks.

(d) The standard deviation of the first landmarks.

Figure 5.11: Convergence of the *esTIMATE* algorithm.

The same can be said of the products of the optimization algorithm, with the estimation error (Fig. 5.11(a)) and standard deviation (Fig. 5.11(b)) of the vehicle pose variables converging. As the update of the inertial landmarks is only performed if the uncertainty diminishes, its standard deviation will either be constant or converging at all times.

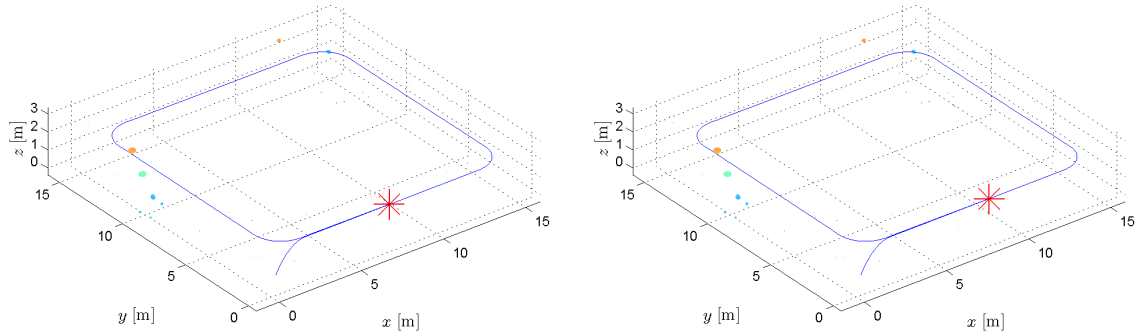
Finally, estimated maps of the environment can be found in Fig. 5.12, the sensor-based versions, and Fig. 5.13, the inertial ones. In these figures, the estimated maps and trajectory of the vehicle are depicted. The coloured ellipsoids represent the 95% uncertainty (2σ) of each landmark and the red star denotes the position of the vehicle at the time. Moreover, the sensor-based maps are rotated and translated using the true quantities, and the trajectory there shown is also the true one. On the other hand, the inertial maps of Fig. 5.13 have the estimated trajectory and current position of the vehicle. The loop closure effects on the uncertainty can be examined in the sensor-based maps of Fig. 5.12. There the difference between a map just before a loop closure event and a map immediately after is obvious. Note that the older landmarks have greater uncertainty, and the landmarks closer to the position of the vehicle, which are the ones more affected by the loop closure, have low uncertainty.



(a) Map at $t = 189$ s, 1 second before a loop closure.

(b) Map at $t = 191$ s, 1 second after a loop closure.

Figure 5.12: The sensor-based estimated maps before and after a loop closure.



(a) Map at $t = 189$ s, 1 second before a loop closure.

(b) Map at $t = 191$ s, 1 second after a loop closure.

Figure 5.13: The inertial estimated maps before and after a loop closure.

This simulation was designed to allow the practical validation of the consistency of the algorithm, by exposing the vehicle to previously visited terrain after exploring new areas, in order to trigger a loop closing. The results show that the sensor-based map is consistent, allowing the loop to be closed repeatedly (see Fig. 5.9). Moreover, the simulation results here presented demonstrate that the uncertainty is coherent with the estimation errors, as shown in Fig. 5.6 and Fig. 5.5.

6

Experimental Results

THE ultimate goal of any scientific work with a practical emphasis is to be able to be employed in real world scenarios. In the field of algorithm development, it is very important to validate any algorithm experimentally, even if the simulation results are satisfactory and made use of reality-like simulated environments.

The simulation results were consolidated by a preliminary experiment, which took place during a short visit to the Sensor-Based Cooperative Robotics Research Laboratory - SCORE Lab of the Faculty of Science and Technology of the University of Macau.

The process of preparing the experiments is detailed, by describing the experimental setup and the equipment employed, as well as the necessary technical details. The chapter also presents the results of a preliminary experiment, comparing them with the available ground truth data.

This chapter is organized as follows. Section 6.1 describes the experimental setup in all its components. In Section 6.2 the results of a preliminary experiment are presented and discussed.

6.1 Experimental Setup

Figure 6.1: The SCORE Lab.

The facilities of the SCORE Lab consist of a $6\mathrm{m}\times6\mathrm{m}\times3\mathrm{m}$ room, approximate dimensions, with a usable area of around 16 m^2, see Fig. 6.1. The room is equipped with a *VICON*® motion capture system, which provides accurate estimates of the position, attitude, and linear and angular velocities of any vehicle placed inside the working area with the correct markers. In terms of mobile robots, the lab has available several small Blade® quadrotors and one *AscTec*® *Pelican* quadrotor. For the purpose of this experiment, the *AscTec*® *Pelican* was instrumented with a *Microsoft Kinect*™ RGB-D camera, for the extraction of visual features of the room, and a *Microstrain 3DM-GX3-25* inertial measurement unit. The objective of the experiment was to assess the SLAM algorithm with real data, resorting to the ground truth data provided by the *VICON*® system. For that purpose, the quadrotor was equipped with several *VICON*® markers. The following subsections detail the equipment utilized.

6.1.1 *VICON*® Motion Capture System

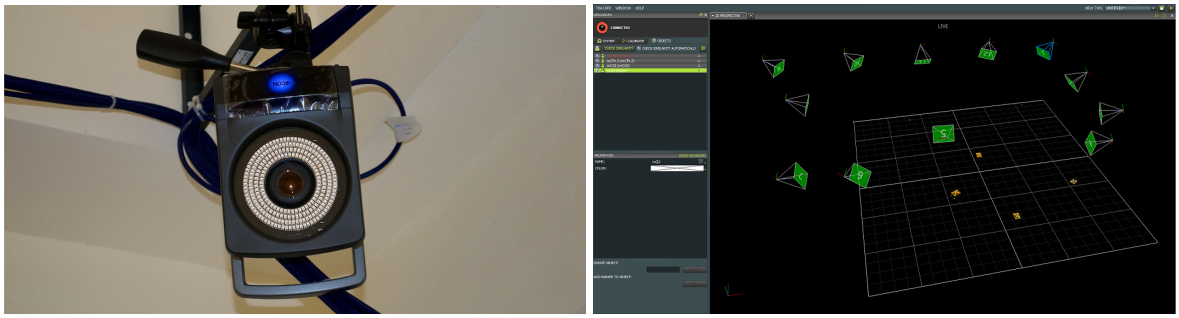
The *VICON*® Motion Capture System [36], as its name suggests, is a state-of-the-art optical system that records the movement of objects or people with millimetric resolution in 3-D, by means of infrared marker-tracking. Its primary uses involve military, entertainment[1], sports, medical, computer vision, and robotic applications. For the last two, it provides a very good basis for validation. The high-accuracy of the tracking achieved by *VICON*®, see [37] for details, enables its use as ground truth for experimental validation of positioning and attitude estimation algorithms.

Optical motion capture uses a series of high-resolution, high-rate, infrared cameras setup around a working area, viewing the scene from a variety of angles[2]. Reflective markers are placed on the object to be tracked, and are recognized by the software, which then combines the information of the full set of cameras to generate a tridimensional picture of the markers[3]. Identifying a set of matrices as a rigid body allows to track the object, and the placement of the markers has to be chosen carefully, in order to maximize the performance of the capture.

The *VICON*® instance present in the SCORE Lab is capable of acquiring more than 100 frames per second, providing the position, attitude, and linear and angular velocities of the objects therein in

[1] http://www.awn.com/news/technology/ilm-employs-vicon-motion-capture-avengers
[2] http://physbam.stanford.edu/cs448x/old/Optical_Motion_Capture_Guide.html
[3] http://www.cs.utah.edu/~halzahaw/MotionCapture_main.html

that fast rate. The latter quantities are obtained by finite differences of the former, and are thus less smoother than the fundamental quantities that determine the pose of the object. An example of the *VICON*® cameras used and a screenshot of the proprietary software can be found in Fig. 6.2.



(a) A *VICON*® camera. (b) The *VICON*® software.

Figure 6.2: The *VICON*® system of the SCORE Lab.

6.1.2 The instrumented *AscTec*® *Pelican*

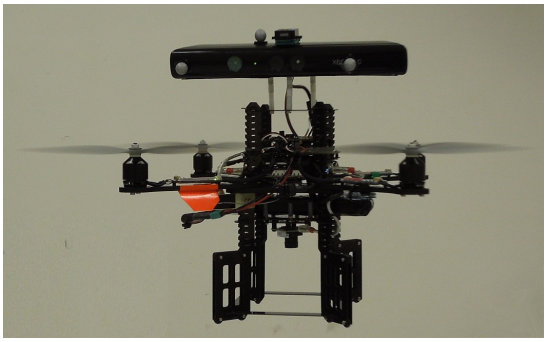


Figure 6.3: The *AscTec*® *Pelican* equipped with a *Microsoft Kinect*™, a *Microstrain 3DM-GX3-25* and *VICON*® markers.

The preliminary experiments detailed in this chapter were conducted, in a hand-driven fashion, using as platform the quadrotor of Fig. 6.3, equipped with an Intel Atom 1.6 GHz processor with 1 GB DDR2 soldered RAM, and into which was added a *Microstrain 3DM-GX3-25* inertial measurement unit working at 200 Hz and a *Microsoft Kinect*™ camera, at 30 Hz. The *AscTec*® *Pelican* is a very versatile platform with the following characteristics[4]:

- Minimum take-off weight of 630 g;
- Maximum payload of 650 g;
- Maximum flight time with full payload of 15 minutes.

The *Microsoft Kinect*™ is a motion sensing device that consists in fact in a depth sensor with 11-bit VGA resolution and a RGB camera with 8-bit VGA resolution. The depth sensor is composed by an

[4] `http://www.asctec.de/uav-applications/research/products/asctec-pelican/`

infrared pattern projector combined with a monochrome CMOS sensor, i.e., an active pixel sensor. Figure 6.4[5] shows a small schematic of the components of this device.

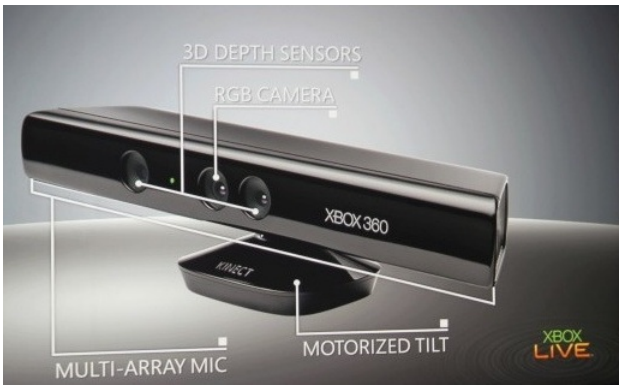


Figure 6.4: A diagram of the technologies in the *Microsoft Kinect*^{TM}.

In the internal board of the quadrotor a version of the Linux distribution Ubuntu was installed, and on top of that, the Robot Operating System version Fuerte (ROS[6]), a framework for robot software development that provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management within other features, see [38] for further details. A driver for the inertial measurement unit developed by the DSOR Lab team of the Institute for Systems and Robotics was employed, as well as community-based drivers[7] and software[8] for interfacing with the *Microsoft Kinect*^{TM}. The experimental setup is represented schematically in Fig. 6.5. A stripped-down version of the interfacing software was ran on the quadrotor's Atomboard, in order to process the raw data of the camera into a coloured image, as well as the corresponding depth map.

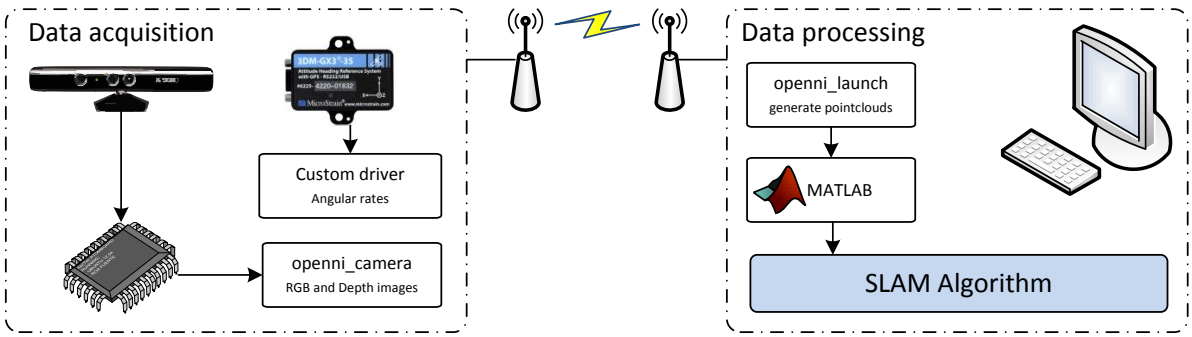


Figure 6.5: The experimental setup.

6.1.3 Algorithm parameters

Although the combination of parameters utilized in the simulation of the previous chapter was chosen according to physical aspects and with the objective of controlling the evolution of the uncer-

[5] Taken from a slide from Microsoft's E3 Conference at `http://en.wikipedia.org/wiki/File:KinectTechnologiesE3.jpg`
[6] `http://www.ros.org`
[7] `http://www.ros.org/wiki/openni_camera`
[8] `http://www.ros.org/wiki/openni_launch`

tainty of the non-visible landmarks, it must be adapted to the real situation. For example, the output noise covariance was tuned through the following process. Of a set of landmarks obtained through the landmark extraction process, a landmark was followed for a period of time with the *Microsoft Kinect*™immobilized. This allowed to obtain the standard deviation of the measurements. Other parameters were readjusted, also taking into account the level of noise of the angular measurements. This systematic process yielded the following set of parameters for the Kalman filter:

- the output noise covariance is $\mathbf{\Theta}_k = \sigma_\theta^2 \mathbf{I}_3$, $\sigma_\theta = 0.022\ \mathrm{m}$;
- the state disturbance covariance is given by $\mathbf{\Xi}_k = T_s\, \mathrm{diag}\left(\sigma_v^2 \mathbf{I}_3, \sigma_{b_\omega}^2 \mathbf{I}_3, \sigma_{p_1}^2 \mathbf{I}_3, \ldots, \sigma_{p_N}^2 \mathbf{I}_3\right)$, with $\sigma_v = 0.05\ \mathrm{m/s}$, $\sigma_{b_\omega} = 10^{-5}\ \mathrm{rad/s}$ and $\sigma_{p_i} = 0.1\ \mathrm{m}$;
- the initial state covariance was set to $\mathbf{\Sigma}_0 = \mathrm{diag}\left(\sigma_{v_0}^2 \mathbf{I}_3, \sigma_{b_{\omega_0}}^2 \mathbf{I}_3\right)$ with $\sigma_{v_0} = 0.039\ \mathrm{m/s}$ and $\boldsymbol{\sigma}_{b_{\omega_0}} = 0.02\ \mathrm{rad/s}$;
- and $\mathbf{\Sigma}_{p_{i_0}} = \sigma_{p_{i_0}}^2 \mathbf{I}_3$, $\sigma_{p_{i_0}} = 0.039\ \mathrm{m}$ for any new landmark.

The initial estimate of the velocity is set to zero, and for the initial condition of the measurement bias it is chosen the average of a preliminary acquisition before starting the filter. The most significant change in parameters occurred in the SLAM-specific set of parameters, due to intrinsic aspects of the experiment: a smaller loop and environment, meaning that most of the landmarks will be visible for a longer time. Thus, the recent landmark set is composed by landmarks seen in the last half second, and the old by landmarks not seen in more than 15 s. No loop closing was tried, because of the small amount of time the experiment lasted.

6.2 Preliminary experimental results

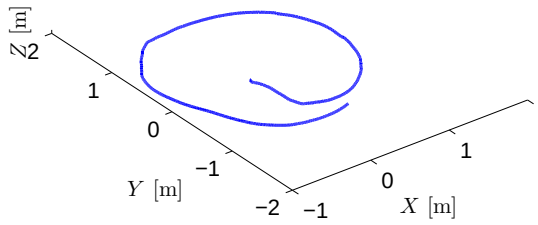
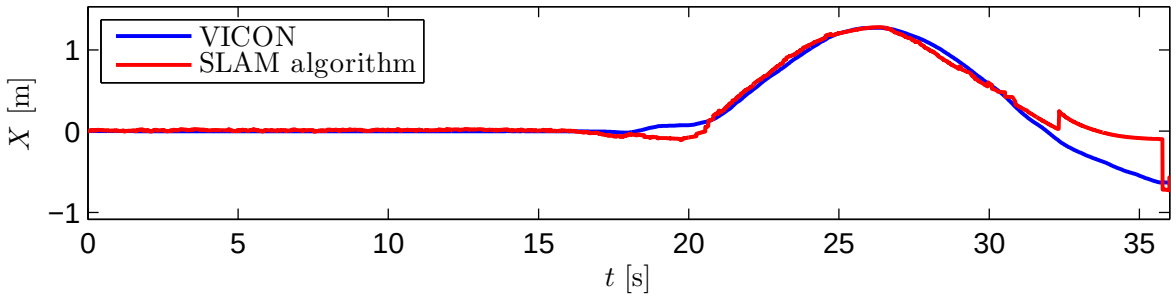


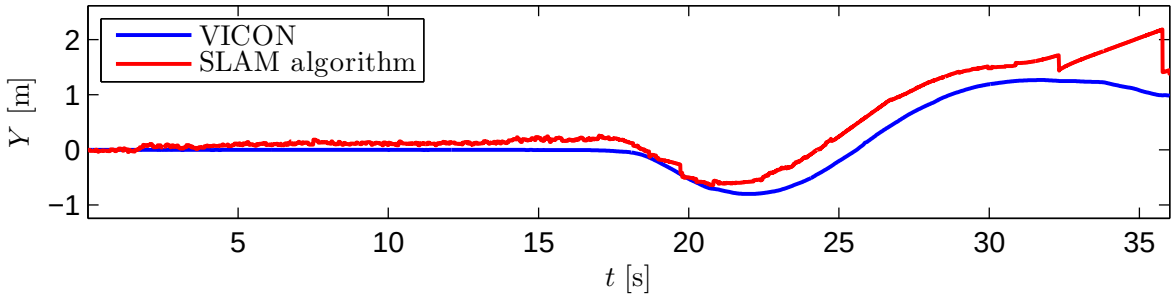
Figure 6.6: The trajectory followed by the vehicle.

The experiment consisted of moving the quadrotor hand-driven around the room described in the previous section, in the circular-like trajectory shown in Fig. 6.6. The maximum speed was 0.63 $\mathrm{m/s}$, and its average was 0.32 $\mathrm{m/s}$. Due to technical problems with the equipment, namely the loss of RGB-D frames, the data usable was reduced to the first 36 seconds, of which around 15 correspond to a period when the vehicle was stopped in order to let the filter converge. The results are presented in the following subsections. Firstly, the variables where ground truth is available, followed by the remaining products of the Kalman filter. The inertial estimates are presented next, ahead of the final subsection, where obtained maps are shown.

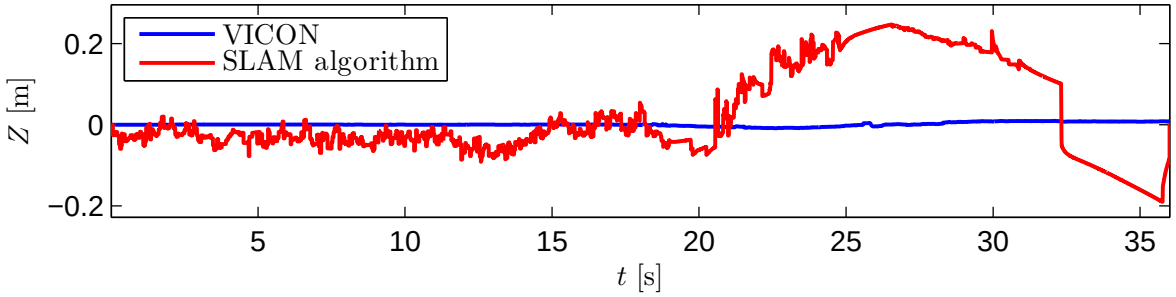
6.2.1 Ground Truth



(a) X coordinate.



(b) Y coordinate.

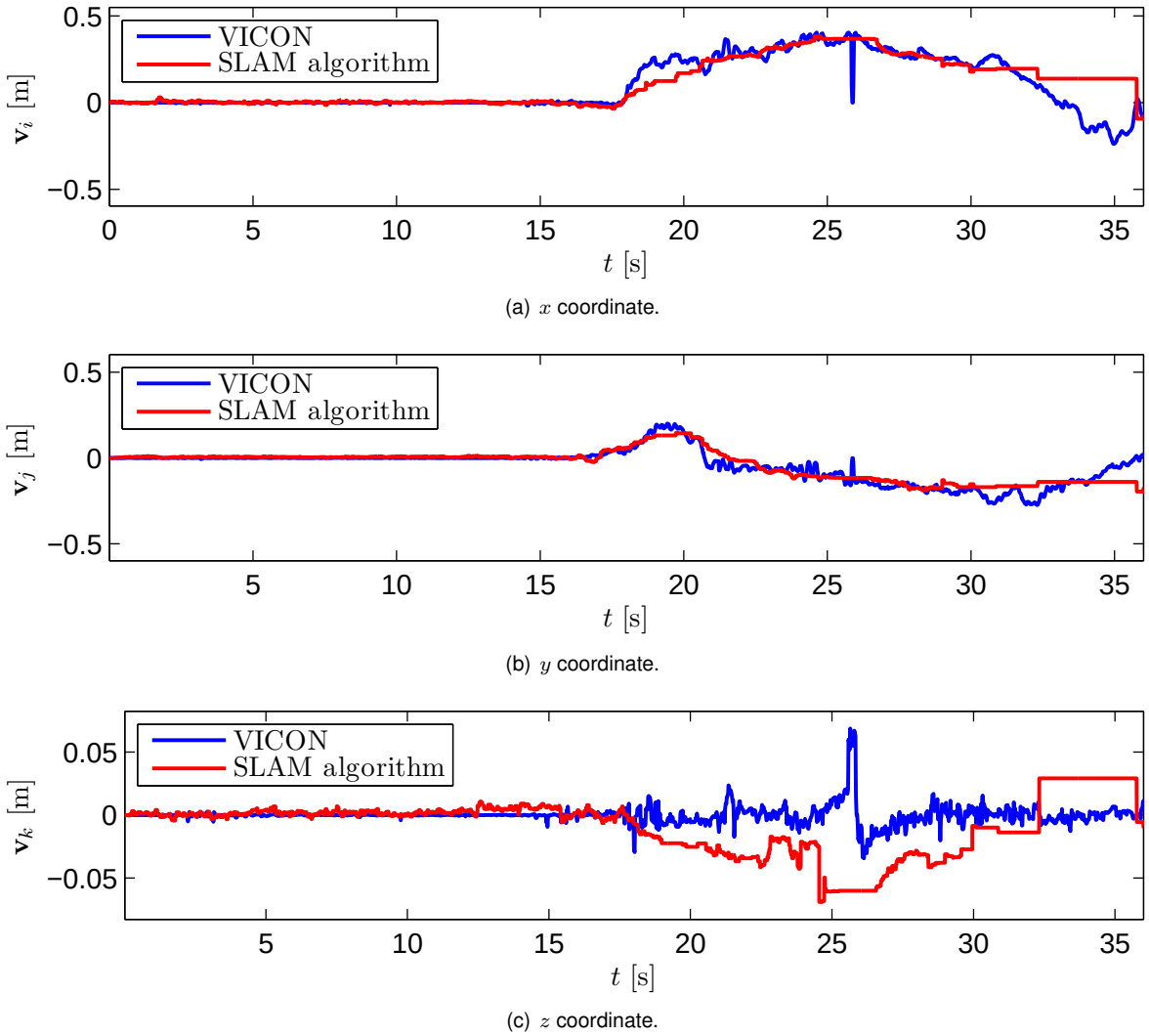


(c) Z coordinate.

Figure 6.7: Time evolution of the real and estimated trajectory

The behaviour of the SLAM algorithm, including both the SLAM filter and the *esTIMATE* algorithm in sequence, can be evaluated in this subsection. Figure 6.7 depicts the estimated trajectory, in red, against the VICON estimation in blue, here taken has ground truth, broken down in the three coordinates for better visualization. It can be seen that the estimation follows very closely the true trajectory of the vehicle, never being more than 20 centimeters off, except after the first 30 seconds, where less observations were available as will be seen in Fig. 6.16. Figure 6.8 compares the ground truth with the estimation of the body-fixed linear velocity of the vehicle. The former is obtained by rotating the inertial estimate with the rotation, both provided by VICON. Again, the estimation follows within reasonable accuracy the ground truth, and, once more, it worsens after the 30 seconds, for the reasons explained. Finally, to demonstrate the good estimation of the vehicle attitude, consider Fig. 6.9. There the body-fixed velocity estimation is rotated with the estimated rotation matrix to obtain an estimate of the inertial velocity, which is then compared with the VICON estimate, to show once more

the good performance of the vehicle. Note that the VICON estimates of the velocity are obtained by differentiation of the position, thus being somewhat noisy.



(a) x coordinate.

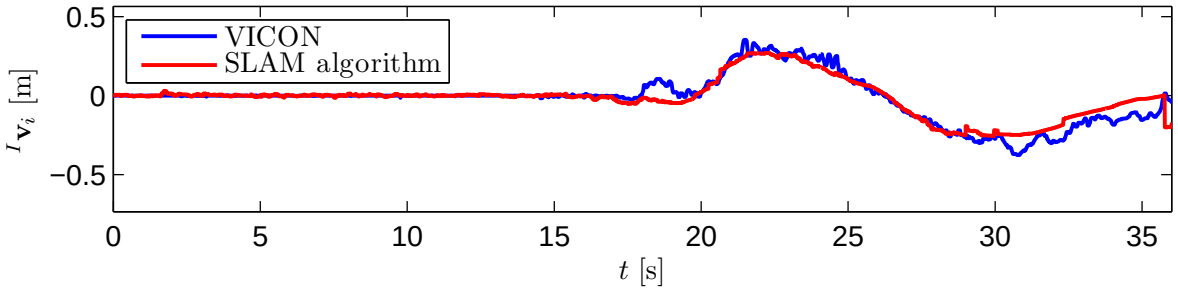
(b) y coordinate.

(c) z coordinate.

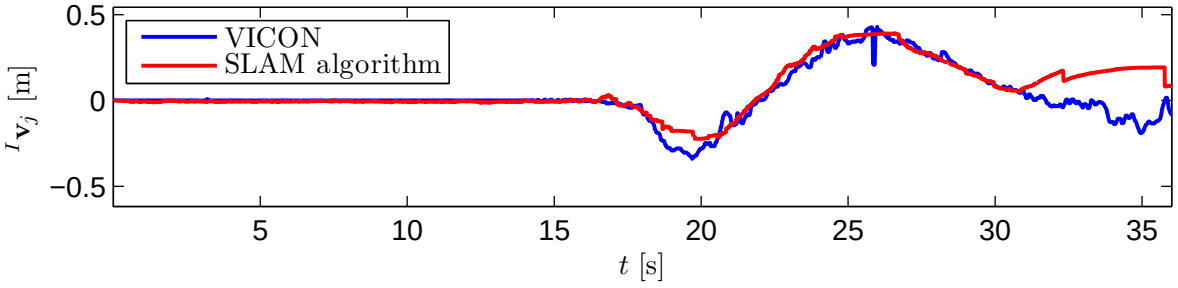
Figure 6.8: Time evolution of the real and estimated velocity in the body-fixed frame.

6.2.2 Sensor-based SLAM Filter

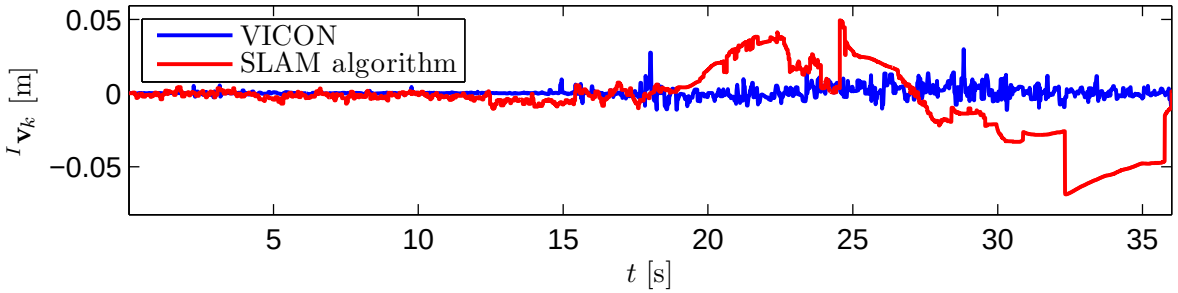
The remaining product of the Sensor-based SLAM filter is the angular measurement and the set of landmarks. The first is depicted in Fig. 6.10. Recall that the filter was initialized with the average of a pre-acquisition, and therefore this estimate doesn't start on zero. The uncertainty of the vehicle variables and of the first 5 landmarks is shown on Fig. 6.11 and Fig. 6.12 respectively. The standard deviation of the linear velocity estimate is shown in blue, and that of the measurement bias estimation in green.



(a) X coordinate.



(b) Y coordinate.



(c) Z coordinate.

Figure 6.9: Time evolution of the real and estimated velocity in the inertial frame.

6.2.3 *esTIMATE* algorithm

The results of the Inertial Map and Trajectory estimation have been partially presented in Subsection 6.2.1 compared with the ground truth, but the uncertainty of the pose and of the inertial map estimates is still to be presented. Figure 6.13 depicts the uncertainty associated with the pose of the vehicle, the position standard deviation in blue, and the orientation standard deviation in green. The uncertainty of the inertial landmarks corresponding to those in Fig. 6.12 is exposed in Fig. 6.14.

6.2.4 Environment maps

An example of two different maps of the environment is shown in Fig. 6.15: a sensor-based version in Fig. 6.15(a), and and inertial map along with the trajectory of the vehicle thus far (blue line) in Fig. 6.15(b), with the respective uncertainty ellipsoids in colour. Note that the sensor-based and inertial maps are not necessarily related by a rotation and translation, as the inertial landmarks estimates may come from different iterations –they are only updated if their uncertainty decreases, while the

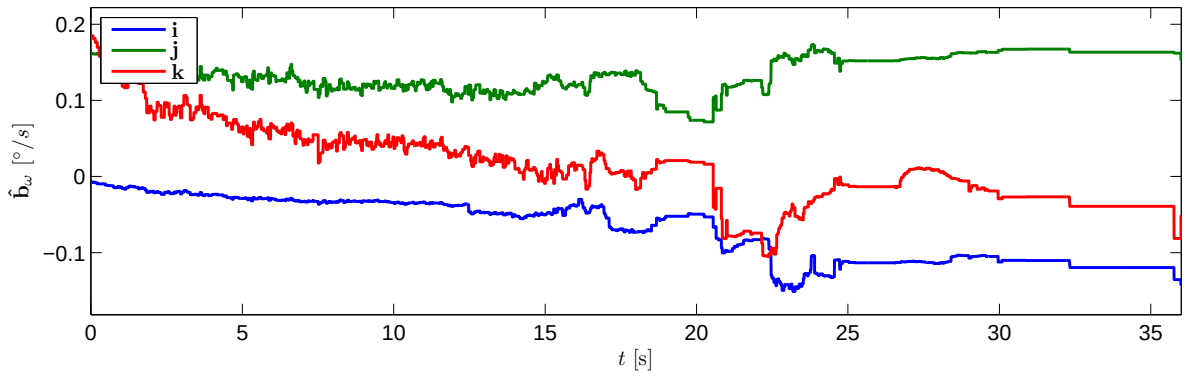


Figure 6.10: The estimation of the angular measurement bias.

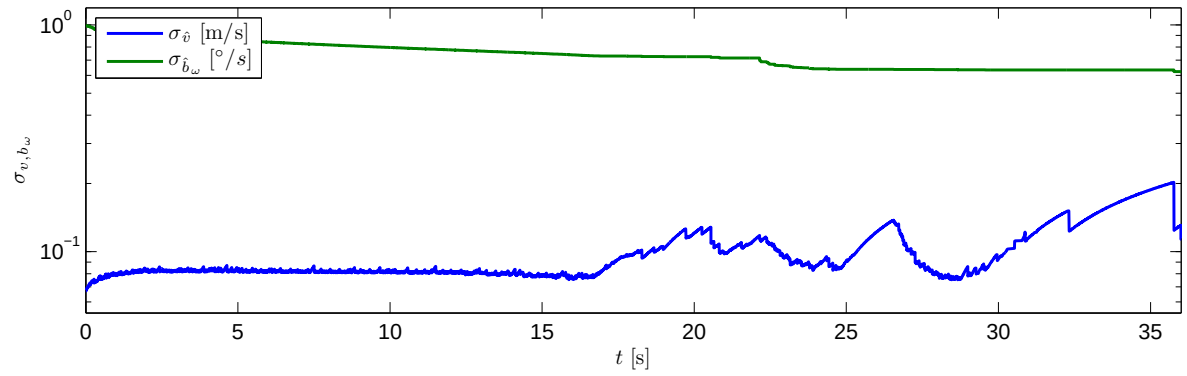


Figure 6.11: The standard deviation of the estimated vehicle variables: the linear velocity and angular measurement bias.

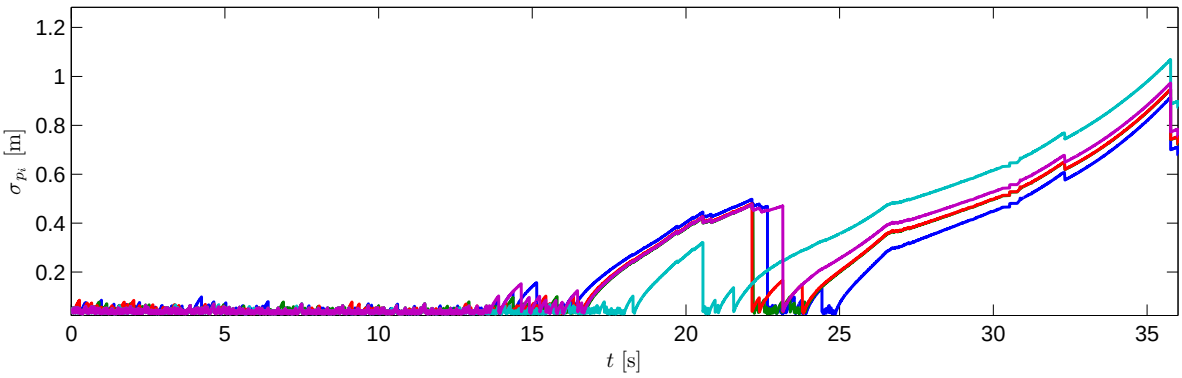


Figure 6.12: Time evolution of the uncertainty of 5 landmarks.

sensor-based ones are updated in every iteration, even if only propagated in open loop.

6.2.5 Concluding remarks

Finally, the evolution of the number of landmarks involved in the algorithm are shown in Fig. 6.16. In blue, the number of landmarks in the SLAM filter state is presented, and, in green, the number of landmarks used by the *esTIMATE* algorithm to compute the optimal pair $\left(\hat{\mathbf{R}}_k, {}^I\hat{\mathbf{p}}_k\right)$. Note that

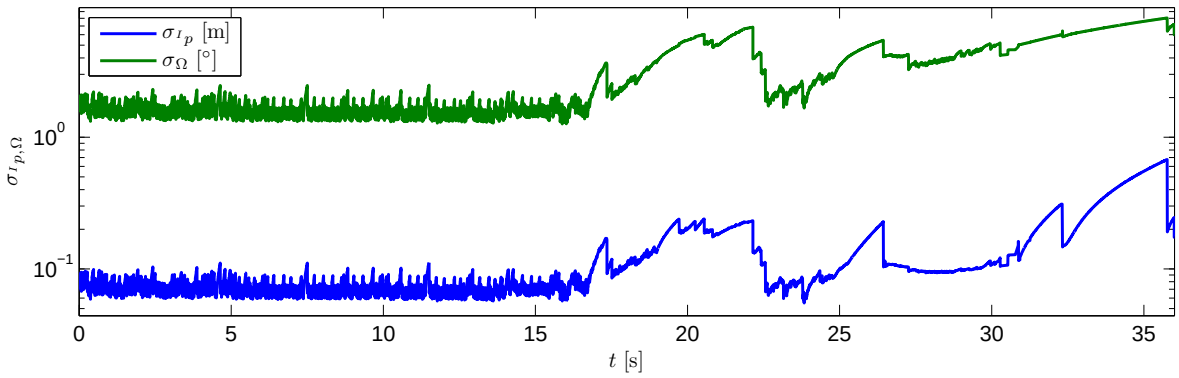


Figure 6.13: The standard deviation of the rotation and translation estimates.

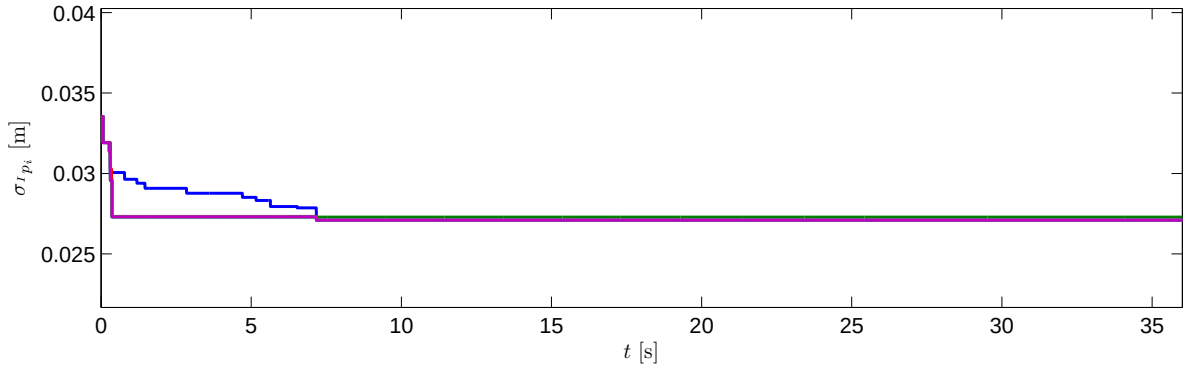
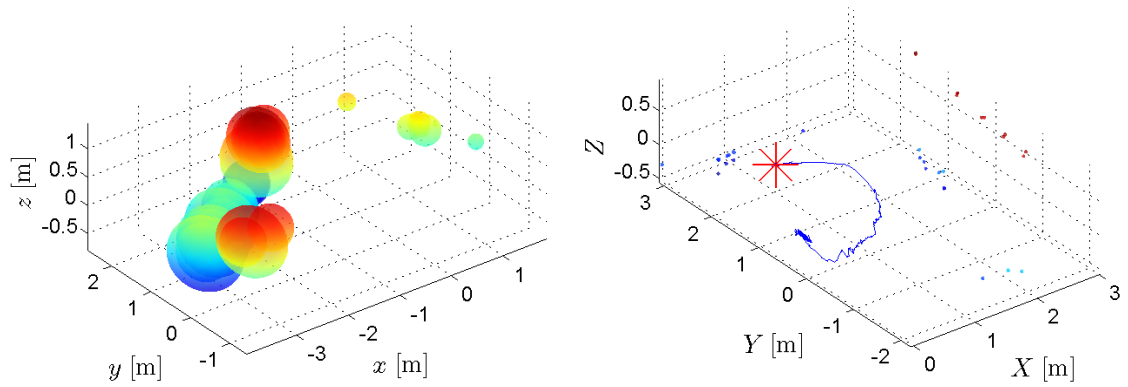


Figure 6.14: Time evolution of the uncertainty of 5 inertial landmarks.



(a) Sensor-based map (b) Inertial map and trajectory estimate

Figure 6.15: The sensor-based and inertial estimated maps 30 seconds after start.

these numbers are constant while the vehicle is stopped. The number of visible landmarks in each observation instant is shown in red. It is worth noticing that, after the first 30 seconds, the refresh rate of the observations is reduced drastically, due to the referred technical issues with the equipment, while the number of landmarks observed is also small. This explains the degradation in the estimation that occurs around that time.

However limited this experience is, the presence of ground truth showed the good performance of

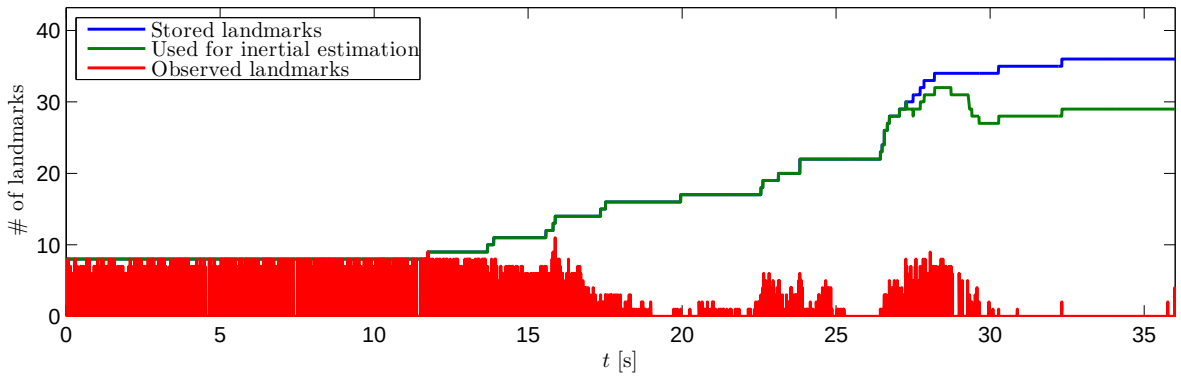


Figure 6.16: Evolution of the number of landmarks used in the Kalman filter (in blue), in the inertial map and trajectory estimation (in green) and observed.

the individual parts of the algorithm, as well as the behaviour of the algorithm as a whole in a real situation. Moreover, the process chosen to obtain landmark measurements effectiveness was also demonstrated, both through the repeatability of the SURF algorithm and the quality of the pointclouds that the software employed provides by combining the coloured and depth images that the *Microsoft Kinect*^{TM}acquires.

7

Conclusions

THIS thesis presented a novel algorithm for Simultaneous Localization and Mapping, reporting the design, analysis, implementation, and validation stages. The work proposed has various contributions, including a novel sensor-based filter and an uncertainty-wise fully characterized optimization orthogonal Procrustes problem. The main idea behind the first work was the development of a formal proof of stability and convergence of the filter. This was accomplished by designing the filtering framework in the sensors space, which enabled the construction of a nonlinear system that can be regarded as linear time-varying, thus allowing the use of the well-established Kalman filter theory. The sensor-based approach, used in [3], [21], [1], and [18], allows the establishment of purely kinematic systems, as well as the direct use of the measurements in the state, especially as the sensor chosen outputs directly Cartesian coordinates. Also, being body-fixed, the pose of the vehicle in that algorithm is deterministically defined by the origin and attitude of the frame. This was the subject of Chapter 2.

The second part of this work addressed the orthogonal Procrustes problem and its statistical characterization. An optimization problem was defined, using the sensor-based map and its inertial counterpart. This optimization problem was built upon the works in [2], [27] and [19], introducing novelty in the sense that individual weights were introduced to model the uncertainty in each of the landmarks used to define the problem. Moreover, the previous works pursuing a statistical description of the problem, such as [28], [29], and [30], have used limiting assumptions, including small rotations, no weights or equal covariances for all landmarks. The novelty of the description here proposed is the inclusion of

individual weights and covariances, and arbitrary translations and rotations. This work was addressed in Chapter 3.

Albeit the underlying purpose of the conception of this work was the use of both algorithms in sequence, the design of each one of these algorithms was executed completely in separate, as they may be used independently, especially the sensor-based SLAM filter that is perfectly capable of working standalone, provided that its use does not require inertial estimates. On the other hand, the second algorithm cannot work without the first – or at least something that provides a sensor-based map and its associated uncertainty. The implementation details of the complete algorithm were addressed in Chapter 4, as well as a description of auxiliary algorithms and practical implementation issues.

Validation of the algorithm was performed: In a simulated setting, full validation was achieved, by means of a demonstration of the algorithm performance in all its extension, particularly by showing that it is capable of handling uncertainty high enough to allow the closure events to occur in a loop of nearly 50 meters, thus proving the consistency of the mapping. Furthermore, the theoretical results were verified, as the convergence of the error dynamics of the LTV Kalman filter was demonstrated in the conditions proved sufficient. Furthermore, a preliminary experiment was set up resorting to an instrumented quadrotor, and, the good performance of the algorithm was evidenced by the comparison of the estimates with the available ground truth data. Technical problems prevented the full test of the algorithm, namely loop closure.

7.1 Future Work

Although very thorough, the work presented in this thesis has still large room for improvement. Starting with the theoretical work, only sufficient conditions for the observability and correspondence of the LTV system with the nonlinear system were found. Therefore, future work will include finding the necessary conditions, either by proving that the sufficient are indeed necessary, or by deriving new conditions altogether. Moreover, the inclusion of landmark directions in the observability analysis will also be addressed.

In the implementation side, the inclusion of landmark directions would require a change in the landmark detection process, which can also be addressed futurely. Furthermore, the SURF algorithm provides information that is currently being discarded. For instance, the feature descriptor, see [23] for more information, may help improve the association process if availed. Thus, there is possible future work in that direction too. Also, the experimental setup has to be reassessed, in order to avoid the technical problems that arose, in the future experimental validation that is recommended for scientific dissemination. Finaly, the real time optimization of the algorithm is of paramount importance for achieving a truly online filter, thus paving the way for the use of small autonomous unmanned aerial vehicles.

Bibliography

[1] P. Batista, C. Silvestre, and P. Oliveira, "Single range aided navigation and source localization: Observability and filter design," *Systems & Control Letters*, vol. 60, no. 8, pp. 665 – 673, 2011.

[2] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[3] P. Batista, C. Silvestre, and P. Oliveira, "On the observability of linear motion quantities in navigation systems," *Systems & Control Letters*, vol. 60, no. 2, pp. 101 – 110, 2011.

[4] R. Brockett, *Finite dimensional linear systems*. Series in decision and control, John Wiley & Sons, 1970.

[5] K. Čapek and C. Novack, *R.U.R. (Rossum's Universal Robots)*. Penguin Classics, Penguin Books, 2004.

[6] W. Schmidt and H. of Alexandria, *Pneumatica et Automata*. Bibliotheca scriptorum Graecorum et Romanorum Teubneriana, B.G. Teubner, 1899.

[7] M. Rosheim, *Leonardo's Lost Robots*. Springer, 2006.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainly," *Int. J. Rob. Res.*, vol. 5, pp. 56–68, Dec. 1986.

[11] J. Castellanos, R. Martinez-Cantin, J. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of EKF-SLAM," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 21 – 29, 2007.

[12] J. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *The International Journal of Robotics Research*, vol. 21, no. 4, p. 311, 2002.

[13] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, Oct. 1997.

[14] A. Doucet, N. d. Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI '00, (San Francisco, CA, USA), pp. 176–183, Morgan Kaufmann Publishers Inc., 2000.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598, AAAI, 2002.

[16] S. Thrun, "Simultaneous localization and mapping," in *Robotics and Cognitive Approaches to Spatial Mapping* (M. Jefferies and W. Yeap, eds.), vol. 38 of *Springer Tracts in Advanced Robotics*, pp. 13–41, Springer, 1st ed., 2008.

[17] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006.

[18] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Sensor-based Simultaneous Localization and Mapping - Part I: GAS Robocentric Filter," in *Proceedings of the 2012 American Control Conference*, (Montréal, Canada), pp. 6352–6357, June 2012.

[19] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Sensor-based Simultaneous Localization and Mapping - Part II: Online Inertial Map and Trajectory Estimation," in *Proceedings of the 2012 American Control Conference*, (Montréal, Canada), pp. 6334–6339, June 2012.

[20] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, (St. Paul, MA, USA), May 2012.

[21] P. Batista, C. Silvestre, and P. Oliveira, "Optimal position and velocity navigation filters for autonomous vehicles," *Automatica*, vol. 46, no. 4, pp. 767 – 774, 2010.

[22] B. Anderson, "Stability properties of Kalman-Bucy filters," *Journal of the Franklin Institute*, vol. 291, no. 2, pp. 137 – 144, 1971.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.

[24] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *Robotics and Automation, IEEE Transactions on*, vol. 17, pp. 890 –897, dec 2001.

[25] R. Brown and P. Hwang, *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*. Wiley, 1997.

[26] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley, 2006.

[27] B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society America*, vol. 5, no. 7, pp. 1127–1135, 1988.

[28] C. Goodall, "Procrustes Methods in the Statistical Analysis of Shape," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 2, pp. 285–339, 1991.

[29] R. Sibson, "Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 41, no. 2, pp. 217–229, 1979.

[30] R. Sibson, "Studies in the robustness of multidimensional scaling: Procrustes statistics," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40, no. 2, pp. 234–238, 1978.

[31] C. Evans, "Notes on the OpenSURF Library," Tech. Rep. CSTR-09-001, University of Bristol, January 2009.

[32] D. J. Kroon, "OpenSURF (including Image Warp)." `http://www.mathworks.com/matlabcentral/fileexchange/28300`.

[33] A. J. Cooper, "A Comparison of Data Association Techniques for Simultaneous Localization and Mapping," Master's thesis, Massachussets Institute Of Technology, June 2005.

[34] J. Neira and J. Castellanos, "SLAM: Course Handouts." `http://webdiis.unizar.es/~neira/slam.html`, April 2007.

[35] F. Beer, E. Johnston, and W. Clausen, *Vector Mechanics for Engineers : Dynamics (SI Units)*. McGraw-Hill Higher Education, 7th ed., 2007.

[36] VICON Motion Systems, "Essentials of motion capture v1.2." `http://www.udel.edu/PT/Research/MAL/essentials_of_motion_capture_v1_2.pdf`.

[37] M. Windolf, N. Götzen, and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems - exemplified on the vicon-460 system," *Journal of Biomechanics*, vol. 41, no. 12, pp. 2776–2780, 2008.

[38] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[39] P. Batista, C. Silvestre, and P. Oliveira, "Sensor-based complementary globally asymptotically stable filters for attitude estimation," in *Proceedings of the 48th IEEE Conference on Decision and Control*, (Shanghai, China), pp. 7563–7568, 2009.

[40] H. Lütkepohl, *Handbook of Matrices*. John Wiley & Sons, 1996.

[41] H. Khalil, *Nonlinear Systems*. Prentice Hall, 3rd ed., 2002.

[42] D. Knuth, *The Art of Computer Programming: Fundamental algorithms*. Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley Publishing Company, 1973.

[43] A. J. Stothers, *On the Complexity of Matrix Multiplication*. PhD thesis, University of Edinburgh, 2010.

[44] P. E. Black, “Big-O Notation,” in *Dictionary of Algorithms and Data Structures* (P. E. Black, ed.), U.S. National Institute of Standards and Technology, 2012.

A

Algorithm Complexity

The time complexity of any computational work is a way of quantifying the time each iteration of the algorithm takes to run in function of the size of its input. With this approach, aspects such as the available computational power, that have great influence on the absolute duration, are not taken into account, thus making the analysis more generic. In fact, the measure of time employed is a relative one: an operation that does not depend on the size of the algorithm's input is of unitary order, an operation that depends linearly with the size of the input is of order n, and so forth. This may be expressed in big-O notation (see [44]), which defines asymptotic functions for the true complexity of an operation, i.e., a function $f(n)$ is of order $g(n)$ if there are positive constants a and N, not depending on n, such that $0 \leq f(n) \leq ag(n)$ for all $n \geq N$. Therefore, the case of unitary order is represented by $\mathcal{O}(1)$ and the case of order n is expressed by $\mathcal{O}(n)$.

The complexity of the complete algorithm is described in this appendix. This is performed by analysing each step of the SLAM filter and of the *esTIMATE* algorithm in terms of the basic operations performed: sum, multiplication and inversion of matrices. Table A.1 presents the complexity of each one of these operations, not taking into account possible faster methods to perform them (see [43]).

Matrix Operation	Elements involved	Complexity
Sum	Two $m \times n$ matrices	$\mathcal{O}(mn)$
Multiplication	A $m \times n$ matrix and a $n \times k$ matrix	$\mathcal{O}(mnk)$
Inversion	A $m \times m$ square matrix	$\mathcal{O}(m^3)$

Table A.1: The time complexity of the basic matrix operations in big-O notation.

Consider the landmark sets $\mathcal{I}_{B_k}$, $\mathcal{I}_{cur_k}$, $\mathcal{I}_{old_k}$, $\mathcal{I}_{O_k}$, $\mathcal{I}_{I_k}$ and $\mathcal{I}_{T_k}$. These serve as input and output of the various parts of the algorithm proposed in this dissertation, and it is their corresponding dimensions that influences the complexity of the algorithm.

The analysis is made by dividing the algorithm in its components, the Sensor-based SLAM filter and the *esTIMATE* algorithm, and further subdividing those into the various steps that compose them. For that purpose, recall the description of each one of these components presented in Section 4.1. Note that all the operations not depending on the number of landmarks are omitted in the following lists.

A.1 Sensor-based SLAM filter complexity

The Sensor-based SLAM filter is composed by the traditional steps of the Kalman filter, predict and update, and by the association and loop closing algorithms. These were broken down in the number of each type of operation in the following subsections.

A.1.1 Prediction Step

In the prediction step, the number of landmarks involved in each iteration is the total number of landmarks, N_M, and therefore the overall complexity of the step is determined as a function of this number.

Operation	Number of occurrences	m	n	k	Complexity
Multiplication	2	$6+3N_M$	$6+3N_M$	$6+3N_M$	$\mathcal{O}(N_M^3)$
Multiplication	1	$6+3N_M$	$6+3N_M$	1	$\mathcal{O}(N_M^2)$
Sum	1	$6+3N_M$	$6+3N_M$	–	$\mathcal{O}(N_M^2)$
Overall complexity					$\mathcal{O}(N_M^3)$

Table A.2: Number, type and complexity of operations in the prediction step of the Kalman filter.

A.1.2 Update Step

The update step involves two different landmark sets, the N_O observed landmarks of $\mathcal{I}_{O_k}$, and the full landmark set $\mathcal{I}_{B_k}$ with N_M landmarks. As a complexity analysis has to account for the worst-case scenario, it is considered that all the landmarks in the set are observed, thus yielding $N_O = N_M$. The list of operations of this step of the SLAM filter is presented in Table A.3.

A.1.3 Landmark Association

The landmark association algorithm is, as the reader may recall from Section 4.3, a depth first search algorithm with a search tree with a number of leaves that depends on the size of the observation set and on the number of landmarks to be associated. The landmark sets involved are $\mathcal{I}_{O_k}$ and $\mathcal{I}_{cur_k}$ with N_O and N_{cur} landmarks respectively. In each of the nodes the algorithm calculates the single

Operation	Number of occurrences	m	n	k	Complexity
Inversion	1	$3N_O$	–	–	$\mathcal{O}(N_O^3)$
Multiplication	2	$6+3N_M$	$6+3N_M$	$3N_O$	$\mathcal{O}(N_M^2 N_O)$
Multiplication	1	$6+3N_M$	$3N_O$	$6+3N_M$	$\mathcal{O}(N_M^2 N_O)$
Multiplication	1	$6+3N_M$	$3N_O$	$3N_O$	$\mathcal{O}(N_M N_O^2)$
Multiplication	1	$6+3N_M$	$3N_O$	1	$\mathcal{O}(N_M N_O)$
Sum	1	$6+3N_M$	$6+3N_M$	–	$\mathcal{O}(N_M^2)$
Sum	1	$6+3N_M$	1	–	$\mathcal{O}(N_M)$
Overall complexity					$\mathcal{O}(N_M^3)$

Table A.3: Number, type and complexity of operations in the update step of the Kalman filter.

Mahalanobis distance, which involves 2 multiplications and 1 inversion that do not depend on the number of landmarks. In total, $N_O \times N_{cur}$ single distances are calculated, which means that the complexity of this calculation is $\mathcal{O}(N_O N_{cur})$. Furthermore, in each node of the search space the joint compatibility of the present hypothesis is verified. The complexity of this process is shown in [24] to be $\mathcal{O}(N_{\mathcal{H}_i}^2)$, where $N_{\mathcal{H}_i}$ is the size of the association hypothesis in question. In the worst case, the algorithm has to do this for every single node of the tree, which implies that $N_O \times (N_{cur}+1)$ verifications are performed with the maximum size of an hypothesis – the complexity of this process is $\mathcal{O}(N_O^3 N_{cur})$. If it is considered that the number of recently observed landmarks is the total number of landmarks in the state, and that the number of associations is also the same, then the overall complexity of the association algorithm is $\mathcal{O}(N_M^4)$.

A.1.4 Loop Closing Procedure

The loop closing procedure uses a data association algorithm and the same equations of the Kalman update step. The association is tried with N_{old} and N_{cur} landmarks. It can be easily shown that the worst case takes place when there is no landmarks in the gap set, and when $N_{old} = \frac{3}{4}N_M$ or $N_{cur} = \frac{3}{4}$, depending on how the search space is built. Either way, the complexity of the worst situation is once again $\mathcal{O}(N_M^4)$.

A.2 The *esTIMATE* algorithm

The Inertial Map and Trajectory Estimation algorithm is composed by two main parts: 1. the computation of the optimal translation and rotation from the body-fixed frame to the inertial frame and of the inertial map; and 2. the computation of the uncertainty of the aforementioned estimates. The major influence in the complexity of the algorithm is that of the last part, as it is shown in the following subsections.

A.2.1 Procrustes Problem and Inertial Map Computation

The number of landmarks used in solving the Procrustes problem is N_T, a subset of the total landmarks, as explained in Section 3.2. However, for the worst case scenario, it may considered that this

subset is simply the original set with N_M landmarks. The matrix operations performed in this part of the algorithm are listed in Table A.4.

Operation	Number of occurrences	m	n	k	Complexity
Inversion	1	N_T	–	–	$\mathcal{O}(N_T^3)$
Multiplication	2	N_T	N_T	N_T	$\mathcal{O}(N_T^3)$
Multiplication	3	3	N_T	N_T	$\mathcal{O}(N_T^2)$
Multiplication	2	3	3	N_T	$\mathcal{O}(N_T)$
Multiplication	1	3	N_T	1	$\mathcal{O}(N_T)$
Sum	1	N_T	N_T	–	$\mathcal{O}(N_T^2)$
Sum	2	3	N_T	–	$\mathcal{O}(N_T)$
Overall complexity					$\mathcal{O}(N_M^3)$

Table A.4: Number, type and complexity of operations in the Procrustes Problem and Inertial Map Computation.

A.2.2 Uncertainty Calculation

The final and most computationally expensive step of this algorithm is the computation of the uncertainty of the inertial estimates. The subset $\mathcal{I}_{T_k}$ is used in these calculations, as is the full landmark set. Once again, for the final overall complexity N_T is considered to be the same as N_M.

Operation	Number of occurrences	m	n	k	Complexity
Multiplication	4	3	$3N_T^2$	$3N_T^2$	$\mathcal{O}(N_T^4)$
Multiplication	3	$3N_T$	$3N_T^2$	1	$\mathcal{O}(N_T^3)$
Multiplication	17	$3N_T$	$3N_T$	$3N_T$	$\mathcal{O}(N_T^3)$
Multiplication	36	3	$3N_T$	$3N_T$	$\mathcal{O}(N_T^2)$
Multiplication	1	3	$3N_T$	N_T	$\mathcal{O}(N_T^2)$
Multiplication	3	1	$3N_T$	1	$\mathcal{O}(N_T)$
Multiplication	12	3	3	$3N_T$	$\mathcal{O}(N_T)$
Multiplication	12	3	$3N_T$	3	$\mathcal{O}(N_T)$
Multiplication	1	3	3	N_M	$\mathcal{O}(N_M)$
Multiplication	7	$3N_M$	$3N_M$	$3N_M$	$\mathcal{O}(N_M^3)$
Sum	6	$3N_T$	$3N_T$	–	$\mathcal{O}(N_T^2)$
Sum	4	$3N_M$	$3N_M$	–	$\mathcal{O}(N_M^2)$
Sum	N_T+1	3	N_T	–	$\mathcal{O}(N_T^2)$
Overall complexity					$\mathcal{O}(N_M^4)$

Table A.5: Number, type and complexity of operations in the computation of the *esTIMATE* algorithm uncertainty.

A.3 Final Remarks

This appendix presented a study of the time complexity of the various parts that compose the novel algorithm proposed in this work. Although only operational complexity was addressed, the thorough description of the operations employed included the size of the matrices involved, thus presenting a measure of the space complexity of the algorithm. The conjugation of the local complexities for each part of the algorithm leads to an overall complexity in the order of $\mathcal{O}(N_M^4)$. Furthermore, the analysis

done here indicates that the two most computationally demanding steps are landmark association and uncertainty calculation. One point that is important to notice is the fact that the complexity assumed for the basic operations is conservative, as there exist optimized algorithms to perform these operations (see [43]).

B

Matrix operations

This appendix addresses the problem of finding matrix operations correspondent to the summation loops of Algorithm 2. Recall the expressions used to compute the uncertainty of the optimal rotation and translation, and the inertial map. For the sake of completeness, they are reproduced again here. Firstly, consider the expression for the covariance of the translation estimate,

$$\boldsymbol{\Sigma}_{I_{p_k}} \approx \frac{1}{N_W^2} \sum_{i,j=1}^{N_T} \sigma_i^{-2} \sigma_j^{-2} \left[\boldsymbol{\Sigma}_{I_{p_{ij_k}}} + \hat{\mathbf{R}}_k^{(0)} \boldsymbol{\Sigma}_{p_{ij_k}} \hat{\mathbf{R}}_k^{(0)^T} + \mathbf{S} \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)} \right] \boldsymbol{\Sigma}_{\Omega_k} \mathbf{S}^T \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)} \right] \right]. \tag{B.1}$$

The first parcel can be computed using the following generic equality,

$$\sum_{i,j=1}^{N} \rho_i \upsilon_j \mathbf{A}_{ij} = \begin{bmatrix} \rho_1 \mathbf{I}_3 & \cdots & \rho_N \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{N1} & \cdots & \mathbf{A}_{NN} \end{bmatrix} \begin{bmatrix} \upsilon_1 \mathbf{I}_3 \\ \vdots \\ \upsilon_N \mathbf{I}_3 \end{bmatrix}, \tag{B.2}$$

where $\mathbf{A}_i j \in \mathbb{R}^{3\times 3}$. The matrices involving ρ_i and υ_i, $i \in 1, \ldots, N$ are equivalent to

$$\boldsymbol{P} := \begin{bmatrix} \mathbf{I}_3 & \cdots & \mathbf{I}_3 \end{bmatrix} \operatorname{diag}(\rho_1 \mathbf{I}_3, \ldots, \rho_N \mathbf{I}_3) = \operatorname{kron}(\mathbf{1}_N^T, \mathbf{I}_3) \operatorname{kron}\left(\operatorname{diag}(\rho_1, \ldots, \rho_N), \mathbf{I}_3\right), \tag{B.3}$$

and

$$\boldsymbol{\Upsilon}^T := \begin{bmatrix}\mathbf{I}_3 & \cdots & \mathbf{I}_3\end{bmatrix} \operatorname{diag}(\upsilon_1\mathbf{I}_3, \ldots, \upsilon_N\mathbf{I}_3) = \operatorname{kron}(\mathbf{1}_N^T, \mathbf{I}_3) \operatorname{kron}\left(\operatorname{diag}(\upsilon_1, \ldots, \upsilon_N), \mathbf{I}_3\right), \tag{B.4}$$

where $\operatorname{kron}(.)$ is the Kronecker product that respects $\operatorname{kron}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \mathbf{A}\right) = \begin{bmatrix} a\mathbf{A} & b\mathbf{A} \\ c\mathbf{A} & d\mathbf{A} \end{bmatrix}$. The second parcel $\sum_{i,j=1}^{N_T} \sigma_i^{-2}\sigma_j^{-2}\hat{\mathbf{R}}_k^{(0)}\boldsymbol{\Sigma}_{p_{ij_k}}\hat{\mathbf{R}}_k^{(0)^T}$ is rearranged to obtain $\hat{\mathbf{R}}_k^{(0)}\left(\sum_{i,j=1}^{N_T} \sigma_i^{-2}\sigma_j^{-2}\boldsymbol{\Sigma}_{p_{ij_k}}\right)\hat{\mathbf{R}}_k^{(0)^T}$, which may now be computed using (B.2). The final term is more difficult to obtain in matrix form, raising the need for auxiliary calculations. Consider the following operation that takes as input a 3 by N matrix and creates a block diagonal matrix of skew-symmetric matrices $\mathbf{S}[.]$ for each of the columns of the input matrix.

```
function B ∈ ℝ^{3N×3N} = tensorize(A ∈ ℝ^{3×N})
  D = diag(column(A));
  X1 = [0 0 1; 0 0 1; 0 1 0]; X2 = [0 1 0; 1 0 0; 1 0 0];
  Y1 = [0 1 −1; −1 0 0; 0 0 0]; Y2 =[0 0 0; 0 0 1; 1 −1 0];
  X1c = kron(eye(3), X1); X2c = kron(eye(3), X2);
  Y1c = kron(eye(3), Y1); Y2c = kron(eye(3), Y2);

  B = Y1c D X1c + Y2c D X2c;
end
```

The intermediate step $\operatorname{columnize}$ takes the same input matrix as the previous function, $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_N\end{bmatrix}$, and generates the column vector $\begin{bmatrix}\mathbf{a}_1^T & \cdots & \mathbf{a}_N^T\end{bmatrix}^T$.

```
function b ∈ ℝ^{3N} = columnize(A ∈ ℝ^{3×N})
  X = kron(eye(N), A);
  Y = [1 zeros(1,N)]';
  Z = kron(ones(N,1), Y);
  Z = Z(1:end−N);

  b = XZ;
end
```

Using these operations, it is possible to obtain a matrix that concatenates $\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\boldsymbol{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right]$ for all $i, j \in 1, \ldots, N_T$,

$$\begin{aligned}&\begin{bmatrix} \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right]\boldsymbol{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right] & \cdots & \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right]\boldsymbol{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right] \\ \vdots & \ddots & \vdots \\ \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right]\boldsymbol{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right] & \cdots & \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right]\boldsymbol{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right] \end{bmatrix} \\ &= \boldsymbol{\Psi}\operatorname{kron}\left(\mathbf{1}_{N_T}\mathbf{1}_{N_T}^T, \boldsymbol{\Sigma}_{\Omega_k}\right)\boldsymbol{\Psi}^T,\end{aligned} \tag{B.5}$$

with

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right] & \cdots & \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{1_k}^{(0)}\right] \\ \vdots & \ddots & \vdots \\ \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right] & \cdots & \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{N_{T_k}}^{(0)}\right] \end{bmatrix} = \text{tensorize}\left(\hat{\mathbf{R}}_k^{(0)}\mathbf{X}_k\right), \tag{B.6}$$

where $\mathbf{X}_k$, the concatenation of the sensor-based landmarks, is defined in Section 3.1. Once more using (B.2) with the proper substitutions yields the intended sum $\sum_{i,j=1}^{N_T}\sigma_i^{-2}\sigma_j^{-2}\mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\right]\mathbf{\Sigma}_{\Omega_k}\mathbf{S}^T\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)}\right]$, which finishes the treatment of expression (B.1), allowing to proceed to the rotation uncertainty expressed by (3.57) and (3.60). The last expression is reproduced here for clarity sake:

$$\langle \mathbf{C}_k\mathbf{C}_k\rangle = \sum_{i=1}^{N_T}\sum_{j=1}^{N_T}\sigma_i^{-2}\sigma_j^{-2}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)^T}\hat{\mathbf{R}}_k^{(0)^T}\langle \mathbf{a}_{i_k}^{(1)}\mathbf{a}_{j_k}^{(1)^T}\rangle + \hat{\mathbf{R}}_k^{(0)}\mathbf{\Sigma}_{p_{i_{j_k}}}\hat{\mathbf{R}}_k^{(0)^T}\mathbf{a}_{i_k}^{(0)}\mathbf{a}_{j_k}^{(0)^T}\right] \tag{B.7a}$$

$$\langle \mathbf{C}_k\mathbf{C}_k^T\rangle = \sum_{i=1}^{N_T}\sum_{j=1}^{N_T}\sigma_i^{-2}\sigma_j^{-2}\left[\hat{\mathbf{R}}_k^{(0)}\hat{\mathbf{p}}_{i_k}^{(0)}\hat{\mathbf{p}}_{j_k}^{(0)^T}\hat{\mathbf{R}}_k^{(0)^T}\,\text{tr}(\mathbf{a}_{i_k}^{(1)}\mathbf{a}_{j_k}^{(1)^T}) + \hat{\mathbf{R}}_k^{(0)}\mathbf{\Sigma}_{p_{i_{j_k}}}\hat{\mathbf{R}}_k^{(0)^T}\mathbf{a}_{i_k}^{(0)^T}\mathbf{a}_{j_k}^{(0)}\right] \tag{B.7b}$$

$$\langle \mathbf{C}_k^T\mathbf{C}_k\rangle = \sum_{i=1}^{N_T}\sum_{j=1}^{N_T}\sigma_i^{-2}\sigma_j^{-2}\left[\hat{\mathbf{p}}_{i_k}^{(0)^T}\hat{\mathbf{p}}_{j_k}^{(0)}\langle \mathbf{a}_{i_k}^{(1)}\mathbf{a}_{j_k}^{(1)^T}\rangle + \text{tr}(\mathbf{\Sigma}_{p_{i_{j_k}}})\mathbf{a}_{i_k}^{(0)}\mathbf{a}_{j_k}^{(0)^T}\right], \tag{B.7c}$$

with

$$\langle \mathbf{a}_{i_k}^{(1)}\mathbf{a}_{j_k}^{(1)^T}\rangle = \mathbf{\Sigma}_{I_{p_{i_{j_k}}}} - \frac{1}{N_W}\sum_{r=1}^{N_T}\sigma_r^{-2}\left(\mathbf{\Sigma}_{I_{p_{i_{r_k}}}} + \mathbf{\Sigma}_{I_{p_{r_{j_k}}}}\right) + \frac{1}{N_W^2}\sum_{r=1}^{N_T}\sum_{s=1}^{N_T}\sigma_r^{-2}\sigma_s^{-2}\mathbf{\Sigma}_{I_{p_{r_{s_k}}}}, \tag{B.8}$$

and $\mathbf{a}_{i_k}^{(.)} := {}^I\hat{\mathbf{p}}_{i_k}^{(.)} - \frac{1}{N_W}\sum_{j=1}^{N_T}{}^I\hat{\mathbf{p}}_{j_k}^{(.)}$. The following step is to compute the concatenation of the decentralized landmarks $\mathbf{Z}_k := \begin{bmatrix}\mathbf{a}_{1_k} & \cdots & \mathbf{a}_{N_{T_k}}\end{bmatrix}$,

$$\mathbf{Z}_k = \mathbf{Y}_k - \frac{1}{N_W}\mathbf{Y}_k\mathbf{\Sigma}_{I_{e_k}}^{-1}\mathbf{1}_{N_T}\mathbf{1}_{N_T}^T. \tag{B.9}$$

Any of the landmark concatenations may help constitute a norm matrix similar to the following, exemplified for the decentralized set,

$$\begin{bmatrix} \mathbf{a}_{1_k}\mathbf{a}_{1_k}^T & \cdots & \mathbf{a}_{1_k}\mathbf{a}_{N_{T_k}}^T \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{N_{T_k}}\mathbf{a}_{1_k}^T & \cdots & \mathbf{a}_{N_{T_k}}\mathbf{a}_{N_{T_k}}^T \end{bmatrix} = \text{columnize}(\mathbf{Z}_k)\,\text{columnize}(\mathbf{Z}_k)^T. \tag{B.10}$$

In the vein of expression (B.2), it is straightforward to see that a similar expression can be found for single summations. This yields

$$\begin{bmatrix} \sum_i^N \upsilon_j\mathbf{A}_{1i} & \cdots & \sum_i^N \upsilon_j\mathbf{A}_{1i} \\ \vdots & \ddots & \vdots \\ \sum_i^N \upsilon_j\mathbf{A}_{Ni} & \cdots & \sum_i^N \upsilon_j\mathbf{A}_{Ni} \end{bmatrix} = \mathbf{A}\,\text{kron}\left(\mathbf{1}_N, \mathbf{\Upsilon}\right). \tag{B.11}$$

The computation of the full covariance matrix for decentralized landmarks is now easy. Consider the previous expression to compute

$$\{\mathbf{\Sigma}_{I x_{M_k}}\}_{sum} = \begin{bmatrix} \sum_{i=1}^{N_T} \sigma_i^{-2} \mathbf{\Sigma}_{I p_{1 i_k}} & \cdots & \sum_{i=1}^{N_T} \sigma_i^{-2} \mathbf{\Sigma}_{I p_{1 i_k}} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{N_T} \sigma_i^{-2} \mathbf{\Sigma}_{I p_{N_T i_k}} & \cdots & \sum_{i=1}^{N_T} \sigma_i^{-2} \mathbf{\Sigma}_{I p_{N_T i_k}} \end{bmatrix}.$$

Knowing this, the full covariance matrix $\mathbf{\Sigma}_{a_k}$ becomes

$$\mathbf{\Sigma}_{a_k} = \mathbf{\Sigma}_{I x_{M_k}} - \frac{1}{N_W}\{\mathbf{\Sigma}_{I x_{M_k}}\}_{sum} - \frac{1}{N_W}\{\mathbf{\Sigma}_{I x_{M_k}}\}_{sum}^T + \frac{1}{N_W^2} \operatorname{kron}\left(\mathbf{1}_{N_T}\mathbf{1}_{N_T}^T, \sum_{i,j=1}^{N_T} \sigma_i^{-2}\sigma_j^{-2} \mathbf{\Sigma}_{I p_{i j_k}}\right), \tag{B.12}$$

where the last term in the argument of the Kronecker product is computed using (B.2).

It is now possible to compute (B.7) as matrix operations. Consider the following generic relations

$$\sum_{i,j=1}^{N} \rho_i \upsilon_j \mathbf{a}_i \mathbf{a}_j^T \mathbf{b}_i^T \mathbf{b}_j = \boldsymbol{P}\mathbf{A} \otimes \operatorname{kron}\left(\mathrm{TR}(\mathbf{B}), \mathbf{1}_3\mathbf{1}_3^T\right) \mathbf{\Upsilon} \tag{B.13}$$

and

$$\sum_{i,j=1}^{N} \rho_i \upsilon_j \mathbf{a}_i \mathbf{a}_j^T \mathbf{b}_i \mathbf{b}_j^T = \boldsymbol{P}\mathbf{A} \otimes \mathbf{B}\mathbf{\Upsilon}, \tag{B.14}$$

where $\otimes$ denotes the element-wise product, and the operation $\mathrm{TR}(.)$ computes the trace of the 3 by 3 submatrices in its input,

$$\mathrm{TR}(\mathbf{A}) = \begin{bmatrix} \mathrm{tr}(\mathbf{A}_1 1) & \cdots & \mathrm{tr}(\mathbf{A}_1 N) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{A}_N 1) & \cdots & \mathrm{tr}(\mathbf{A}_N N) \end{bmatrix} \tag{B.15}$$

```
function T ∈ ℝ^{N×N} = TR(P ∈ ℝ^{3N×3N})
  n=N/3;
  A_1 = kron(eye(N), [1 0 0]);
  A_2 = kron(eye(N), [0 1 0]);
  A_3 = kron(eye(N), [0 0 1]);

  T = A_1 P A_1^T + A_2 P A_2^T + A_3 P A_3^T
end
```

A close observation of (B.7) and comparison with (B.13) shows that the expressions are completely equivalent, with the correct substitutions of course. Given that $\mathbf{\Sigma}_{\Omega_k}$ is a combination of these three matrices, it is fully defined now as a matricial operation, which leaves only the inertial map covariance

to compute. Recall the expression that yields the covariance of two inertial landmarks $i, j \in \mathcal{I}_{I_k}$,

$$\begin{aligned}\boldsymbol{\Sigma}_{{}^I p_{i_{j_{k+1}}}} \approx \boldsymbol{\Sigma}_{{}^I p_k} + \hat{\mathbf{R}}_k^{(0)} \boldsymbol{\Sigma}_{p_{i_{j_k}}} \hat{\mathbf{R}}_k^{(0)^T} + \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)}\right] \boldsymbol{\Sigma}_{\Omega_k} \mathbf{S}^T \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)}\right] \\ + \mathbf{S}^T \left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{i_k}^{(0)}\right] \boldsymbol{\Sigma}_{{}^I p_k \Omega_k}^T + \boldsymbol{\Sigma}_{{}^I p_k \Omega_k} \mathbf{S}\left[\hat{\mathbf{R}}_k^{(0)} \hat{\mathbf{p}}_{j_k}^{(0)}\right],\end{aligned} \tag{B.16}$$

The full covariance matrix of the concatenated state ${}^I\mathbf{x}_{M_k}$ can be computed with matrix operations by

$$\begin{aligned}\boldsymbol{\Sigma}_{{}^I x_{M_{k+1}}} = &\operatorname{kron}\left(\mathbf{1}_{N_M} \mathbf{1}_{N_M}^T, \boldsymbol{\Sigma}_{{}^I p_k}\right) + \operatorname{kron}\left(\mathbf{I}_{N_M} \hat{\mathbf{R}}_k^{(0)}\right) \boldsymbol{\Sigma}_{x_{M_k}} \operatorname{kron}\left(\mathbf{I}_{N_M} \hat{\mathbf{R}}_k^{(0)^T}\right) + \\ &\boldsymbol{\Psi} \operatorname{kron}\left(\boldsymbol{\Sigma}_{\Omega_k}\right) \boldsymbol{\Psi}^T + \boldsymbol{\Psi}^T \operatorname{kron}\left(\mathbf{1}_{N_M} \mathbf{1}_{N_M}^T, \boldsymbol{\Sigma}_{{}^I p_k \Omega_k}^T\right) + \operatorname{kron}\left(\mathbf{1}_{N_M} \mathbf{1}_{N_M}^T, \boldsymbol{\Sigma}_{{}^I p_k \Omega_k}\right) \boldsymbol{\Psi},\end{aligned} \tag{B.17}$$

where each column of the cross covariance between the translation and rotation estimates is a combination of the $\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle$ $l, m = 1, \ldots, 3$, which in turn are computed in two parts: the body-related and inertial-related,

$$\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle \approx \frac{1}{N_W} \sum_{i,j=1}^{N_M} \sigma_i^{-2} \sigma_j^{-2} \left[\langle {}^I\hat{\mathbf{p}}_{i_k}^{(1)} \mathbf{a}_{j_k}^{(1)}(m) \rangle \hat{\mathbf{R}}_k^{(0)}(l,:) \hat{\mathbf{p}}_{j_k}^{(0)} - \hat{\mathbf{R}}_k^{(0)} \langle \hat{\mathbf{p}}_{i_k}^{(1)} \hat{\mathbf{R}}_k^{(0)}(l,:) \hat{\mathbf{p}}_{j_k}^{(1)} \rangle \mathbf{a}_{j_k}^{(0)}(m)\right], \tag{B.18}$$

The body-related uncertainty is given by

$$\left\{\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle\right\}_{body} = \hat{\mathbf{R}}_k^{(0)} \operatorname{kron}\left(\mathbf{1}_{N_M}, \mathbf{I}_3\right) \operatorname{kron}\left(\boldsymbol{\Sigma}_{{}^I e_k}^{-1}\right) \boldsymbol{\Sigma}_{x_{M_k}} \boldsymbol{\Lambda}_k \hat{\mathbf{R}}_k^{(0)}(l,:)^T, \tag{B.19}$$

with $\boldsymbol{\Lambda}_k$ given by $\boldsymbol{\Lambda}_k = \operatorname{kron}\left(\operatorname{diag}\left(\operatorname{diag}\left(\operatorname{diag}\left(\boldsymbol{\Sigma}_{{}^I e_k}^{-1}\right) \mathbf{Z}_k(m,:)\right)\right)\right)$. The computation of the inertial-related uncertainty requires several steps:

$\mathbf{sel}_c$ = **zeros**(1,3);
$\mathbf{sel}_c(m)$ = 1;
$\left\{\boldsymbol{\Sigma}_{{}^I x_M a_k}\right\}_{sel}$ = $\boldsymbol{\Sigma}_{{}^I x_M a_k}$(:, logical(**kron**(ones(1,N_T)), $\mathbf{sel}_c$));
$\left\{\langle {}^I\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle\right\}_{inertial}$ = rowenize($\left\{\boldsymbol{\Sigma}_{{}^I x_M a_k}\right\}_{sel} \left(\hat{\mathbf{R}}_k^{(0)}(l,:) \mathbf{X}_k \boldsymbol{\Sigma}_{{}^I e_k}^{-1}\right)^T$) **diag**($\boldsymbol{\Sigma}_{{}^I e_k}^{-1}$)

where the cross covariance between the inertial landmark state vector and the decentralized landmarks is given by

$$\boldsymbol{\Sigma}_{{}^I x_M a_k} = \boldsymbol{\Sigma}_{{}^I x_{M_k}} - \frac{1}{N_W} \boldsymbol{\Sigma}_{{}^I x_{M_k}} \operatorname{kron}\left(\mathbf{1}_{N_T}^T, \operatorname{kron}\left(\boldsymbol{\Sigma}_{{}^I e_k}^{-1}, \mathbf{I}_3\right) \operatorname{kron}\left(\mathbf{1}_{N_T}, \mathbf{I}_3\right)\right), \tag{B.20}$$

and the operation $\operatorname{rowenize}(.)$ transforms a column vector of dimension $3N$ into a matrix of dimensions $3 \times N$

function $\mathbf{A} \in \mathbb{R}^{3\times N}$ = rowenize($\mathbf{a} \in \mathbb{R}^{3N}$)
  $\mathbf{X}$ = **kron**(ones(1,N), **eye**(3));
  $\mathbf{D}$ = **diag**($\mathbf{a}$);
  $\mathbf{Y}$ = **kron**(**eye**(N), ones(3,1));

  $\mathbf{A}$ = $\mathbf{XDY}$;
end

With both parts of the cross covariance $\langle {}^{I}\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle$ computed, the final expression is now

$$\langle {}^{I}\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle = \frac{1}{N_W} \left(\left\{ \langle {}^{I}\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle \right\}_{inertial} - \left\{ \langle {}^{I}\hat{\mathbf{p}}_k^{(1)} c_{lm} \rangle \right\}_{body} \right). \tag{B.21}$$

This concludes this appendix, where all the expressions derived in Section 3.3 were rewritten as matrix operations for time optimization of their computation.