

# *Strategies for uncertainty optimization through motion planning in GES sensor-based SLAM*

Pedro Lourenço, Pedro Batista, Paulo Oliveira, and Carlos Silvestre

*Robotics and Autonomous Systems*, vol. 113, pp. 38-55, March 2019

<https://doi.org/10.1016/j.robot.2018.12.005>

## Accepted Version

Level of access, as per info available on SHERPA/ROMEO

<http://www.sherpa.ac.uk/romeo/search.php>

Robotics and Autonomous Systems

Publication Information	
Title	Robotics and Autonomous Systems (English)
ISSNs	Print: 0921-8890
URL	<a href="http://www.elsevier.com/wps/product/cws_home/505622/description">http://www.elsevier.com/wps/product/cws_home/505622/description</a>
Publishers	Elsevier [Commercial Publisher] North-Holland [Associate Organisation] Intelligent Autonomous Systems (IAS) Society [Associate Organisation]

Publisher Policy	
Open Access pathways permitted by this journal's policy are listed below by article version. Click on a pathway for a more detailed view.	
Published Version [pathway a]	None  CC BY-NC-ND PMC, Non-Commercial Repository, Research for Development Repository, +2
Published Version [pathway b]	None  CC BY Institutional Repository, Subject Repository, PMC, Research for Development Repository, +2
Published Version [pathway c]	None  CC BY  PMC Institutional Repository, Subject Repository, PMC, Research for Development Repository, +2
Accepted Version [pathway a]	None  CC BY-NC-ND arXiv, RePEc, Author's Homepage
Embargo	No Embargo
Licence	CC BY-NC-ND
Location	Author's Homepage Named Repository (arXiv, RePEc)
Conditions	Must link to publisher version with DOI
Notes	Authors can share their accepted manuscript immediately by updating a preprint in arXiv or RePEc with the accepted manuscript.
Accepted Version [pathway b]	24m  CC BY-NC-ND Institutional Repository, Subject Repository
Accepted Version [pathway c]	12m  CC BY-NC-ND Institutional Repository, Subject Repository
Submitted Version	None Any Website, +2

For more information, please see the following links:

- Sharing Policy
- Green open access
- Unleashing the power of academic sharing
- Journal Embargo List for UK Authors
- Open access
- Funding Body Agreements
- Attaching a User License
- Sharing and Hosting Policy FAQ
- Open access licenses
- Article Sharing
- Journal Embargo Period List

# Strategies for Uncertainty Optimization through Motion Planning in GES sensor-based SLAM

Pedro Lourenço<sup>a,\*</sup>, Pedro Batista<sup>a,b</sup>, Paulo Oliveira<sup>a,c</sup>, Carlos Silvestre<sup>a,d</sup>

<sup>a</sup>*Institute for Systems and Robotics, Laboratory for Robotics and Engineering Systems, Portugal.*

<sup>b</sup>*Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*

<sup>c</sup>*Department of Mechanical Engineering, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*

<sup>d</sup>*Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau*

---

## Abstract

This paper addresses the problem of minimizing the uncertainty through motion planning in a globally exponentially stable sensor-based simultaneous localization and mapping algorithm, with the objective of performing active exploitation. This is done by designing an optimization problem that weighs the final uncertainty, the overall uncertainty in the horizon considered, and the cost of the control. Using the Pontryagin minimum principle and building on the derivation of the Kalman filter by Athans and Tse as well as on Hussein's extension for motion planning, the optimization problem is transformed into a two-point boundary value problem that encodes necessary conditions for the input that minimizes the uncertainty. A strategy is proposed to solve this problem numerically, and particular examples are analysed. Following the shortcomings identified in this procedure, the original optimization problem is modified assuming that the input velocities are piecewise constant functions. A direct approach is used to solve this new optimization problem, allowing the in-depth analysis of more realistic scenarios.

*Keywords:* Simultaneous localization and mapping, optimal control, uncertainty optimization.

---

---

\*Corresponding author

*Email addresses:* plourenco@isr.tecnico.ulisboa.pt (Pedro Lourenço), pbatista@isr.tecnico.ulisboa.pt (Pedro Batista), paulo.j.oliveira@tecnico.ulisboa.pt (Paulo Oliveira), csilvestre@umac.mo (Carlos Silvestre)

## 1. Introduction

Simultaneous localization and mapping (SLAM) is the problem of navigating a vehicle in an unknown environment, by simultaneously building a map of the area and using this map to deduce its location, without the need for a priori knowledge of the location. See [1] and [2] for a thorough survey on the algorithms proposed in the first decades of SLAM research, [3] for a specialized review of visual SLAM, [4] for a more up-to-date survey focused in the recent theoretical achievements and [5] for an overarching survey on the history and remaining present and future challenges of SLAM, e.g., robustness and scalability, and so forth.

The SLAM problem can be formulated in several different ways, and many methodologies follow the idea of performing the estimation in an Earth-fixed frame. A different trend was exploited by the authors yielding sensor-based filters for solving the simultaneous localization and mapping problem in its main formulations – range-and-bearing (2-D [6] and 3-D [7]), range-only [8], and bearing-only [9] – while achieving convergence guarantees previously absent from the literature. These filters, more consistent than the world-centric counterpart [10], avoid the inclusion of the pose of the vehicle in the filter state, one of the main sources of non-linearity. This results in globally exponentially stable (GES) error dynamics.

The main paradigm in SLAM is to move to gain new knowledge and improve what is known. In the formative years of SLAM, the question of how to move was generally separated from the estimation problem. However, the field of active perception or active sensing has been developing concurrently (see [11], [12] and [13]). In fact, there are works that bridge those two communities, yielding, for example, SLAM algorithms with active vision [14]. In the light of this, several works addressed the issue of optimizing moving in the context of SLAM, thus introducing Active SLAM (see, for example, [15], [16], and [17]).

### 1.1. Active SLAM: a complex problem

The objective of Active SLAM is to plan ahead the motion of the vehicle in order to maximize the explored areas and minimize the uncertainty associated with the estimation. These two objectives are, in a sense, complementary: exploration involves moving in previously unvisited terrain with the objective of increasing the overall knowledge of the environment, while the latter is exploitation, i.e., it involves revisiting areas to maximize the information gain. For example, [18] addresses the concept of exploration as long term optimization, versus the short-term optimization that, in their view, underlies exploitation. Possible strategies for this involve following walls [19] or (pre-set) goals [20], for instance. This last paper and references therein are examples of algorithms that mix exploration and exploitation. In this paper, however, the focus is on the exploitation part of Active SLAM, which requires that some form of exploration is already done, i.e., that a map has been populated. One of the main ideas behind exploitation is related to loop closing, a very important step of a simultaneous localization and mapping algorithm. Therefore, exploitation in

45 the context of SLAM may be seen as a strategy to decide when and where to close loops, as is argued in [21].

A relevant discussion in the optimization of the exploitation step is how to measure uncertainty, or alternatively, how to quantify the information gain [22]. The main possibilities, stemming from the EKF approach to SLAM [23],  
50 are 1. *A-opt* - the trace of the covariance, i.e., the average variance across all states; 2. *D-opt* - the determinant of the covariance, which is proportional to the volume of the covariance matrix; 3. *E-opt* - the minimum eigenvalue of the covariance; and 4. the entropy of the distribution. In [22] a comparison of these uncertainty criteria with planning under uncertainty is provided. The  
55 planning is performed as a search in a set of finite-horizon EKF-SLAM runs and finding the one that minimizes the different functionals. The authors conclude that *D-opt* is, in their case, the one with best results, even though most of the literature in Active SLAM uses the *A-opt* criterion. Following that line of work,  
the authors of [24] and [25] studied the monotonicity of all the criteria in the  
60 exploration phase (where the pose is computed by linearized dead-reckoning) and strengthened the case in favour of *D-opt*. It should be noted that, while it may yield better results, using the determinant raises unnecessary difficulties in the mathematical work of minimizing the functional when compared with the common criterion *A-opt*. Furthermore, this paper is focused on exploitation  
65 using a linear (and not linearized) system, where some of the raised issues may not be as relevant. The most relevant issue in Active SLAM is, however, how to find the path that minimizes the uncertainty. One possibility is to address this as the problem of planning the sequence of discrete positions through the environment that forms the maximally informative trajectory, by discretizing  
70 the environment into a grid, as is done in [26] for EKF-SLAM and in [27] for particle filter SLAM. Other approaches include model predictive control [28, 29] or the recent trend of multiple robot missions, such as [30] which uses relative entropy optimization and an EKF. Even though this is still a growing field of research, there are already applications as diverse as office navigation [31] and  
75 underwater inspection [32]. Not specifically applied to Active SLAM, but still with the objective of uncertainty reduction, an interesting idea is presented in [33], where the authors approach the problem of path planning for uncertain exploration grounded in the information theory of Shannon.

## 80 1.2. Proposed approach

This paper is rooted on the idea proposed in [34], which deals with motion planning with the objective of reducing the uncertainty in a filter. That work recovers a traditional optimal control and estimation result [35], and, using the Pontryagin minimum principle (PMP) [36], proposes a control strategy for  
85 mobile sensors that influences the evolution of the uncertainty in a Kalman filter. Given the linear character of the Kalman filter, this is only possible if any of the parameters of the filter depends, even if indirectly, on the input of another system. In this paper, the cost functional proposed in [34] is used to define an

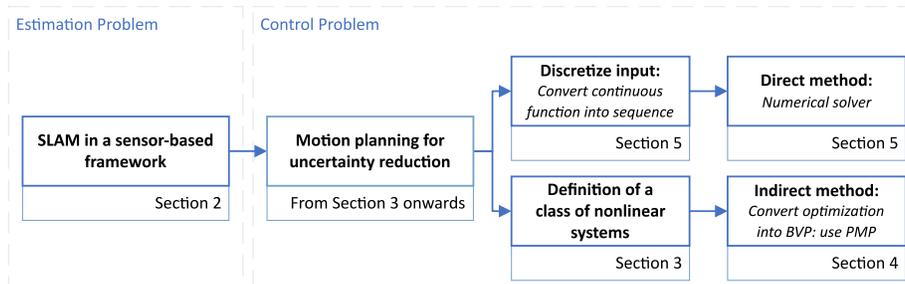


Figure 1: Workflow diagram detailing the evolution of the work reported in this paper.

optimization problem for a more general class of systems into which the sensor-based SLAM framework fits. Having this cost functional, based on the  $A$ -opt criterion for the reasons stated above, there are two possibilities for optimization – indirect and direct methods – both explored in this paper, as explained in Figure 1. As an indirect method, the Pontryagin minimum principle is used in after Hussein’s idea. This results in the proposal of necessary conditions for the trajectory that optimizes the uncertainty in a SLAM Kalman filter. These conditions appear in the form of a two-point boundary value problem (BVP) that is solved numerically, and some examples are analysed in detail. This research was first presented in conference form in [37]. However, even though it is an elegant solution to the active SLAM problem, solving the two-point boundary value problem is very difficult. In fact, to be able to numerically solve the BVP for a reasonable horizon, it seems to be necessary to constrain the motion of the vehicle to slow movements and to a fully actuated model, at least with the currently available computational tools. To tackle the shortcomings of this solution, the optimization problem that led to the use of the Pontryagin minimum principle is modified to constrain the inputs to piecewise constant functions, thus reducing the problem to a search of a finite sequence depending on the problem horizon and the duration of each step. This new optimization problem can then be solved numerically with standard optimization tools for a much larger set of parameters and realistic motion models, which constitutes the direct method. With this tool at hand, it is possible to perform an in-depth study of the problem and of the obtained optimal trajectories. All this is new research not included in [37].

In summary, the contributions of this paper are: (i) the extension of the idea in [34] to a more general class of systems, that not only applies to sensor-based SLAM [38] but also to a variety of other problems, from source localization [39] to attitude determination [40]; (ii) the first application of the PMP to SLAM, in particular to sensor-based SLAM, providing a framework for active exploitation; (iii) a direct numerical solution to active exploitation based on a globally exponentially stable SLAM filter; and (iv) relevant insights into the behaviour of the uncertainty in mapping and how it relates to the position of the vehicle and its motion.

### 1.3. Paper organization and notation

The paper is organized as follows. Section 2 overviews the sensor-based SLAM approach and presents the problem statement. In Section 3 the results on which this paper builds are reviewed, and a control strategy for uncertainty reduction through optimal motion planning using an indirect method is presented. Section 4 applies the previous results to active SLAM, introduces the algorithm proposed to solve the resulting boundary value problem, and presents the incipient numerical results of the solution with a short discussion. This concludes the first part of the paper. In the second part, the optimization problem advanced before is modified and solved using a direct method in Section 5, followed by simulation results exploring various scenarios. Finally, Section 6 presents some conclusions and future work directions.

*Notation.* The following symbol convention is used in this paper: vectors are represented in bold lower case letters, matrices in bold capital letters and scalar symbols are expressed in italic, constants by capital letters, and variables in lower case letters. The superscript  $E$  indicates a vector or matrix expressed in the Earth-fixed frame  $\{E\}$ . For the sake of clarity, when no superscript is present, the vector is expressed in the body-fixed frame  $\{B\}$ . Particularly, the symbol  $\mathbf{0}_{n \times m}$  denotes an  $n \times m$  matrix of zeros (if only one subscript is present, the matrix is square),  $\mathbf{I}_n$  is an identity matrix with dimension  $n \times n$ ,  $\mathbf{1}_n$  is a  $n \times 1$  column vector of ones, and  $\bar{\mathbf{1}}_i$  is a vector whose  $i$ -th element is equal to 1 and is the only non-zero one. The dimensions of the latter can be ascertained from context.  $\mathbf{S}(\mathbf{a}) \in \mathbb{R}^{3 \times 3}$  is a skew-symmetric matrix, henceforth called the cross-product matrix, as  $\mathbf{S}(\mathbf{a}) \mathbf{b} = \mathbf{a} \times \mathbf{b}$  with  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ . When dealing with two-dimensional quantities,  $\mathbf{S}(a) \in \mathbb{R}^{2 \times 2}$  is also a skew-symmetric matrix with only one parameter  $a \in \mathbb{R}$ . For generalization of the results of this paper it is convenient to define the matrix

$$\bar{\mathbf{S}}^T(\mathbf{a}) = \begin{cases} \mathbf{S}(1)\mathbf{a}, & n = 2 \\ -\mathbf{S}(\mathbf{a}), & n = 3 \end{cases} \quad (1)$$

for  $\mathbf{a} \in \mathbb{R}^n$ . The inverse function  $\mathbf{S}^{-1} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{\frac{n(n-1)}{2} \times 1}$  extracts the vector that characterizes the skew-symmetric matrix, i.e.,  $\mathbf{S}^{-1}(\mathbf{S}(\mathbf{a})) = \mathbf{a}$ . Finally, quantities denoted with  $(\cdot)|_*$  are evaluated along the optimal trajectory.

## 2. Sensor-based SLAM Overview

Building on the idea of robocentric filtering, [38] addresses the problem of designing a navigation system in a sensor-based framework for a vehicle capable of sensing landmarks in a previously unknown environment. This is done resorting to a purely sensor-based SLAM algorithm where no linearization or approximation is used whatsoever and pose representation in the state is suppressed, therefore avoiding its pitfalls. This section presents a brief overview of

145 the work proposed in those papers, that serve as groundwork for the optimal control problem to address in this work.

### 2.1. Nonlinear system dynamics

Let the pair  $(\mathbf{R}(t), {}^E\mathbf{p}(t)) \in \text{SO}(n) \times \mathbb{R}^n$ ,  $n \in \{2, 3\}$ , encode the transformation from the body-fixed frame  $\{B\}$  to an inertial or Earth-fixed frame  $\{E\}$ .  
 150  $\mathbf{R}(t)$  is a rotation matrix satisfying  $\dot{\mathbf{R}}(t) = \mathbf{R}(t)\mathbf{S}(\boldsymbol{\omega}(t))$ , where  $\boldsymbol{\omega}(t) \in \mathbb{R}^{\frac{n(n-1)}{2}}$  is the angular velocity, expressed in body-fixed coordinates, and  ${}^E\mathbf{p}(t)$  represents the vehicle position, assumed coincident with the origin of the body-fixed frame, in the Earth-fixed frame. Consider also the existence of static landmarks in the environment whose coordinates can be perceived by the vehicle. These  
 155 define the map and can be separated in two complementary sets,  $\mathcal{M}_O$  and  $\mathcal{M}_U$ . The former contains the  $N_O$  visible landmarks, while the latter contains the  $N_U$  non-visible.

Consider that the vehicle is equipped with a triad of orthogonally mounted rate gyros, rendering the angular velocity of the vehicle available through the biased rate gyros measurements  $\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \mathbf{b}_\omega(t)$ , where the bias  $\mathbf{b}_\omega(t) \in \mathbb{R}^{\frac{n(n-1)}{2}}$  is assumed constant and unknown. Taking this into account, it is possible to assemble the system

$$\begin{cases} \dot{\mathbf{p}}_i(t) = -\mathbf{S}(\boldsymbol{\omega}(t)) \mathbf{p}_i(t) - \mathbf{v}(t), \forall i \in \mathcal{M} \\ \dot{\mathbf{b}}_\omega(t) = \mathbf{0} \\ \mathbf{y}_i(t) = \mathbf{p}_i(t), \forall i \in \mathcal{M}_O \\ \boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \mathbf{b}_\omega(t) \end{cases}, \quad (2)$$

where  $\mathbf{p}_i(t) \in \mathbb{R}^n$  is the position of a landmark,  $\mathbf{y}_i(t) \in \mathbb{R}^n$  is the corresponding measurement, and  $\mathbf{v}(t) \in \mathbb{R}^n$  is the velocity of the vehicle, all expressed in the  
 160 body-fixed frame. The last two quantities are measured. This system can be transformed to incorporate  $\boldsymbol{\omega}_m(t)$  in the dynamics of the landmarks, using the property  $\mathbf{S}(\mathbf{a})\mathbf{b} = \bar{\mathbf{S}}^T(\mathbf{b})\mathbf{a}$ , where  $\bar{\mathbf{S}}^T(\mathbf{b})$  is defined in (1).

The linear velocity and the angular measurement bias constitute the vehicle state, denoted by  $\mathbf{x}_V(t) := [\mathbf{v}^T(t) \ \mathbf{b}_\omega^T(t)]^T \in \mathbb{R}^{nv}$ . Both are assumed, in a deterministic setting, as constant. In the adopted filtering framework, however, the inclusion of state disturbances allows to consider them as slowly time-varying. The landmarks in each set can be stacked in state vectors denoted as  $\mathbf{x}_{O_i}(t) = \mathbf{p}_i(t)$ ,  $i \in \mathcal{M}_O$  and  $\mathbf{x}_{U_i}(t) = \mathbf{p}_i(t)$ ,  $i \in \mathcal{M}_U$ , which together form the landmark state vector  $\mathbf{x}_M(t) = [\mathbf{x}_O^T(t) \ \mathbf{x}_U^T(t)]^T \in \mathbb{R}^{n_o+n_u}$ . This is the framework derived in [7] for the sensor-based simultaneous localization and mapping algorithm proposed therein to estimate the map, the angular velocity bias, and the linear velocity of the vehicle. However, in this paper the linear velocity will act as an input and therefore will not be present in the full system state,  $\mathbf{x}_T(t) = [\mathbf{b}_\omega^T(t) \ \mathbf{x}_M^T(t)]^T$ . The full input vector is  $\mathbf{u}(t) = [\mathbf{v}^T(t) \ \boldsymbol{\omega}^T(t)]^T$ ,

with the full system dynamics reading as

$$\begin{cases} \dot{\mathbf{x}}_T(t) = \mathbf{A}_T(\mathbf{y}(t), \mathbf{x}_U(t), \mathbf{u}(t))\mathbf{x}_T(t) + \mathbf{B}_T\mathbf{v}(t) \\ \mathbf{y}(t) = \mathbf{x}_O(t) \end{cases}, \quad (3)$$

with

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

and  $\mathbf{B}_T = [\mathbf{0}_{n \times \frac{1}{2}n(n-1)} \quad \mathbf{I}_n \quad \cdots \quad \mathbf{I}_n]^T$ , where

$$\mathbf{A}_{MV}(\mathbf{y}(t), \mathbf{x}_U(t)) = [\bar{\mathbf{S}}(\mathbf{p}_1(t)) \quad \cdots \quad \bar{\mathbf{S}}(\mathbf{p}_m(t))]^T$$

and  $\mathbf{A}_M(\mathbf{u}(t)) = \text{diag}(-\mathbf{S}(\boldsymbol{\omega}_m(t)), \dots, -\mathbf{S}(\boldsymbol{\omega}_m(t)))$ . From (3) it follows that the system may be expressed in a way similar to the usual linear system form.

165 However, it can be seen that the system above is nonlinear, as the dynamics matrix depends on the landmarks that constitute the state, as well as on the input. On the other hand, it must be noted that, as  $\mathbf{y}(t) = \mathbf{x}_O(t)$ , the dynamics matrix can be written as a function of the system output and the non-visible landmarks.

170 For the purpose of observer design, the perturbed version with additive disturbances of (3) is considered. The resulting system has process noise  $\boldsymbol{\xi}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Xi})$  and measurement noise  $\boldsymbol{\theta}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$ . The validity of this approach for sensor-based SLAM has been shown with experimental results in [38]. Furthermore, it is the standard procedure in (E)KF-based SLAM filters [10].

## 175 2.2. System observability and observer convergence

In [6] and [7], the version of system (3) with the velocity as part of the state is studied for observability purposes and conditions for its observability and uniform complete observability are found. These include, for example, the existence of two landmarks (2-D) or three landmarks that define a plane (3-D) as sufficient conditions. The observability analysis leads towards the design of a state observer, such as the linear time-varying (LTV) Kalman filter, with globally exponentially stable error dynamics. It is shown also that this observer is also an observer for the nominal nonlinear system (2) while converging exponentially to the true state.

185 In those works, the linear velocity was one of the states not directly observed (the rate-gyro bias being the other). Now, the linear velocity is an input, and therefore the observability restrictions are lightened. This is done to let the definition of the trajectory of the vehicle free, depending upon a control strategy which provides linear and angular velocities.

190

## 2.3. Applications and scenarios

The short overview of sensor-based range-and-bearing SLAM presented in this section only provides an introduction to the approach and the necessary

information for the later stages of this paper. This overview is focused on  
 195 system and observer design, which are based on a generic kinematic model  
 that can be applied to numerous vehicles, missions, and scenarios as long as  
 it is possible to determine the positions of landmarks relative to the vehicle.  
 This usually requires the existence of an RGB-D camera, such as the *Microsoft*  
*Kinect* for three-dimensional environments, or a LiDAR, when working on two  
 200 dimensions. Both options have been used in SLAM experiments detailed in [7]  
 and [6], respectively, where the platform of choice were aerial vehicles, namely  
 quadrotors. However, other scenarios can be considered, such as ground robots  
 [9] or marine vehicles [39].

#### 2.4. Problem Statement

205 The problem considered in this paper is that of finding a control strategy that  
 provides  $\mathbf{v}(t)$  and  $\boldsymbol{\omega}(t)$  such that the uncertainty in the Kalman filters derived  
 in [6] and [7] is optimized, i.e., it is reduced to a minimum. This fits under the  
 problem of active exploitation in simultaneous localization and mapping, i.e.,  
 optimal motion planning to reduce the uncertainty on the estimates, as depicted  
 210 in Figure 2, where this work is contextualized within an active SLAM algorithm.  
 Towards that goal, the first problem tackled is to design a control strategy for a  
 generic class of systems similar to (3) that can be applied to a variety of SLAM  
 problems, such as range-and-bearing [7, 6], range-only [8], and bearing-only [9],  
 as well as to other navigation problems [39, 41]. This then leads to solutions  
 215 particular to the range-and-bearing SLAM problem, as described in this section.

### 3. Optimal Motion Planning for Uncertainty Reduction using an indirect method

220 In this section, the problem of optimal motion planning for uncertainty reduction  
 for a generic class of systems is tackled by means of the Pontryagin  
 minimum principle (PMP) for cost functionals with fixed time, free endpoint,  
 and terminal cost. The following subsection addresses this issue.

#### 3.1. The Pontryagin Minimum Principle

The PMP, as stated in [36, Theorem 5-11], is a very powerful and elegant  
 result in the field of optimal control, even though it only establishes necessary  
 conditions for the optimality of a control law, as it transforms a potentially  
 cumbersome optimization problem into an ordinary differential equation with  
 two-point boundary conditions, provided that a minimum for the Hamiltonian is  
 found. This result applies for a very general class of systems that can be driven  
 by an input. Its most popular application is the linear case with a quadratic  
 cost, better known as the linear quadratic regulator (LQR) [36]. However, it can  
 also be applied to optimal estimation, as advanced in [35]. The Kalman filter  
 can be derived using the PMP and a terminal cost functional depending on the  
 trace of the filter covariance  $\mathbf{P}(t)$ . This is done by defining a matrix costate  $\boldsymbol{\Lambda}(t)$

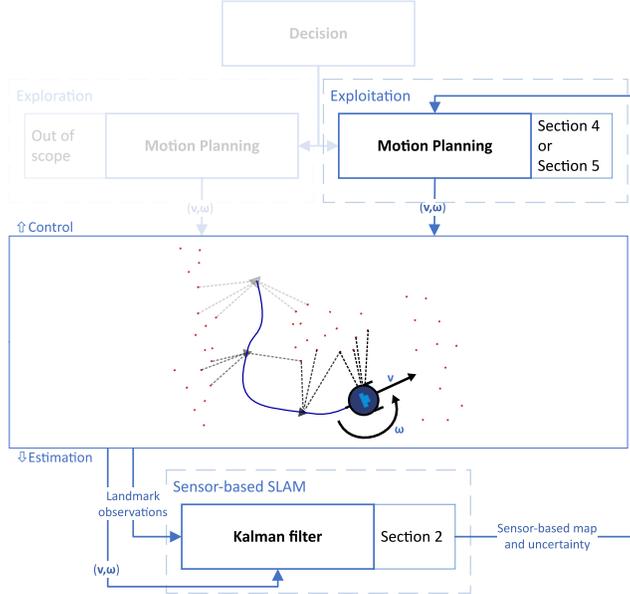


Figure 2: The flow of information on a complete active SLAM algorithm. The processes addressed in this paper have the respective sections associated, the remaining, outside of the scope of this paper, are dimmed down.

whose elements serve as costates for the elements of the filter covariance, and, through the PMP, finding the “control”, i.e., the filter gain, that minimizes the trace of the covariance weighted by some matrix  $\mathcal{M}$ . Based on the innovative approach proposed by Athans in [35], Hussein [34] proposed a filter for linear systems where the covariance of the measurement noise  $\Theta(t)$  depends upon the state of an underlying linear dynamical system. There, the derivation of the Kalman filter found in [35] was combined with the traditional optimal control formulation, by using the input  $\mathbf{u}(t)$  of the underlying system as another degree-of-freedom (along with the Kalman gain). To solve the problem, a new costate  $\lambda(t)$  is added as a counterpart for the dynamics of the underlying system, and the PMP is used to provide necessary conditions for the optimal state trajectory of the underlying nonlinear system that minimizes the filter covariance. This results in the cost functional

$$J(\mathbf{u}(t)) = \text{tr}(\mathcal{M}\mathbf{P}(t_f)) + \int_{t_0}^{t_f} \left( \frac{1}{2} \mathbf{u}^T(t) \mathcal{R} \mathbf{u}(t) + \text{tr}(\mathcal{Q}\mathbf{P}(t)) \right) dt \quad (4)$$

where  $\mathcal{R}$  and  $\mathcal{M}$  are positive-definite matrices,  $\mathcal{Q}$  is a positive semi-definite matrix, and  $t \in [t_0, t_f]$ .<sup>1</sup> The control problem at hand is to find an admissible function  $\mathbf{u}(t) \in \mathbb{R}^{n_u}$  that minimizes this cost functional, i.e.,

$$\mathbf{u}^*(t) = \arg \min_{\mathbf{u}(t)} J(\mathbf{x}(t_0), \mathbf{P}(t_0), t_0, \mathbf{u}(t))$$

subject to the dynamics of the covariance  $\mathbf{P}(t)$  and of the underlying system. For this problem, Hussein proposes the Hamiltonian

$$\begin{aligned} H(\mathbf{P}(t), \mathbf{x}(t), \boldsymbol{\lambda}(t), \boldsymbol{\Lambda}(t), \mathbf{u}(t)) &= \text{tr} \left( \boldsymbol{\Lambda}^T(t) \dot{\mathbf{P}}(t) \right) + \\ &+ \boldsymbol{\lambda}^T(t) \dot{\mathbf{x}}(t) + \frac{1}{2} \mathbf{u}^T(t) \mathcal{R} \mathbf{u}(t) + \text{tr} (\boldsymbol{\mathcal{Q}} \mathbf{P}(t)), \end{aligned} \quad (5)$$

and using the PMP he manages to solve analytically the sufficient conditions for minimization of a scalar field  $H(\mathbf{u})$  [42, Theorem 2.4],

$$\nabla_{\mathbf{u}} H(\mathbf{u}(t)) \Big|_* = 0 \quad (6)$$

and

$$\frac{\partial^2}{\partial \mathbf{u} \partial \mathbf{u}^T} H(\mathbf{u}(t)) \Big|_* \succ \mathbf{0}, \quad (7)$$

and to find expressions for the costates.

225 In this section, a control strategy is proposed that applies to a special kind of system that can be related to the problem introduced above. In order to lighten the notation, the dependence on time of all the variables will be left implicit except when it is evaluated in a particular time instant.

### 3.2. Control strategy for a class of nonlinear systems

Let  $\mathbf{x}(t) \in \mathbb{R}^n$  be the state of a system with nonlinear dynamics

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}) \end{cases}, \quad (8)$$

where  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  can be written as  $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{A}(\mathbf{x}, \mathbf{u})\mathbf{x} + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{u}$ . Suppose that this system accepts a transformation  $(\mathbf{x}_T, \mathbf{y}_T) = \mathbf{T}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n_T} \times \mathbb{R}^{n_{yT}}$  that results in a system affine in its state that can be regarded as LTV

$$\begin{cases} \dot{\mathbf{x}}_T = \mathbf{A}_T(\mathbf{y}, \mathbf{u})\mathbf{x}_T + \mathbf{B}_T(\mathbf{y}, \mathbf{u})\mathbf{u} \\ \mathbf{y}_T = \mathbf{C}_T(\mathbf{y}, \mathbf{u})\mathbf{x}_T \end{cases}, \quad (9)$$

and whose dynamics mimic the dynamics of (8). If conditions for the uniform complete observability of the pair  $(\mathbf{A}_T(\mathbf{y}, \mathbf{u}), \mathbf{C}_T(\mathbf{y}, \mathbf{u}))$  are verified, then a Kalman filter can be designed for system (9) with globally exponentially stable error dynamics [43]. Furthermore, the transformation  $\mathbf{T}(\mathbf{x}, \mathbf{y})$  is assumed to respect the conditions in [44], and, as such, this observer can also be used as an observer for the nonlinear system (8). Adding perturbation noise to both the dynamics and the output equations in (9), and denoting as  $\mathbf{P}(t) \in \mathbb{R}^{n_T \times n_T}$  the covariance of the estimation error of  $\mathbf{x}_T$ ,  $\boldsymbol{\Xi}$  as the covariance of the perturbed version of (9), and  $\boldsymbol{\Theta}(\mathbf{y}, \mathbf{u})$  as the covariance of the perturbed output  $\mathbf{y}_T$  that may depend on the input and output of the system, the dynamics of the covariance of the estimation error are

$$\begin{aligned} \dot{\mathbf{P}} &= \mathbf{A}_T(\mathbf{y}, \mathbf{u})\mathbf{P} + \mathbf{P}\mathbf{A}_T^T(\mathbf{y}, \mathbf{u}) + \boldsymbol{\Xi} \\ &- \mathbf{P}\mathbf{C}_T^T(\mathbf{y}, \mathbf{u})\boldsymbol{\Theta}^{-1}(\mathbf{y}, \mathbf{u})\mathbf{C}_T(\mathbf{y}, \mathbf{u})\mathbf{P}. \end{aligned} \quad (10)$$

230 As they depend on the input and output of the system to be estimated, different inputs (which may lead to a diversity of outputs) will induce different levels of uncertainty in the estimation. The following results address this issue.

**Lemma 1.** *If  $\mathbf{u}^*(t)$  is a local minimizer of the Hamiltonian (5), and  $\frac{\partial^2 H}{\partial \mathbf{u} \partial \mathbf{u}^T}$  exists and is continuous in an open neighborhood of  $\mathbf{u}^*(t)$ , then  $\mathbf{u}^*(t)$  is the solution of*

$$\begin{aligned}
-\mathcal{R}\mathbf{u} &= \mathbf{g}\left(\mathbf{x}\boldsymbol{\lambda}^T, \frac{\partial \mathbf{A}}{\partial \mathbf{u}}\right) - \mathbf{g}\left(\mathbf{P}(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T)\mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}, \frac{\partial \mathbf{C}_T}{\partial \mathbf{u}}\right) \\
&+ \mathbf{g}\left(\mathbf{u}\boldsymbol{\lambda}^T, \frac{\partial \mathbf{B}}{\partial \mathbf{u}}\right) + \mathbf{g}\left(\mathbf{P}(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T), \frac{\partial \mathbf{A}_T}{\partial \mathbf{u}}\right) + \\
&+ \mathbf{B}^T\boldsymbol{\lambda} + \mathbf{g}\left(\boldsymbol{\Theta}^{-1}\mathbf{C}_T\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}, \frac{\partial \boldsymbol{\Theta}}{\partial \mathbf{u}}\right)
\end{aligned} \tag{11}$$

with the auxiliary function  $\mathbf{g}(\boldsymbol{\Gamma}, \boldsymbol{\Upsilon}) \in \mathbb{R}^{q \times 1}$  defined in (A.2). The control  $\mathbf{u}^*(t)$  must also satisfy

$$\left. \frac{\partial^2 H(\mathbf{u}(t))}{\partial \mathbf{u} \partial \mathbf{u}^T} \right|_* \succeq 0. \tag{12}$$

Moreover, if  $\mathbf{u}^*(t)$  is the solution of (11) and the Hessian of (5) with respect to  $\mathbf{u}(t)$  evaluated at  $\mathbf{u}^*(t)$  is definite positive, then  $\mathbf{u}^*(t)$  is a strict minimizer of the Hamiltonian (5).

*Proof.* Consider the substitution of (8) and (10) into (5). This yields

$$\begin{aligned}
H(\mathbf{x}, \mathbf{P}, \boldsymbol{\lambda}, \boldsymbol{\Lambda}) &= \boldsymbol{\lambda}^T \mathbf{A}\mathbf{x} + \boldsymbol{\lambda}^T \mathbf{B}\mathbf{u} + \mathbf{u}^T \mathcal{R}\mathbf{u} + \text{tr}(\boldsymbol{\mathcal{Q}}\mathbf{P}) \\
&+ \text{tr}(\boldsymbol{\Lambda}^T \mathbf{A}_T \mathbf{P}) + \text{tr}(\boldsymbol{\Lambda}^T \mathbf{P} \mathbf{A}_T^T) \\
&+ \text{tr}(\boldsymbol{\Lambda}^T \boldsymbol{\Xi}) - \text{tr}(\boldsymbol{\Lambda}^T \mathbf{P} \mathbf{C}_T^T \boldsymbol{\Theta}^{-1} \mathbf{C}_T \mathbf{P}).
\end{aligned} \tag{13}$$

Recall now the sufficient conditions for the minimum of a scalar field, as expressed by (6) and (7). Computing the gradient of the Hamiltonian and setting it to zero, while using the chain rule (A.1), see Appendix AppendixA, and matrix gradient identities [45], yields the equation in (11). The condition (12) is simply the second sufficient condition, (7), for minimization of a scalar function applied to the Hamiltonian. This concludes the proof of the second part of the lemma. The first part comes directly from the second order necessary conditions for the minimization of a functional [42, Theorem 2.3], i.e., (6) and the positive semi-definiteness of the Hessian of the Hamiltonian.  $\square$

245 **Theorem 1.** *Let  $\mathbf{u}^*(t)$  be an admissible control which transfers  $(\mathbf{x}_0, \mathbf{P}_0, t_0)$  to the target set  $\mathbb{R}^{n_x} \times \mathbb{R}^{n_T \times n_T} \times \{t_f\}$ . Let  $\mathbf{x}^*(t)$  and  $\mathbf{P}^*(t_0)$  be the trajectories of (8) and (10) corresponding to  $\mathbf{u}^*(t)$ , originating at  $(\mathbf{x}_0, \mathbf{P}_0, t_0)$  and meeting the target set at  $t_f$ . If  $\mathbf{u}^*(t)$  is optimal for the cost functional (4), then it is necessary that there exist functions  $\boldsymbol{\lambda}^*(t)$  and  $\boldsymbol{\Lambda}^*(t)$  such that:*

(i) the vector costate  $\boldsymbol{\lambda}^*(t)$  satisfies

$$\begin{aligned}
\dot{\boldsymbol{\lambda}} &= -\mathbf{A}^T \boldsymbol{\lambda} - \mathbf{g}\left(\mathbf{x}\boldsymbol{\lambda}^T, \frac{\partial \mathbf{A}}{\partial \mathbf{x}}\right) - \mathbf{g}\left(\mathbf{u}\boldsymbol{\lambda}^T, \frac{\partial \mathbf{B}}{\partial \mathbf{x}}\right) \\
&\quad - \mathbf{g}\left(\mathbf{P}(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T), \frac{\partial \mathbf{A}_T}{\partial \mathbf{x}}\right) \\
&\quad + \mathbf{g}\left(\mathbf{P}(\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T)\mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}, \frac{\partial \mathbf{C}_T}{\partial \mathbf{x}}\right) \\
&\quad - \mathbf{g}\left(\boldsymbol{\Theta}^{-1}\mathbf{C}_T\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}, \frac{\partial \boldsymbol{\Theta}}{\partial \mathbf{x}}\right)
\end{aligned} \tag{14}$$

250 with boundary condition  $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ ;

(ii) the matrix costate  $\boldsymbol{\Lambda}^*(t)$  associated with the covariance  $\mathbf{P}(t)$  satisfies

$$\begin{aligned}
\dot{\boldsymbol{\Lambda}} &= -\left(\mathbf{A}_T - \mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}\mathbf{C}_T\right)^T \boldsymbol{\Lambda} \\
&\quad - \boldsymbol{\Lambda}\left(\mathbf{A}_T - \mathbf{P}\mathbf{C}_T^T\boldsymbol{\Theta}^{-1}\mathbf{C}_T\right) - \boldsymbol{\mathcal{Q}}
\end{aligned} \tag{15}$$

with boundary condition  $\boldsymbol{\Lambda}(t_f) = \boldsymbol{\mathcal{M}}$ ; and

(iii)  $\mathbf{u}^*(t)$  is a solution of (11), verifying the condition (12).

*Proof.* The first and second conditions come directly from the first condition of [36, Theorem 5-11]. Recall the particular definition of the Hamiltonian given by (13). Computing its gradient with respect to  $\mathbf{x}(t)$  yields a differential equation for  $\boldsymbol{\lambda}(t)$  and its gradient with respect to  $\mathbf{P}(t)$  will yield a differential equation for  $\boldsymbol{\Lambda}(t)$ . In (13), the parcels in  $\boldsymbol{\Xi}$ ,  $\boldsymbol{\mathcal{R}}$  and  $\boldsymbol{\mathcal{Q}}$  do not depend on either  $\mathbf{x}(t)$  or  $\mathbf{P}(t)$ , and therefore do not contribute. Using the chain rule (A.1), see the Appendix AppendixA, and properties of the derivatives of traces with respect to matrices, it is a matter of tedious computation to reach (14). Notice that, even though matrices with subscript  $(\cdot)_T$  depend only on  $\mathbf{y}(t)$  and  $\mathbf{u}(t)$ , there is a direct explicit relation between  $\mathbf{y}(t)$  and  $\mathbf{x}(t)$  expressed by  $\mathbf{y}(t) = \mathbf{h}(\mathbf{x})$  and therefore the terms in  $(\cdot)_T$  do not vanish in (14). The same reasoning is applied to expand  $\dot{\boldsymbol{\Lambda}} = -\frac{\partial}{\partial \mathbf{P}}H(\mathbf{x}, \mathbf{P}, \boldsymbol{\lambda}, \boldsymbol{\Lambda})$  and yield (15). As for the boundary conditions, consider those established by the PMP while applied to the problem at hand,

$$\boldsymbol{\lambda}(t_f) = \frac{\partial}{\partial \mathbf{x}}\Psi(\mathbf{x}(t), \mathbf{P}(t))\Big|_{t=t_f}$$

and

$$\boldsymbol{\Lambda}(t_f) = \frac{\partial}{\partial \mathbf{P}}\Psi(\mathbf{x}(t), \mathbf{P}(t))\Big|_{t=t_f}.$$

The terminal cost here is  $\Psi(\mathbf{x}(t), \mathbf{P}(t))\Big|_{t=t_f} = \text{tr}(\boldsymbol{\mathcal{M}}\mathbf{P}(t_f))$ , which only depends on  $\mathbf{P}(t_f)$ . Therefore, the terminal conditions are  $\boldsymbol{\lambda}(t_f) = \mathbf{0}$  and  $\boldsymbol{\Lambda}(t_f) = \boldsymbol{\mathcal{M}}$ . The final condition is a direct consequence of the last condition in [36, Theorem 5-11] (the Hamiltonian has an absolute minimum as a function of the input at  $\mathbf{u} = \mathbf{u}^*(t)$ ) and Lemma 1.  $\square$

255

The previous result establishes necessary conditions to find the control law that minimizes the uncertainty of the estimates provided by the Kalman filter when applied to systems of the form (8). A variety of estimation problems falls under this category, for example, sensor-based simultaneous localization and mapping with measurements of ranges only, bearings only, or mixed, source-localization, and so forth. The following section explores this line of thought.

In this section and the following, it becomes apparent that working with the *A-opt* criterion greatly simplifies the computations. If, as proposed in [25], the *D-opt* criterion was used, the traces in the Hamiltonian (5) would be replaced by determinants, which, given that  $\frac{d}{d\mathbf{P}} \det(\mathbf{P}) = \det(\mathbf{P})\mathbf{P}^{-1}$ , would render the computation of the costates and  $\mathbf{u}^*$  significantly harder, even taking into account the monotonicity and performance advantages that it might lend to the solution.

#### 4. Uncertainty reduction in sensor-based SLAM

Building on the results presented in the previous section, the objective of this section is to apply the uncertainty reduction control strategy of Theorem 1 to the sensor-based SLAM algorithm exposed in Section 2. The system designed therein has only one input, the angular velocity. The linear velocity is modelled as constant in the body-fixed frame and is one of the states to be estimated. However, to achieve a more flexible solution, the system is modified to leave the linear velocity in the body-fixed frame as another input. This results in the following nonlinear system,

$$\begin{cases} \dot{\mathbf{p}}_i(t) = -\mathbf{S}(\boldsymbol{\omega}(t)) \mathbf{p}_i(t) + \mathbf{B}_v \mathbf{v}(t), \forall i \in \mathcal{M} \\ \dot{\mathbf{b}}_\omega(t) = \mathbf{0} \\ \mathbf{y}_j(t) = \mathbf{p}_j(t), \forall j \in \mathcal{M}_O \end{cases}, \quad (16)$$

which serves as the basis for the work on this section. The landmarks  $\mathbf{p}_i$  can be two- or three-dimensional, the angular rate bias is  $\mathbf{b}_\omega(t) \in \mathbb{R}^{\frac{1}{2}n(n-1) \times 1}$  with  $n$  being the dimensionality of the problem, the linear velocity input has dimension  $n_v$ , and  $\mathbf{B}_v \in \mathbb{R}^{n \times n_v}$  is equal to  $-\mathbf{I}_n$  if the vehicle can move in all directions, or, for example,  $\mathbf{B}_v = [-1 \quad \mathbf{0}_{1 \times n-1}]^T$  if the vehicle has only horizontal movement and cannot move sideways ( $n_v = 1$ ).

Before proceeding, it is necessary to take into account that in a simultaneous localization and mapping framework not all the landmarks are visible in every instant. This means that, for the derivation in this section, the reduced version of the RB-SLAM system that does not include the non-visible landmarks is used. Therefore, in this section, the following simplifying hypothesis is needed.

**Assumption 1.** *All landmarks are always visible in the interval  $[t_0, t_f]$ , i.e., the set  $\mathcal{M}_U$  is empty and the set  $\mathcal{M}_O$  does not vary in time.*

With this assumption in mind, and following the approach explained in [38] and references therein, it is possible to write a system that can be regarded as

LTV, but with the linear velocity as an input. The resulting system is

$$\begin{cases} \dot{\mathbf{x}}_T(t) = \mathbf{A}_T(\mathbf{y}(t), \boldsymbol{\omega}_m(t))\mathbf{x}_T(t) + \underbrace{\begin{bmatrix} \mathbf{0}_{n_\omega \times n_v} \\ \mathbf{B}_v \\ \vdots \\ \mathbf{B}_v \end{bmatrix}}_{\mathbf{B}_T} \mathbf{v}(t) + \boldsymbol{\xi}(t) \\ \mathbf{y}_T = \underbrace{\begin{bmatrix} \mathbf{0}_{n_O \times n_\omega} & \mathbf{I}_{n_O} \end{bmatrix}}_{\mathbf{C}_T} \mathbf{x}_T(t) + \boldsymbol{\theta}(t) \end{cases}, \quad (17)$$

with

$$\mathbf{A}_T(\mathbf{y}(t), \boldsymbol{\omega}_m(t)) = \left[ \begin{array}{c|ccc} \mathbf{0}_{n_\omega} & & & \\ \hline \bar{\mathbf{S}}^T(\mathbf{y}_1(t)) & -\mathbf{S}(\boldsymbol{\omega}_m(t)) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{S}}^T(\mathbf{y}_{N_O}(t)) & \mathbf{0} & \cdots & -\mathbf{S}(\boldsymbol{\omega}_m(t)) \end{array} \right],$$

where  $\bar{\mathbf{S}}(\mathbf{p}_i(t))$ , defined in (1), is used here to condense the 2-D and 3-D cases with a single notation. Note that uncorrelated additive disturbances were included in  $\boldsymbol{\xi}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Xi})$  and  $\boldsymbol{\theta}(t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Theta})$ , as is customary in the design of Kalman filters [46, Chapter 4]. Since this system has a smaller state than (3), the conditions in which the former is observable should be less restrictive. In fact, applying the same steps as in the proof of [7, Theorem 1] and [6, Theorem 2], it is a simple matter of computation to see that, in 3-D, two non-collinear landmarks or two non-collinear observations of one landmark suffice to render the system observable, and, in 2-D, one non-zero landmark serves the same purpose.

Having dealt with the adaptation of the RB-SLAM systems to allow the linear velocity to be an input, the following summarizes the problem at hand.

**Problem 1.** Consider the Kalman filter for the system (17). Given the matrix differential equation (10) satisfied by the error covariance of the filter with initial condition  $\mathbf{P}_0$ , the underlying nonlinear system (16) whose output alters the filter dynamics, the terminal time  $t_f$ , and the cost functional (4), determine the input  $\mathbf{u}(t)$ ,  $t_0 \leq t \leq t_f$ , so as to minimize the cost functional.

The first step is to check whether the system (16) fits under the category of system (8). Rewriting it results in a system of the form

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}(\mathbf{u})\mathbf{x} + \mathbf{B}\mathbf{v} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \quad (18)$$

where the state is  $\mathbf{x} = [\mathbf{b}_\omega^T \quad \mathbf{p}_1^T \quad \cdots \quad \mathbf{p}_{N_O}^T]^T$ , the dynamics matrix is

$$\mathbf{A}(\mathbf{u}) = \begin{bmatrix} \mathbf{0} & & \mathbf{0} \\ \mathbf{0} & \text{diag}(-\mathbf{S}(\boldsymbol{\omega}), \dots, -\mathbf{S}(\boldsymbol{\omega})) \end{bmatrix},$$

and both the input and output matrices are the same as defined in (17). Combining this information with the system (17) that can be regarded as LTV, it is clear that Theorem 1 applies. This is the subject of the next result, where, to simplify,  $(\cdot)_{N_O+1} := (\cdot)_b$  are the quantities corresponding to the bias.

305 **Theorem 2.** *Let  $\mathbf{u}^*(t)$  be an admissible control which transfers  $(\mathbf{x}_0, \mathbf{P}_0, t_0)$  to the target set  $\mathbb{R}^{n_x} \times \mathbb{R}^{n_T \times n_T} \times \{t_f\}$ . Let  $\mathbf{x}^*(t)$  and  $\mathbf{P}^*(t_0)$  be the trajectories of (18) and (10) corresponding to  $\mathbf{u}^*(t)$ , originating at  $(\mathbf{x}_0, \mathbf{P}_0, t_0)$  and meeting the target set at  $t_f$ . In order for  $\mathbf{u}^*(t)$  to be optimal for the cost functional (4), it is necessary that there exist functions  $\boldsymbol{\lambda}^*(t)$  and  $\boldsymbol{\Lambda}^*(t)$  such that:*

(i) *the vector costate  $\boldsymbol{\lambda}^*(t)$  satisfies*

$$\dot{\boldsymbol{\lambda}}_i = -\mathbf{S}(\boldsymbol{\omega})\boldsymbol{\lambda}_i + \sum_{j=1}^{N_O+1} \mathbf{S}^T(\mathbf{P}_{ij} (\boldsymbol{\Lambda}_{jb} + \boldsymbol{\Lambda}_{bj}^T)) \quad (19)$$

*for the bidimensional case, and*

$$\begin{aligned} \dot{\boldsymbol{\lambda}}_i &= -\mathbf{S}(\boldsymbol{\omega})\boldsymbol{\lambda}_i \\ &\quad - 2 \sum_{j=1}^{N_O+1} \mathbf{S}^{-1}(\text{skew}(\mathbf{P}_{ij} (\boldsymbol{\Lambda}_{jb} + \boldsymbol{\Lambda}_{bj}^T))) \end{aligned} \quad (20)$$

310 *for the tridimensional case. In both situations, the boundary condition is  $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ ;*

(ii) *the matrix costate  $\boldsymbol{\Lambda}^*(t)$  associated with the covariance  $\mathbf{P}(t)$  satisfies (15) with boundary condition  $\boldsymbol{\Lambda}(t_f) = \mathcal{N}_O$ ;*

(iii)  *$\mathbf{u}^*(t)$  is given by*

$$\begin{bmatrix} \mathbf{v}^* \\ \boldsymbol{\omega}^* \end{bmatrix} = -\mathcal{R}^{-1} \begin{bmatrix} \mathbf{z}_v(\boldsymbol{\lambda}) \\ \mathbf{z}_\omega(\mathbf{x}, \mathbf{P}, \boldsymbol{\lambda}, \boldsymbol{\Lambda}) \end{bmatrix} \quad (21)$$

*where the part associated with the linear velocity is*

$$\mathbf{z}_v = \mathbf{B}_v^T \sum_{i=1}^{N_O} \boldsymbol{\lambda}_i \quad (22)$$

*and that associated with the angular velocity is*

$$\begin{aligned} \mathbf{z}_\omega &= - \sum_{i=1}^{N_O} \mathbf{S}(\mathbf{x}_i) \boldsymbol{\lambda}_i \\ &\quad + 2 \sum_{i=1}^{N_O} \sum_{j=1}^{N_O+1} \mathbf{S}^{-1}(\text{skew}(\mathbf{P}_{ij} (\boldsymbol{\Lambda}_{ji} + \boldsymbol{\Lambda}_{ij}^T))). \end{aligned} \quad (23)$$

*Proof.* This result follows directly from Theorem 1 when applied to Problem 1. First, consider the partial derivatives employed as arguments in  $\mathbf{g}(\cdot, \cdot)$  in expressions (11) and (14). Looking at the dependences of the matrices in (16) and (17), it is clear that most of the partial derivatives vanish, yielding

$$\frac{\partial \mathbf{B}}{\partial \mathbf{u}} = \mathbf{0}, \frac{\partial \mathbf{C}_T}{\partial \mathbf{u}} = \mathbf{0}, \frac{\partial \mathbf{R}}{\partial \mathbf{u}} = \mathbf{0}$$

for the partial derivatives with respect to both inputs,

$$\frac{\partial \mathbf{A}}{\partial \mathbf{v}} = \mathbf{0}, \frac{\partial \mathbf{A}_T}{\partial \mathbf{v}} = \mathbf{0}$$

for the partial derivatives with respect to the linear velocity, and

$$\frac{\partial \mathbf{A}}{\partial \mathbf{x}} = \mathbf{0}, \frac{\partial \mathbf{B}}{\partial \mathbf{x}} = \mathbf{0}, \frac{\partial \mathbf{C}_T}{\partial \mathbf{x}} = \mathbf{0}, \frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \mathbf{0}$$

for the partial derivatives with respect to the state of the nonlinear system. This simplifies greatly both the expressions for  $\mathbf{u}^*(t)$  and  $\dot{\boldsymbol{\lambda}}^*(t)$ . Consider now the two partial derivatives that remain in (11). Denoting  $\omega_i(t)$  as the  $i$ -th component of the angular velocity, one obtains

$$\frac{\partial \mathbf{A}}{\partial \omega_i} = \frac{\partial \mathbf{A}_T}{\partial \omega_i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \text{diag}(-\mathbf{S}(\bar{\mathbf{1}}_i)) \end{bmatrix}$$

where  $\bar{\mathbf{1}}_i$  is tridimensional or scalar depending of the dimensions of  $\boldsymbol{\omega}(t)$ . From this, the part of the right-hand side of (11) corresponding to derivatives with respect to the linear velocity input is reduced to the component  $\mathbf{B}^T \boldsymbol{\lambda}$ , that can be rewritten to yield (22). The components of the right-hand side that correspond to derivatives with respect to the angular velocity are more complex, even though only the first and fourth parcels remain. Evaluating it element-by-element, with  $i = 1, \dots, \frac{1}{2}n(n-1)$ , leads to

$$\begin{aligned} z_{\omega_i} = & - \sum_{j=1}^{N_O} \text{tr}(\boldsymbol{\lambda}_j \mathbf{x}_j \mathbf{S}(\bar{\mathbf{1}}_i)) \\ & - \sum_{j=1}^{N_O} \sum_{k=1}^{N_O+1} \text{tr} \left( \left( \boldsymbol{\Lambda}_{jk} + \boldsymbol{\Lambda}_{kj}^T \right) \mathbf{P}_{kj} \mathbf{S}(\bar{\mathbf{1}}_i) \right) \end{aligned} \quad (24)$$

315 which can be further simplified when addressing all the components of  $\mathbf{z}_\omega$  at once, yielding (23). To obtain (21), left-multiply both sides of (11) by  $\boldsymbol{\mathcal{R}}^{-1}$ . Finally, the condition (12) must be verified. Given that in (11) only the left-hand side depends on  $\mathbf{u}$ , this yields  $\frac{\partial^2 H}{\partial \mathbf{u} \partial \mathbf{u}^T} = \boldsymbol{\mathcal{R}} \succ 0$ , which, using Lemma 1, ensures that not only (5) has a minimum at (21), it is indeed the input that strictly minimises the Hamiltonian.

To show that the costate dynamics is given by (19) or (20), consider the differential equation in (14). Removing the parcels that depend on the null

derivatives yields

$$\dot{\boldsymbol{\lambda}} = -\mathbf{A}^T \boldsymbol{\lambda} - \mathbf{g} \left( (\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^T) \mathbf{P}, \frac{\partial \mathbf{A}_T}{\partial \mathbf{x}} \right),$$

which, when evaluated for dimension  $j = 1, \dots, n$  of each component  $\boldsymbol{\lambda}_i$ ,  $i \in \mathcal{M}$ , is

$$\dot{\lambda}_{i_j} = \bar{\mathbf{I}}_j^T \mathbf{S}(\boldsymbol{\omega}) \boldsymbol{\lambda}_i - \sum_{k=1}^{N_O+1} \text{tr} \left( \left( \boldsymbol{\Lambda}_{bk} + \boldsymbol{\Lambda}_{kb}^T \right) \mathbf{P}_{ki} \mathbf{S}^T(\bar{\mathbf{I}}_j) \right).$$

320 To obtain the dynamics for the full  $\boldsymbol{\lambda}_i$ , special considerations must be taken, depending on the dimensionality in analysis. In the bidimensional case, the argument of the trace is scalar and it can be shown that it yields (19). For the tridimensional case, using the same reasoning used to go from (24) to (23), it is possible to obtain (20). It must be noted that the costate associated with  
 325  $\mathbf{b}_\omega(t)$  does not appear in any of the expressions for the input  $\mathbf{u}^*(t)$ , landmark costates  $\boldsymbol{\lambda}_i^*(t)$ ,  $i \in \mathcal{M}$ , or covariance costate  $\boldsymbol{\Lambda}^*(t)$  and therefore there is no need to compute it explicitly. Taking all this into account, using Theorem 1, the proof is concluded.  $\square$

This result provides necessary conditions for the input that allows for optimal uncertainty reduction in the sensor-based SLAM problem. It achieves  
 330 that by transforming the optimization problem into a two-point boundary value problem. However, solving this problem can still be complex. In optimal control problems, the forward-time costate equations are unstable, even in the linear case of the LQR. This is a major problem in shooting methods [47] that are, nevertheless relatively fast. On the other hand, collocation methods [48] are  
 335 very sensitive to bad initial guesses. In the first part of this paper, the two algorithms are combined to solve the BVPs ensuing from the control problem of the current section.

#### 4.1. Solving the Boundary Value Problem

340 Two-point boundary value problems are often encountered in mechanics and optimal control problems, and several numerical methods have been proposed to solve them. In the 1960's shooting methods were the most common tool to solve two-point BVP, including the simple shooting method (SSM) [47] and the multiple simple shooting method (MSSM) [49]. The former consists on  
 345 solving initial value problems with guesses for the unknown initial values and thus trying to find the ones that lead to the correct final time conditions. The other alternative at the time was finite difference method where the unknowns are considered to be the values of the true solution at a number of interior mesh points. The main advantage of the shooting method is that it is generally  
 350 faster than others. However, it requires the ability to integrate the differential equations forward in time, which can be a problem if the equations are so unstable that the solution blows up in the integration time, even for accurate initial guesses. The multiple simple shooting method (MSSM) [49] is an attempt to fuse these approaches. The integration interval is divided into segments

355 and initial value problems are solved in each of them, while the continuity  
of the overall solution is guaranteed by matching the initial condition of one  
segment to the final condition of the previous one. [50] proposed the modified  
simple shooting method (MSSM) that aims at improving the consistency of the  
solutions of the MSM, that may not be true solutions of the BVP. It has been  
360 used in optimal control problems with the PMP [51]. The current tool of choice  
is the collocation method used, for example, in the function `bvp4c` [48] present in  
MATLAB. However, it requires a guess function close to the solution, especially  
for nonlinear problems such as the one at hands. To try to circumvent this  
problem, in this paper shooting methods are used to compute solutions in small  
365 intervals that are combined to serve as an initial guess for the collocation. The  
algorithm is as follows.

---

**Algorithm 1** BVP solver

---

**Input:**  $t_0, t_f, \mathbf{x}_0, \mathbf{P}_0, t_{\delta_1}, t_{\delta_2} \gg t_{\delta_1}$

**Output:**  $\mathbf{x}(t), \mathbf{P}(t), \boldsymbol{\lambda}(t), \boldsymbol{\Lambda}(t), \forall t \in [t_0, t_f]$

Divide the integration time in timesteps of span  $t_{\delta_1}$ .

Solve the control problems of timespan  $t_{\delta_1}$  with SSM, each with initial conditions  $(\mathbf{x}_0, \mathbf{P}_0)$  defined by the final state of the previous segment, and initial costate conditions equal to the initial costate of the previous segment.

**if** a solution cannot be found for a particular segment **then**

    solve the problem with  $\mathcal{R}_\epsilon^{-1} = \epsilon \mathcal{R}^{-1}$ , and iterate  $\epsilon$  from 0 to 1, using the initial conditions found as new guesses

**end if**

Concatenate the solutions by SSM and use them as a guess for the `bvp4c` procedure for  $[t_0, t_f]$ .

**if** the process fails **then**

    solve for the intervals  $t_{\delta_2}$  while maintaining continuity in  $\mathbf{x}(t)$  and  $\mathbf{P}(t)$

**end if**

---

As will be seen in sequel, while this is a theoretically sound solution to the Active SLAM problem, solving the BVP numerically is rather difficult, requiring either short time intervals or very slow motion.

370 *4.2. Simulation results and analysis*

In this section, the results of solving numerically the BVP expressed by (10), (15), (18), (19), and (21) with the respective boundary conditions are presented. Several combinations of parameters are tested and their results analysed. All of the scenarios explored in this section are bidimensional or planar ( $n = 2$   
375 and  $\frac{1}{2}n(n - 2) = 1$ ), and in all of them the control weight is  $\mathcal{R} = \alpha \mathbf{I}$ . For better visualization and to avoid unnecessary computational problems,  $N_O = 3$ , which is enough to guarantee the convergence of the Kalman filter. Given that observability is guaranteed, the filter will converge regardless of the input. For that reason, in this section the results obtained with  $\mathbf{u}(t) = \mathbf{u}^*(t)$  as given by  
380 Theorem 2 are compared with what is obtained when the vehicle does not move at all.

The main scenarios explored in this section have in common the isotropic nature of the parameters, i.e.,  $\mathbf{P}_0 = \mathbf{I}$ ,  $\mathcal{M} = \mathbf{I}$ ,  $\Xi = 0.1\mathbf{I}$ , and  $\Theta = 0.1\mathbf{I}$ . The map has landmarks at  $(3.75, -3.75)$ ,  $(3.75, 3.75)$ , and  $(7.5, 0)$ , meaning that the initial position of the vehicle (at the origin) is 5 meters away from the centroid of the map. The BVP is initialized with these positions and the covariances above. The main variation of parameters is between Case A:  $\mathcal{Q} \succ \mathbf{0}$ ; and Case B:  $\mathcal{Q} = \mathbf{0}$ . This change in cost functional will most likely influence the trajectory followed by the vehicle but not the steady state.

The spacial configuration of both scenarios in the Earth-fixed frame is depicted in Figure 3. There, the blue line represents the path of the vehicle, starting in the origin and moving towards what is seen to be the centroid of the map. The dots inside the ellipses are the landmarks, and the ellipses represent the covariance of estimation error of the filter. The large dashed ellipses are the initial condition, and the smaller bold ellipses represent the final state of  $\mathbf{P}$ : the solid ones are those resulting from applying  $\mathbf{u}^*(t)$  to the system and the dashed ones resulting from not applying any input. Given that the steady states of both strategies are almost identical, only the overall view of Case A is shown in that figure. The main difference between the results of both scenarios is in the evolution of the position of the vehicle in time. Even though the path followed is a direct line from the origin to the map centroid for both cases, when  $\mathcal{Q} = \mathbf{0}$ , the “average” covariance is not taken into account in the cost, and therefore the only intervening parts in the cost functional are the control cost and the terminal

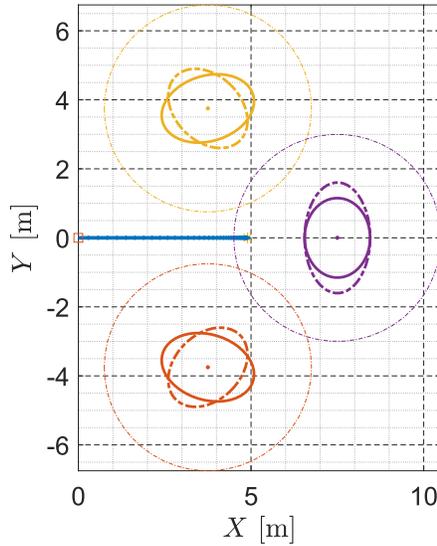
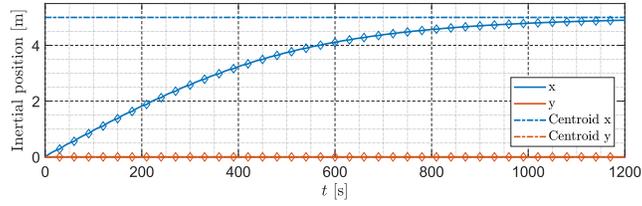
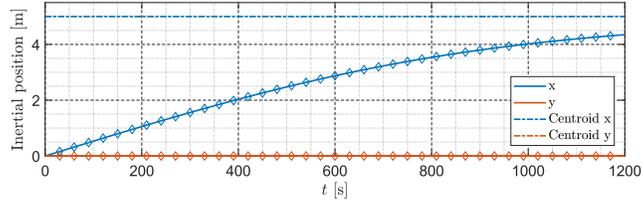


Figure 3: The scenario of the different cases. The blue solid line is the vehicle path, the small triangles indicate the vehicle heading at the end of each segment, the ellipsoids represent the  $3\sigma$  bounds of the landmarks: the solid line is for  $\mathbf{u}(t) = \mathbf{u}^*(t)$ , the dashed bold line for  $\mathbf{u}(t) = \mathbf{0}$ , and the light dashed line is the initial covariance.

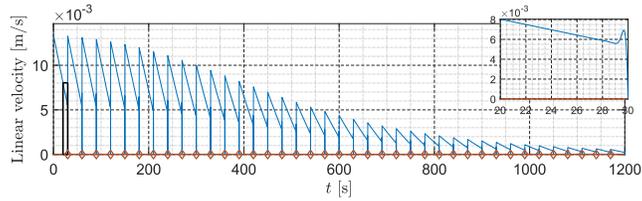


(a) Case A

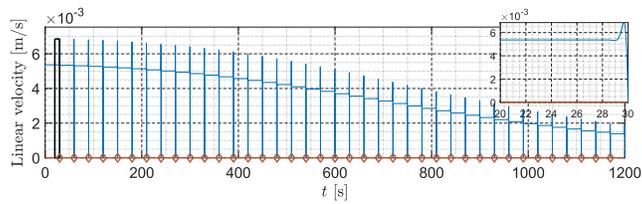


(b) Case B

Figure 4: The trajectory of the vehicle in time for both cases. The dashed line marks the centroid of the map, and the small diamonds mark the segments in which the BVP was solved (30 seconds each).



(a) Case A



(b) Case B

Figure 5: The input linear velocity for both cases (the blue line is the  $x$  coordinate and the red on is the  $y$  coordinate, which in this case is zero at all times). In the top right a close up view of the end of a segment.

405 covariance cost. It is then expected that the control will take the system straight to the position that optimizes the final covariance, i.e., the centroid of the map, in contrast with the more smooth trajectory obtained when  $\mathbf{Q} \succ \mathbf{0}$ . This is confirmed by the results of Figure 4 and Figure 5. Note that, in these cases, the attitude of the vehicle is not changed during the trajectory, as the  $\omega(t)$  input

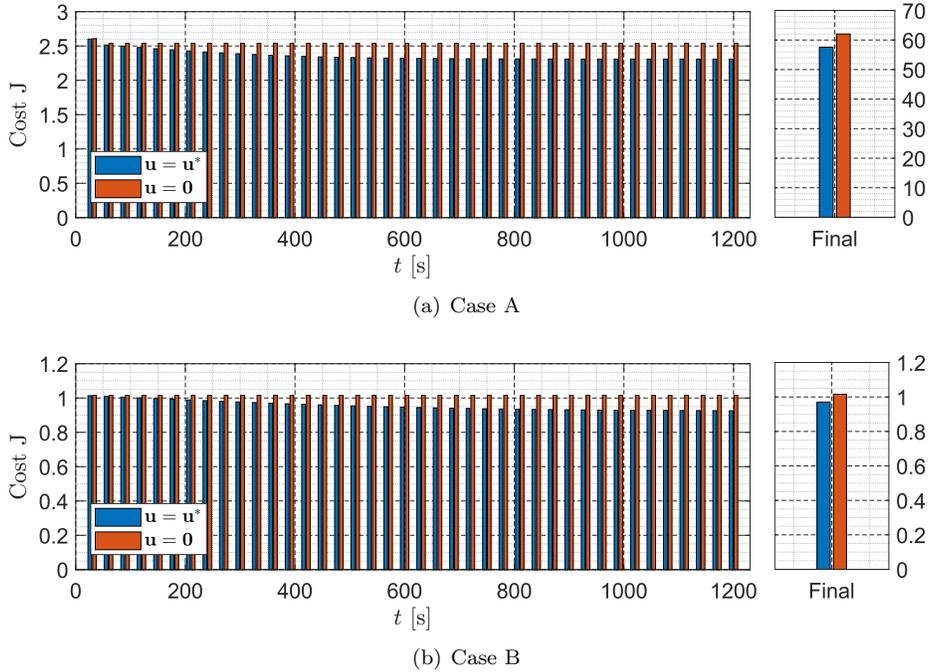


Figure 6: The cost  $J(\mathbf{u})$  computed in each segment for both scenarios. The final cost is computed as if the whole simulation was a single segment.

is zero. As seen in (21) and (22), when  $\mathcal{R}^{-1}$  has no components relating  $\mathbf{z}_\omega$  to  $\mathbf{v}(t)$ , the input linear velocity is completely guided by the costate. This means that the linear velocity in the body-fixed frame will always be zero at the end of the interval of optimization, as that is the boundary condition on the costate. This is verified by the numerical results in Figure 5, where all the segments respect Theorem 2 independently. Even though the necessary conditions do not guarantee optimality, these results not only lead towards lower uncertainty, as seen in Figure 7, but also to a lower cost when compared to not moving. This is confirmed for the two scenarios studied in Figure 6 where the cost of each segment is shown for both  $\mathbf{u}(t) = \mathbf{u}^*(t)$  and  $\mathbf{u}(t) = \mathbf{0}$ . It can be observed that the difference between the two costs is rather small in the first segment (0.27%), which is expectable, since the covariance traces are almost identical during the first segments as shown in Figure 7. However, the difference in the total costs is much larger being around 8%.

A few other scenarios with minor variations were also approached. For example, when different size initial covariances are chosen for each landmark, the result is the same as with equal covariances. This may be related to the exponentially fast convergence of the filter. Also, when the centroid of the map is in line with the vehicle and one of the landmarks, the trajectory goes straight through the landmark. Even though this makes sense, as there is nothing in the

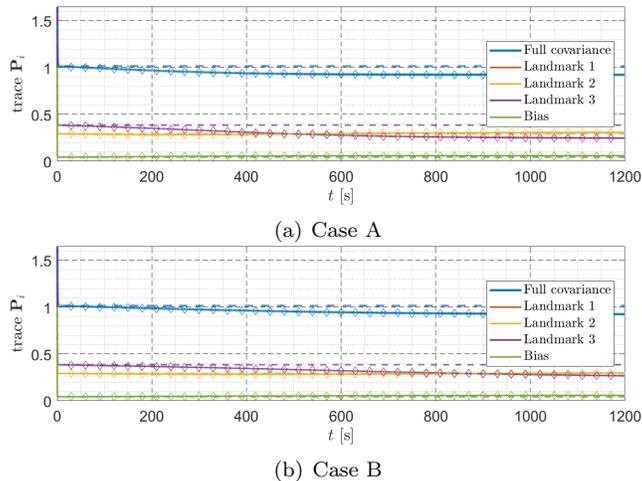


Figure 7: The evolution of the covariance  $\mathbf{P}(t)$  in the different scenarios, for  $\mathbf{u}^*(t)$  (solid line) and  $\mathbf{u}(t) = \mathbf{0}$  (dashed). In blue is the trace of the full covariance and in violet the variance of the measurement bias estimation error. The remaining lines are  $\text{tr}(\mathbf{P}_i)$ ,  $i \in \mathcal{M}$ , with the colors corresponding to the landmarks in Figure 3.

cost functional avoiding this, it should be addressed in future work. Finally, it  
 430 was observed that the time the system takes to go to the centroid of the map  
 is greatly reduced when  $\mathcal{M}$  is increased or  $\mathcal{R}$  decreased. One of the main con-  
 clusions taken from these results is that the system tends to the centroid of the  
 map. However, this may be due to the fact that the measurement covariance is  
 equal for all the landmarks. In fact, a preliminary trial hinted that the vehicle  
 435 tends to a point closer to the landmarks with greater measurement covariance  
 $\Theta$ .

## 5. Uncertainty reduction with direct methods: using piecewise constant inputs

As was observed in the previous section, it was only possible to solve the  
 440 BVP in very restrictive conditions, with rather slow results. For that reason,  
 a different approach is proposed in this section. The idea is to use a direct  
 method for the optimization, while maintaining the cost functional presented  
 previously. For that purpose, several possibilities could be employed ranging  
 from moving to a discrete time model using a discrete Kalman filter, which  
 445 would imply a complete redesign of the problem, to slight adaptations of the  
 current approach to be solved directly. The chosen approach is to discretize only  
 the input functions, maintaining the continuous time nature of the problem as  
 it was proposed.

Consider now that the input function  $\mathbf{u}(t)$  is a piecewise constant function

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_0, & t_0 \leq t < t_1 \\ \vdots & \\ \mathbf{u}_{N-1}, & t_{N-1} \leq t < t_f \end{cases},$$

where  $t_k - t_{k-1} = \frac{t_f - t_0}{N} := T$  for all  $k \in \{1, \dots, N\}$ . In that case, the optimization problem boils down to finding the sequence  $\{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}\}$  that minimizes the cost functional. This can now be rewritten as

$$\mathbf{u}^*(t) = \arg \min_{\mathbf{u}_0, \dots, \mathbf{u}_{N-1}} J(\mathbf{x}(t_0), t_0, \mathbf{u}_0, \dots, \mathbf{u}_{N-1})$$

with the cost functional

$$J(\mathbf{x}(t_0), t_0, \mathbf{u}_0, \dots, \mathbf{u}_{N-1}) = \text{tr}(\mathcal{M}\mathbf{P}(t_f)) + \sum_{k=0}^{N-1} \left( \frac{T}{2} \mathbf{u}_k^T \mathcal{R} \mathbf{u}_k + \int_{t_k}^{t_{k+1}} \text{tr}(\mathcal{Q}\mathbf{P}(t)) dt \right). \quad (25)$$

Even though part of this cost functional is quadratic on the input, the nature of the relation between the covariance (and its integral) with the input sequence is not clear. In order to evaluate this aspect of the optimization problem, a few simple two-dimensional examples were studied. It was observed that, when plotted against the inputs the 1-step cost function has a valley shape, that is preserved when looking at the plots for the 2-step cost function against each pair of inputs  $(v_i, \omega_j)$ ,  $i, j = 1, 2$ . This can be seen in Figure 8, where the cost function was computed with  $\mathcal{R}$  set to zero to remove the influence of the quadratic term. Another interesting observation that further hints at good properties of the cost function is shown in Figure 9. There, the cost function is plotted against the final positions of the vehicle after two steps following the input sequences represented in Figure 8. It is clear from the colour evolution that the map centroid is the final point with lower cost and that the cost function evolves smoothly towards that minimum.

With this in mind, the following step is to use a numerical optimization tool to find the optimal sequence of inputs. Since a closed form for the covariance matrix  $\mathbf{P}(t)$  is not available, it is not feasible to compute an explicit gradient of the cost function. Two options for unconstrained optimization that can be used are simplex algorithms such as the Nelder–Mead simplex method [52], a direct search method that uses only function evaluations – it does not use numerical or analytic gradients – but is not guaranteed to converge to local minima, and gradient-based algorithms such as Quasi-Newton methods that build up curvature information at each iterations while avoiding an explicit numerical computation of the Hessian [53, 54, 55, 56]. Even though other possibilities can be considered, there are readily available implementations of the enumerated methods in MATLAB, respectively the `fminsearch` and `fminunc` functions.

In the following subsection, problems with several different combinations of parameters are solved using `fminunc`.

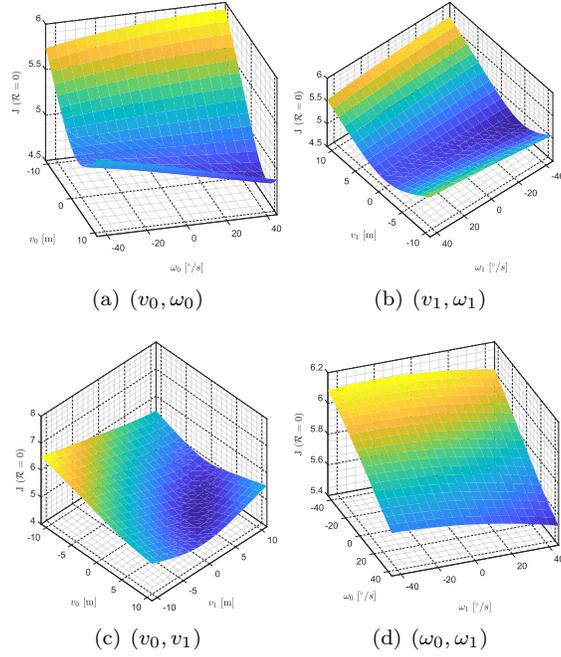


Figure 8: The evolution of the two-step cost function  $J$  in (25) with several input pairs, while fixing the remaining inputs and setting  $\mathcal{R} = \mathbf{0}$ .

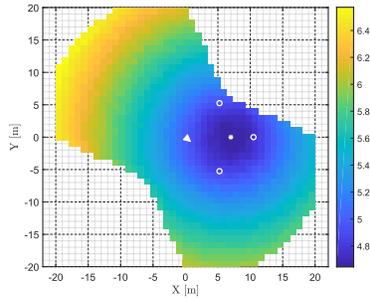


Figure 9: The evolution of the cost function  $J$  (represented in color) in (25).  $X$  and  $Y$  are the final coordinates after two steps following the inputs shown in Figure 8. The initial pose is represented by the white triangle, the white circles are the landmarks, and the white star represents the map centroid.

### 5.1. Simulations

As in Section 4, the scenarios explored in this section are two-dimensional and in all of them the control weight is  $\mathcal{R} = \alpha \mathbf{I}$ . However, while a fully actuated model was used in those simulations, in this section the vehicle is assumed to be restricted to forward linear movement. This provides a far more interesting and

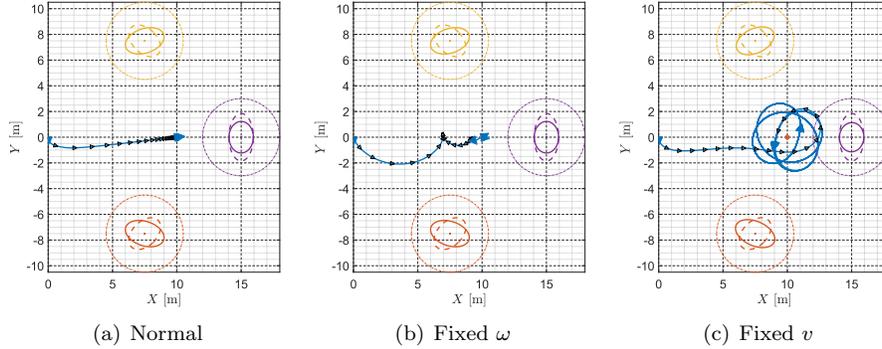


Figure 10: The scenario of the different cases. The blue solid line is the vehicle path, the triangles indicate the vehicle heading, the ellipsoids represent the  $3\sigma$  bounds of the landmarks: the solid line is for  $\mathbf{u}(t) = \mathbf{u}^*(t)$ , the dashed bold line for  $\mathbf{u}(t) = \mathbf{0}$ , and the light is the initial covariance.

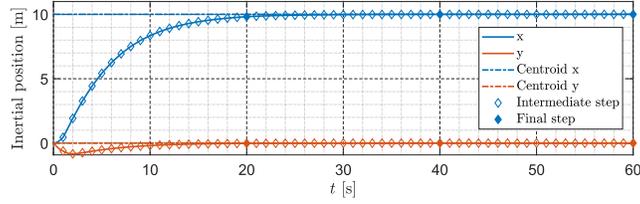
realistic situation. To better explore this difference, the initial heading of the vehicle relative to the centroid of the map (which has been seen to be the destination of the optimized path in the previous sections) is set to  $89^\circ$ . The distance to the centroid is set to 10 m. Furthermore, three different models are provided:

- 485 Normal forward linear and angular velocities are available as inputs for optimization of the trajectory;
- Fixed  $\omega$  the angular velocity is fixed and only the forward linear velocity can be controlled; and
- Fixed  $v$  the forward linear velocity is fixed and only the angular velocity can be controlled.

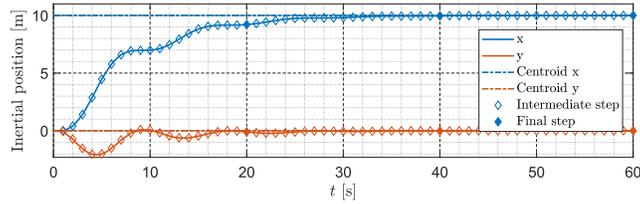
For each of these cases, several simulations were performed with different step sizes  $T$  and number of steps  $N$ . As in the PMP-based solution, the optimization is repeated after  $t_f$  is reached for a determined time. In the cases where the linear velocity is free, the vehicle is shown to eventually stop in under 60 seconds. Naturally, when the linear velocity is fixed and different from zero the vehicle will never stop. In that case, the total time is set to 200 seconds.

The parameters of the Kalman filter are the same as before, i.e.,  $\mathbf{P}_0 = \mathbf{I}$ , and  $\mathbf{\Xi} = 0.1\mathbf{I}$ ,  $\mathbf{\Theta} = 0.1\mathbf{I}$ . The chosen optimization parameters are  $\mathcal{M} = 3\mathbf{I}$ ,  $\mathbf{Q} = 0.5\mathbf{I}$ , and  $\mathcal{R} = 0.05\mathbf{I}$ . The inertial map is now composed of landmarks in  $(7.5, -7.5)$ ,  $(7.5, 7.5)$ , and  $(15, 0)$ . Hence, the initial real state is composed of the body-fixed version of this map (rotated by the initial heading) and the rate-gyro bias, set to zero. For the most part of the simulations presented in this section, the system is initialized with zero error, to better understand the behaviour of the optimization problem. For a more realistic scenario, the section is closed with a larger scale example where the optimization problem is fed with the results of the online filter.

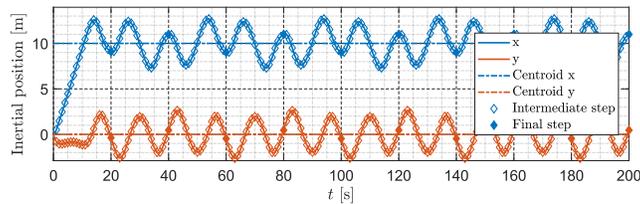
The final result of the main simulations for the three cases is presented in Figure 10. In all its composing figures the size of each step is set to  $T = 1$



(a) Normal



(b) Fixed  $\omega$

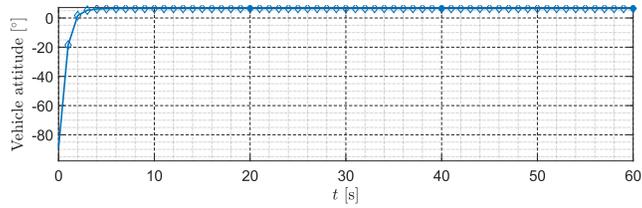


(c) Fixed  $v$

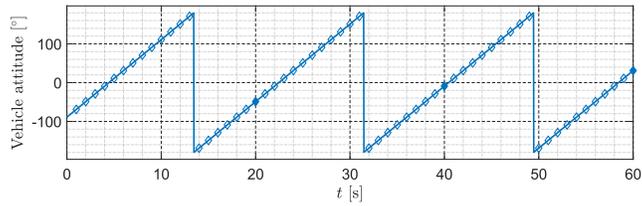
Figure 11: The trajectory of the vehicle in time for the different cases. The dashed lines mark the centroid of the map, and the small rhombuses mark the segments in which the optimization problem was solved (1 second each for the intermediate steps, 20 seconds for each complete optimization).

505 second, and the number of steps optimized at each time is  $N = 20$ . The blue line is the concatenation of each optimal segment of 20 seconds, the ending of which are marked with a filled triangle indicating the orientation of the vehicle at that time. Similarly, the black triangles indicate the orientation of the vehicle in each step of the first optimization interval. The geometric centroid of the map is marked with the orange asterisk, and the ellipsoids represent the  $3\sigma$  bounds of the filter covariances for each landmark: the solid line is the final covariance when the optimal input is used, the thicker dashed lines are the final covariances if the vehicle is stopped at the origin of the map, and the thinner dashed lines are the initial covariances in both cases. For all the situations

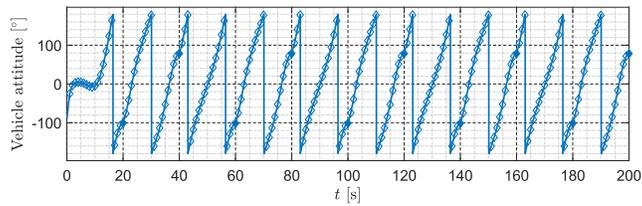
510 depicted, it can be seen that the vehicle moves toward the centroid of the map, as expected. This is achieved by continuously turning in its direction using  $\omega$  when it is available, in opposition to the fully actuated case of Section 4. Even though all these scenarios start with the same initial heading, simulations with headings up to  $90^\circ$  have very similar results, in which the control tends to first



(a) Normal



(b) Fixed  $\omega$

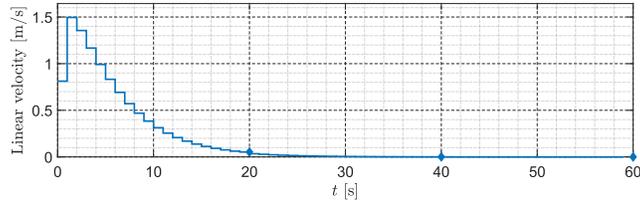


(c) Fixed  $v$

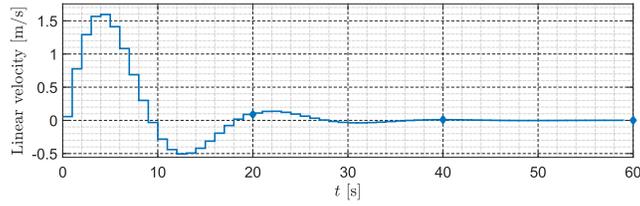
Figure 12: The orientation of the vehicle in time for the different cases. The dashed lines mark the centroid of the map, and the small rhombuses mark the segments in which the optimization problem was solved (1 second each for the intermediate steps, 20 seconds for each complete optimization).

520 correct the heading and then drive the vehicle to the centroid. This of course  
depends on the choice of control weight  $\mathcal{R}$  in the optimization problem. For  
angles greater than  $90^\circ$ , the path of least cost drives the vehicle backwards with  
negative linear velocity unless a constraint on  $\mathbf{v} \geq \mathbf{0}$  is enforced, which can be  
done, for example, with `fmincon`. In that case, the control first rotates  $90^\circ$   
525 counter-clockwise and only on the second step starts translating the vehicle.

Of the two cases with fixed velocities, the fixed linear velocity is especially  
interesting, since it showcases the importance of the region near the centroid  
of the map. In fact, it is observed that the vehicle orbits the centroid with  
three main modes each closer to one of the landmarks. On the fixed angular  
530 velocity case, it is seen that the control only manages to bring the vehicle to  
the centroid by using a negative linear velocity when the angular velocity would  
force the vehicle to move away from the centroid with a positive linear velocity.  
These behaviours can be better observed by looking at Figures 11 and 13. The  
first depicts the optimal trajectories through time and the second the optimal

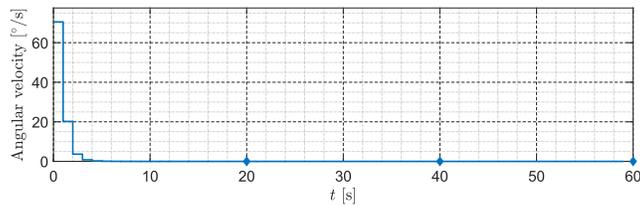


(a) Normal

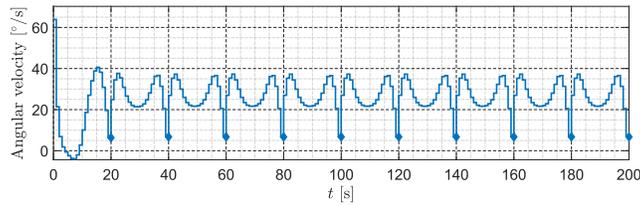


(b) Fixed  $\omega$

Figure 13: The input linear velocity for the different cases.



(a) Normal

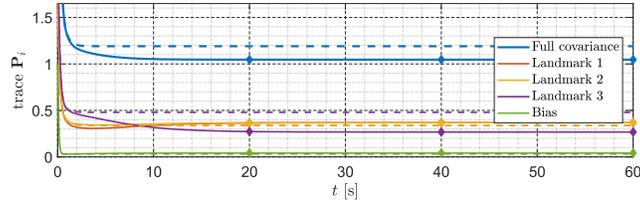


(b) Fixed  $v$

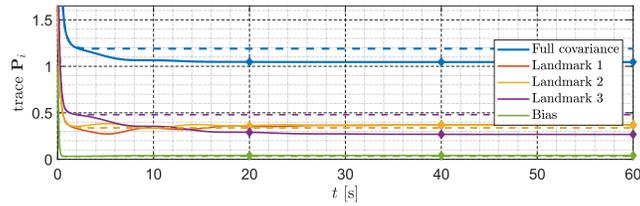
Figure 14: The input angular velocity for the different cases.

535 linear velocity. In these figures, the empty rhombuses are each intermediate  
 step  $k < N$  of the optimization, while the filled ones represent the final step  
 of each optimization process. Note that, while in the PMP-based solution the  
 linear velocity was zero at the end of each optimization interval, that is not the  
 case here.

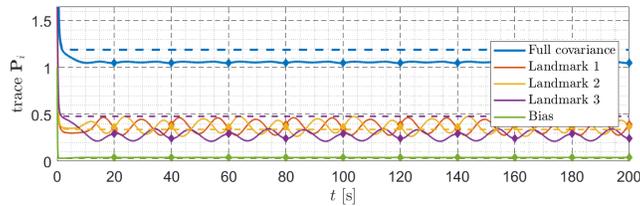
540 Figures 12 and 14 are connected to the attitude of the vehicle, respectively  
 its heading and its angular velocity. These show how the control steers the



(a) Normal



(b) Fixed  $\omega$



(c) Fixed  $v$

Figure 15: The evolution of the covariance  $\mathbf{P}(t)$  in the different scenarios, for  $\mathbf{u}^*(t)$  (solid line) and  $\mathbf{u}(t) = \mathbf{0}$  (dashed). In blue is the trace of the full covariance and in violet the variance of the measurement bias estimation error. The remaining lines are  $\text{tr}(\mathbf{P}_i)$ ,  $i \in \mathcal{M}$ , with the colours corresponding to the landmarks in Figure 10.

vehicle towards the centroid, even when the vehicle is not allowed to stop in its vicinity. Since the orientation is just the angular velocity integrated, Figure 12(b) does not introduce any new information but is included for completeness of results.

An interesting aspect observable on Figure 13 is the apparent continuity of the linear velocity input between optimal segments, which contrasts with what happened in the BVP-based solution. In the angular velocity input, that is no longer the case, but it can be seen that it becomes periodic for the fixed  $v$  case in Figure 14(b).

The mentioned figures allow a better understanding of the behaviour of the system with optimal input. However, the actual effect of the optimal trajectory in the estimation process has not been addressed yet. That is the purpose of Figure 15, which depicts the evolution of the trace of the covariance  $\mathbf{P}(t)$  and its components: the three landmarks and the rate-gyro bias. The blue line is the trace of the full covariance, the green line is the bias covariance, and the

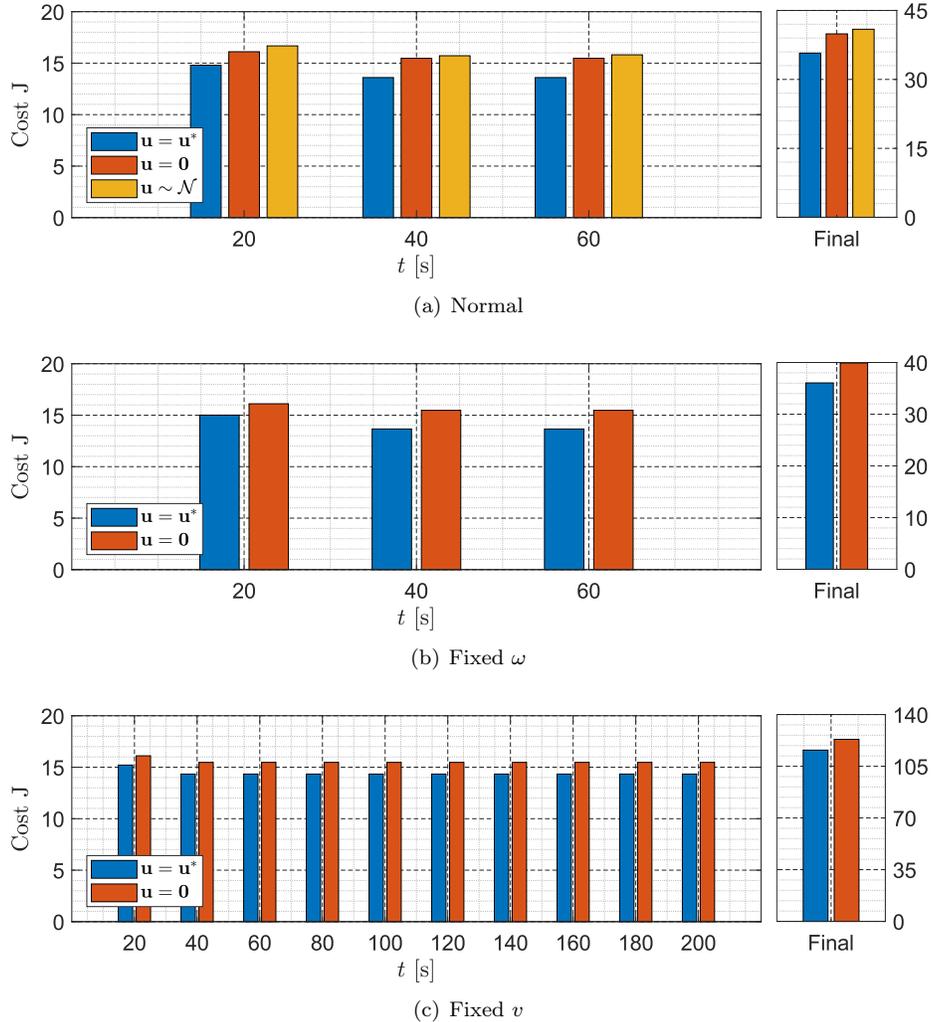


Figure 16: The cost  $J(\mathbf{u})$  computed in each segment for the different scenarios. The final cost is computed as if the whole simulation was a single segment.

remaining colours represent the traces of each individual landmark covariance. To better showcase the improvement brought by the optimal trajectory, the results are compared in each scenario with the covariance evolution with no input (dashed lines). In this figure, it can be seen that there is a clear overall information gain when compared to not moving. However, when looking at individual covariances for landmarks and the bias, it can be observed that only the trace of the covariance for the landmark in front of the vehicle is lower than what would happen if the vehicle did not move, and it is lower than that of the other landmarks as well. Since the covariance weights  $\mathcal{M}$  and  $\mathcal{Q}$  are the same

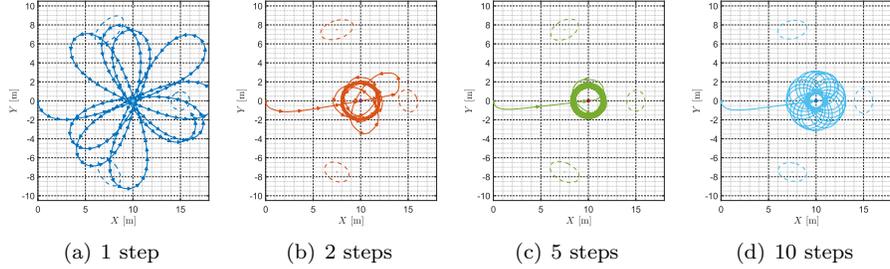


Figure 17: A comparison of the optimized paths for different numbers of optimization steps for the fixed linear velocity case.

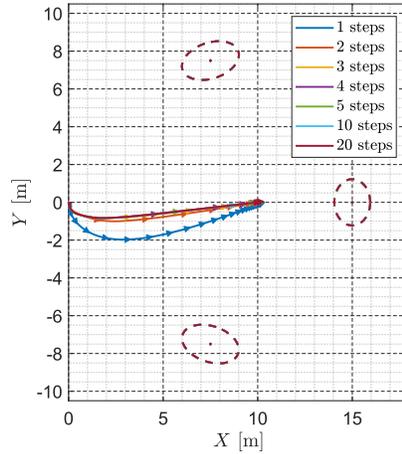


Figure 18: A comparison of the optimized paths for different numbers of optimization steps for the normal case.

for all of them, the control only cares about the complete trace, which is indeed reduced.

Finally, the costs for each of the segments are presented in Figure 16 along with the final cost. Given that for all the cases most of the motion happens in the first segment, that the vehicle is farthest from the place where the potential uncertainty is lowest, and that in the beginning the covariances are set to a higher value, that one naturally represents the highest cost. **The cost is compared to that of not moving, and of a random walk in the area, with inputs computed using  $\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{v}_k$ ,  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \text{diag}((0.3\text{m/s})^2, (0.5^\circ/\text{s})^2))$ .**

To provide a better understanding of what is the impact of increasing the horizon  $N$ , two different cases are studied. Figures 18 and 19 summarize the path, covariance evolution and the cost function for seven different horizons,  $N = 1, 2, 3, 4, 5, 10, 20$ , in the normal situation (optimal linear and angular velocities). A clear conclusion is that increasing the number of steps to be optimized at a time reduces the cost and the trace of the covariance, and makes the

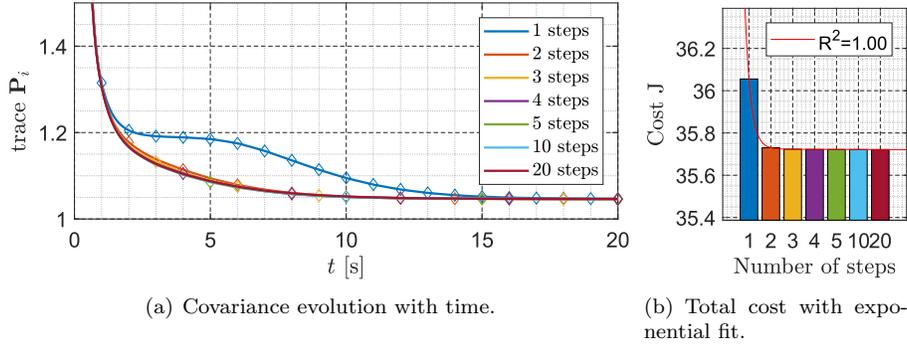


Figure 19: A comparison of the results for different numbers of optimization steps for the normal case.

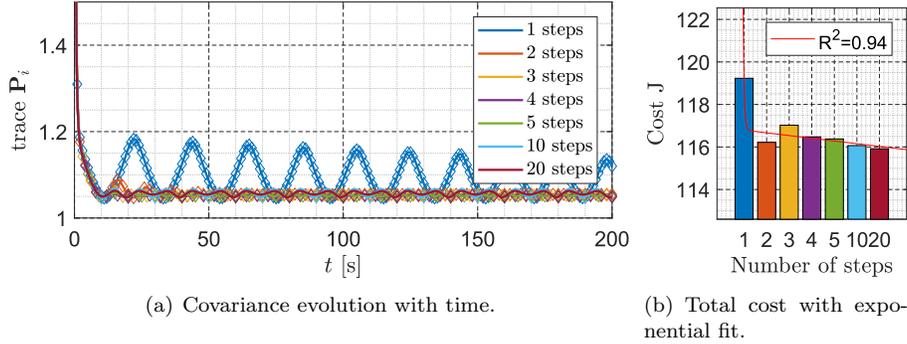


Figure 20: A comparison of the results for different numbers of optimization steps for the fixed linear velocity case.

vehicle turn faster towards the centroid. However, it can be seen that there is a diminishing return with increasing  $N$ . In fact, there is an obvious difference in all the relevant aspects between  $N = 1$  and  $N = 2$ , but when comparing  $N = 2$  to  $N = 20$  that difference is much less evident. This behaviour is manifest in the exponential decay that fits the evolution of the cost with number of steps.

The influence of the horizon in the trajectory is even more relevant in the fixed linear velocity situation, as Figures 17 and 20 show. While all the different optimal inputs take the vehicle to a trajectory around the centroid, the shapes of the actual trajectories are very different depending on the number of steps. For example, when  $N = 1$  the vehicle always moves towards the centroid but is driven away by the constant linear velocity, thus curving again and repeating this behaviour. For  $2 < N \leq 10$  the vehicle always passes once through the centroid, before starting to orbit around it. The shape of this orbit is always different for each case. As seen in Figure 10(c), for  $N = 20$  the vehicle has an eccentric orbit around the centroid but never passes through it. When looking at the evolution of the cost with the number of steps, the exponential trend that

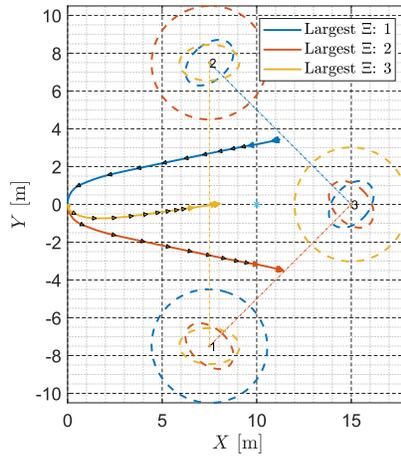


Figure 21: A comparison of the optimized paths obtained when one landmark has a larger process covariance for the normal case. The dashed lines are the lines uniting the remaining landmarks.

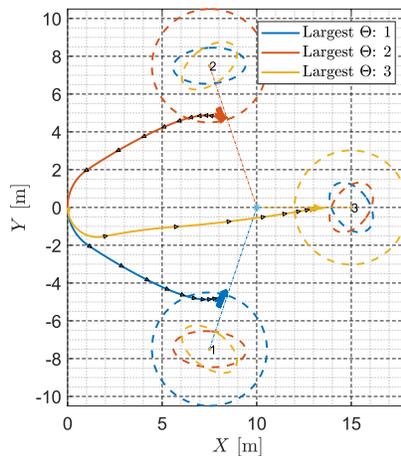


Figure 22: A comparison of the optimized paths obtained when one landmark has a larger measurement covariance for the normal case. The dashed lines are the lines uniting the geometric centroid and the landmark with largest  $\Theta$ .

was observed before is no longer as clear, though there is a much larger decrease in cost from  $N = 1$  to  $N = 2$  than between any other. Curiously, the cost actually increases from  $N = 2$  to  $N = 3$  before decreasing steadily afterwards.

600 In all the simulations shown up to now, the covariances of the Kalman filter were always isotropic, with all landmarks having the same process and measurement covariances. As stated before, it was hinted in previous trials that the geometric centroid of the map is only the endpoint of the optimized path due to the fact that all the landmarks contribute equally to the uncertainty. In

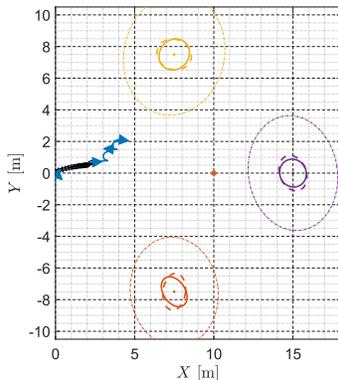


Figure 23: The final scenario when all the covariances are anisotropic and randomly chosen. The blue solid line is the vehicle path, the triangles indicate the vehicle heading, the ellipsoids represent the  $3\sigma$  bounds of the landmarks: the solid line is for  $\mathbf{u}(t) = \mathbf{u}^*(t)$ , the dashed bold line for  $\mathbf{u}(t) = \mathbf{0}$ , and the light is the initial covariance.

605 order to confirm that this is the case, Figures 21 and 22 are provided. In the first, the covariance in  $\Xi$  corresponding to each landmark at a time is set to  $10\mathbf{I}$ , while the remaining are left as originally. In the last, a similar approach is used for the measurement covariance  $\Theta$ . These figures show that the endpoint of the path is indeed dependent on the covariances of the system that underlies the Kalman filter. When a landmark has a much larger process covariance than the others the optimization intrinsically discards that landmark, and the vehicle moves towards what seems to be the centroid of the remaining landmarks. The opposite happens when a landmark has a much larger measurement covariance. In that case, the vehicle tends to a point closer to that landmark, apparently lying in the line defined by the geometric centroid and that landmark.

620 When these situations are combined and taken to the extreme of randomizing  $\mathbf{P}(t_0)$ ,  $\Xi$ , and  $\Theta$ , the results can be both harder to analyse and to solve. Consider Figure 23, where the resulting trajectory is shown along with the initial and final covariances. The initial landmark covariances are now elliptical and correlated between each other. While the map uncertainty is clearly reduced by the optimization procedure (the solid line ellipses are significantly smaller than the dashed line ones), the trajectory is rather odd. Nevertheless, it must be noted that this trajectory is mostly due to the random nature of  $\Xi$  and  $\Theta$ , which, even though they are problem-dependent, can be tuned at will to avoid this kind of trajectories.

630 To close this section, consider a more realistic simulation with 30 visible landmarks. In this final case the optimization process is not initialized with zero error. Instead, the initial state estimate is randomly computed with covariance  $\mathbf{P}(t_0) = \mathbf{I}$ . After each complete optimization with  $N = 20$  steps, the subsequent optimization is performed with the current estimate of the map at that moment. As can be observed in Figure 24, the vehicle maintains the same behaviour as in the toy examples above – it tends to the centroid of the map. Furthermore, as

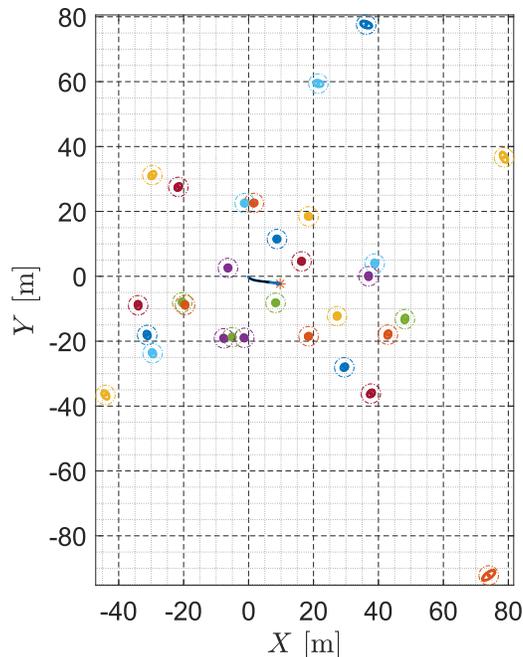


Figure 24: The scenario of the different cases. The blue solid line is the vehicle path, the triangles indicate the vehicle heading, the ellipsoids represent the  $3\sigma$  bounds of the landmarks: the solid line is for  $\mathbf{u}(t) = \mathbf{u}^*(t)$ , the dashed bold line for  $\mathbf{u}(t) = \mathbf{0}$ , and the light is the initial covariance.

seen in Figure 25, the resulting uncertainty is again reduced when compared to not moving (0.34% in trace, 3.4% in determinant) and to a random walk (9% in trace, 99% in determinant), in the two most common criterion,  $A$ -opt and  $D$ -opt. Even though the focus of the research presented in this paper is the uncertainty, Figure 26 shows the estimation error of the complete map (Figure 26(a)), and the details of two variables, one landmark in Figure 26(b) and the rate-gyro bias 26(c). Several observations can be made on these results, most of which are properties explored in experimental results in previous works by the authors [6, 7, 38]. One of the most important is that the filter is consistent, regardless of the choice of input. It can also be concluded that the estimation error is small, since each landmark coordinate has centimetric errors after convergence and the mean error norm for the full map including the transient period is less than 1 m. Furthermore, the lowest mean error norm is achieved with  $\mathbf{u}^*(t)$ . The final observation is the fast convergence of the estimates and their uncertainty, even with non-measured quantities such as the rate-gyro bias.

These simulations aimed at exploring in detail the nature of the optimization problem and its numerical solution. It was shown that, in general, there is a diminishing information gain with increasing the optimization horizon and that,

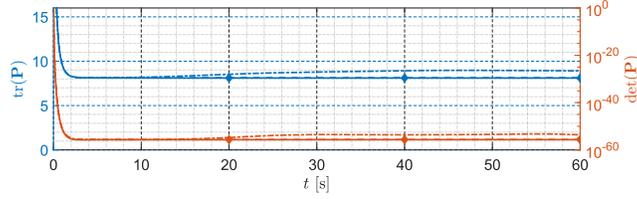


Figure 25: The evolution of the covariance  $\mathbf{P}(t)$  in the filtered scenario, for  $\mathbf{u}^*(t)$  (solid line),  $\mathbf{u}(t) = \mathbf{0}$  (dashed), and  $\mathbf{u}(t) \sim \mathcal{N}$  (dotted). In blue is the trace of the full covariance and in red the determinant of the covariance.

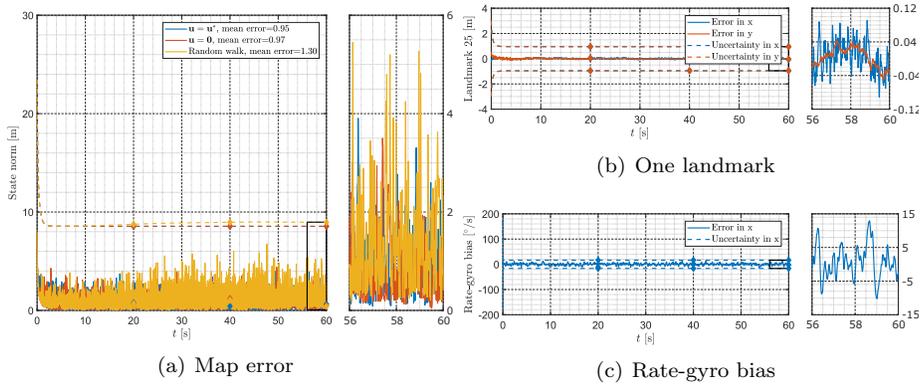


Figure 26: The evolution of the estimation error. On the left, the norm of the estimation error for the whole map, for  $\mathbf{u}^*(t)$  (blue line),  $\mathbf{u}(t) = \mathbf{0}$  (red), and  $\mathbf{u}(t) \sim \mathcal{N}$  (yellow). On the right, the estimation error for  $\mathbf{u}^*(t)$  is shown for two quantities. The dashed lines are the  $3\sigma$  uncertainty for each variable.

for the distances and speeds involved,  $N = 20$  steps is enough to take the vehicle to its optimal destination. Even though the examples shown here are rather simple, they are very informative on the type of trajectories that result from the optimization. Furthermore, it became apparent that the behaviour of the vehicle does not depend on the size of the map, but can vary wildly with the restrictions on the motion model and with the nature of the filter covariances  $\Xi$  and  $\Theta$ .

## 6. Conclusions

In this paper, a novel approach to uncertainty reduction by proper motion planning is presented, grounded in the idea of optimal sensor motion planning in [34] and extended to a class of nonlinear systems where the sensor-based framework of simultaneous localization and mapping fits. Necessary conditions are found using Pontryagin minimum principle for the optimality of the input that minimizes a cost functional that weighs the average covariance in finite horizon and the final covariance. These conditions lead to a two-point boundary

value problem that can be solved numerically for horizons depending on the harshness of the optimization parameters. It is shown numerically that, when there is omnidirectional visibility of the map, this control strategy takes the vehicle to the centroid of the map.

This research was complemented with a direct approach to the optimization problem first solved using the Pontryagin minimum principle. This was achieved by restricting the inputs to piecewise constant functions and using a standard gradient-based optimization algorithm. With this method, it was possible to explore more realistic scenarios such as faster, nonholonomic, systems, and longer (in terms of covered distance) optimization intervals. The numerical results obtained allowed to assess the type of trajectories that result from the optimization process depending on the input parameters, as well as to confirm what was hinted at by the previous BVP-based implementation.

Possible directions of future work are the inclusion of a limited field of view, proving the optimality of the control law by exploring the sufficient conditions associated with the Hamilton-Jacobi equation [36, Section 5], and defining policies for exploration. These are necessary (i) when the field of view of the vehicle does not contain any landmark; and also (ii) to decide when to start exploring.

## Appendix A. Chain rule for scalar functions of matrix functions of vector variables

Let  $\alpha(\mathbf{\Gamma}(\boldsymbol{\beta})) \in \mathbb{R}$ ,  $\mathbf{\Gamma}(\boldsymbol{\beta}) \in \mathbb{R}^{p \times s}$ , and  $\boldsymbol{\beta} \in \mathbb{R}^q$ . The derivative of  $\alpha(\mathbf{\Gamma}(\boldsymbol{\beta}))$  w.r.t. the  $i$ -th element of  $\boldsymbol{\beta}$ , denoted as  $\beta_i$ , is

$$\begin{aligned} \frac{\partial}{\partial \beta_i} \alpha(\mathbf{\Gamma}(\boldsymbol{\beta})) &= \sum_{j=1}^p \sum_{k=1}^s \frac{\partial \alpha(\mathbf{\Gamma}(\boldsymbol{\beta}))}{\partial \gamma_{jk}} \frac{\partial \gamma_{jk}(\boldsymbol{\beta})}{\partial \beta_i} \\ &= \text{tr} \left( \frac{\partial \alpha(\mathbf{\Gamma}(\boldsymbol{\beta}))}{\partial \mathbf{\Gamma}} \frac{\partial \mathbf{\Gamma}(\boldsymbol{\beta})}{\partial \beta_i} \right), \end{aligned}$$

where  $\gamma_{jk}$  is the element  $(j, k)$  of the matrix  $\mathbf{\Gamma}(\boldsymbol{\beta})$ . This can be agglomerated to yield the gradient of  $\alpha$  w.r.t.  $\boldsymbol{\beta}$ ,

$$\nabla_{\boldsymbol{\beta}} \alpha(\mathbf{\Gamma}(\boldsymbol{\beta})) = \begin{bmatrix} \frac{\partial}{\partial \beta_1} \alpha(\mathbf{\Gamma}(\boldsymbol{\beta})) \\ \vdots \\ \frac{\partial}{\partial \beta_q} \alpha(\mathbf{\Gamma}(\boldsymbol{\beta})) \end{bmatrix} = \mathbf{g} \left( \frac{\partial \alpha(\mathbf{\Gamma}(\boldsymbol{\beta}))}{\partial \mathbf{\Gamma}}, \frac{\partial \mathbf{\Gamma}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right) \quad (\text{A.1})$$

where the function  $\mathbf{g}(\mathbf{A}, \mathbf{B}) \in \mathbb{R}^{q \times 1}$  is defined as follows

$$\mathbf{g}(\mathbf{A}, \mathbf{B}) = [\text{tr}(\mathbf{A}\mathbf{B}_1) \quad \cdots \quad \text{tr}(\mathbf{A}\mathbf{B}_q)]^T \quad (\text{A.2})$$

for  $\mathbf{A} \in \mathbb{R}^{s \times p}$  and  $\mathbf{B} = \{\mathbf{B}_1, \dots, \mathbf{B}_q\}$  with  $\mathbf{B}_i \in \mathbb{R}^{p \times s}$ .

### *Acknowledgments*

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) through ISR under LARSyS UID/EEA/50009/2013, and through ID-MEC, under LAETA UID/EMS/50022/2013 contracts. The work of Carlos Silvestre was supported by the University of Macau Projects MYRG2018-00198-FST and MYRG2016-00097-FST, and by the Macao Science and Technology Development Fund under Grant FDCT/026/2017/A1.

### **References**

- [1] H. Durrant-Whyte, T. Bailey, Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms, *IEEE Robotics & Automation Magazine* 13 (2) (2006) 99–110.
- [2] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): Part II, *IEEE Robotics & Automation Magazine* 13 (3) (2006) 108–117.
- [3] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J. M. Rendón-Mancha, Visual simultaneous localization and mapping: a survey, *Artificial Intelligence Review* 43 (1) (2015) 55–81. doi:10.1007/s10462-012-9365-8.
- [4] S. Huang, G. Dissanayake, A critique of current developments in simultaneous localization and mapping, *International Journal of Advanced Robotic Systems* 13 (5) (2016) 1–13. doi:10.1177/1729881416669482.
- [5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, *IEEE Transactions on Robotics* 32 (6) (2016) 1309–1332. doi:10.1109/TR0.2016.2624754.
- [6] B. J. Guerreiro, P. Batista, C. Silvestre, P. Oliveira, Globally Asymptotically Stable Sensor-based Simultaneous Localization and Mapping, *IEEE Transactions on Robotics* 29 (6) (2013) 1380–1395.
- [7] P. Lourenço, B. J. Guerreiro, P. Batista, P. Oliveira, C. Silvestre, Simultaneous Localization and Mapping for Aerial Vehicles: a 3-D sensor-based GAS filter, *Autonomous Robots* 40 (2016) 881–902. doi:10.1007/s10514-015-9499-z.
- [8] P. Lourenço, P. Batista, P. Oliveira, C. Silvestre, C. L. P. Chen, Sensor-based Globally Exponentially Stable Range-Only Simultaneous Localization and Mapping, *Robotics and Autonomous Systems* 68 (2015) 72–85. doi:10.1016/j.robot.2015.01.010.
- [9] P. Lourenço, P. Batista, P. Oliveira, C. Silvestre, A Globally Exponentially Stable filter for Bearing-Only Simultaneous Localization and Mapping with Monocular vision, *Robotics and Autonomous Systems* 100 (2018) 61–77. doi:10.1016/j.robot.2017.11.001.

- [10] J. Castellanos, R. Martinez-Cantin, J. Tardós, J. Neira, Robocentric map joining: Improving the consistency of EKF-SLAM, *Robotics and Autonomous Systems* 55 (1) (2007) 21–29.
- 730 [11] R. Bajcsy, Active perception, *Proceedings of the IEEE* 76 (8) (1988) 966–1005. doi:10.1109/5.5968.
- [12] Y. Aloimonos (Ed.), *Active Perception*, Computer Vision Series, Lawrence Erlbaum Associates, 1993.
- [13] R. Bajcsy, Y. Aloimonos, J. K. Tsotsos, Revisiting active perception, *Autonomous Robots* 42 (2) (2018) 177–196. doi:10.1007/s10514-017-9615-3.
- 735 [14] A. J. Davison, D. W. Murray, Simultaneous localization and map-building using active vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7) (2002) 865–880. doi:10.1109/TPAMI.2002.1017615.
- 740 [15] H. J. S. Feder, J. J. Leonard, C. M. Smith, Adaptive mobile robot navigation and mapping, *The International Journal of Robotics Research* 18 (7) (1999) 650–668. doi:10.1177/02783649922066484.
- [16] A. A. Makarenko, S. B. Williams, F. Bourgault, H. F. Durrant-Whyte, An experiment in integrated exploration, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, 2002, pp. 534–539. doi:10.1109/IRDS.2002.1041445.
- 745 [17] C. Leung, S. Huang, G. Dissanayake, Active SLAM using Model Predictive Control and Attractor based Exploration, in: *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5026–5031. doi:10.1109/IR0S.2006.282530.
- 750 [18] S. B. Thrun, K. Möller, Active exploration in dynamic environments, in: J. E. Moody, S. J. Hanson, R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems* 4, Morgan-Kaufmann, 1992, pp. 531–538.
- [19] Y.-H. Choi, T.-K. Lee, S.-Y. Oh, A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot, *Autonomous Robots* 24 (1) (2008) 13–27. doi:10.1007/s10514-007-9050-y.
- 755 [20] I. Maurović, M. Seder, K. Lenac, I. Petrović, Path Planning for Active SLAM Based on the D\* Algorithm With Negative Edge Weights, *IEEE Transactions on Systems, Man, and Cybernetics: Systems PP* (99) (2017) 1–11. doi:10.1109/TSMC.2017.2668603.
- 760 [21] C. Stachniss, D. Hahnel, W. Burgard, Exploration with active loop-closing for fastslam, in: *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol. 2, 2004, pp. 1505–1510. doi:10.1109/IR0S.2004.1389609.
- 765

- [22] H. Carrillo, I. Reid, J. Castellanos, On the comparison of uncertainty criteria for active SLAM, in: Proc. of the 2012 IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 2080–2087. doi:10.1109/ICRA.2012.6224890.
- 770 [23] J. Tardós, J. Neira, P. Newman, J. Leonard, Robust mapping and localization in indoor environments using SONAR data, The International Journal of Robotics Research 21 (4) (2002) 311–330.
- [24] H. Carrillo, Y. Latif, M. L. Rodríguez-Arevalo, J. Neira, J. A. Castellanos, On the monotonicity of optimality criteria during exploration in active SLAM, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1476–1483. doi:10.1109/ICRA.2015.7139384.
- 775 [25] M. L. Rodríguez-Arévalo, J. Neira, J. A. Castellanos, On the Importance of Uncertainty Representation in Active SLAM, IEEE Transactions on Robotics 34 (3) (2018) 829–834. doi:10.1109/TR0.2018.2808902.
- 780 [26] R. Sim, N. Roy, Global A-Optimal Robot Exploration in SLAM, in: Proc. of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 661–666. doi:10.1109/ROBOT.2005.1570193.
- [27] L. Carlone, J. Du, M. Kaouk Ng, B. Bona, M. Indri, Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence, Journal of Intelligent & Robotic Systems 75 (2) (2014) 291–311. doi:10.1007/s10846-013-9981-9.
- 785 [28] C. Leung, S. Huang, G. Dissanayake, Active SLAM in structured environments, in: 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 1898–1903. doi:10.1109/ROBOT.2008.4543484.
- 790 [29] Y. Chen, S. Huang, R. Fitch, J. Yu, Efficient Active SLAM Based on Submap Joining, Graph Topology and Convex Optimization, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 1–8. doi:10.1109/ICRA.2018.8460864.
- [30] M. Kontitsis, E. Theodorou, E. Todorov, Multi-robot active SLAM with relative entropy optimization, in: Proc. of the 2013 American Control Conference, 2013, pp. 2757–2764. doi:10.1109/ACC.2013.6580252.
- 795 [31] S.-Y. An, L.-K. Lee, S.-Y. Oh, Ceiling vision-based active slam framework for dynamic and wide-open environments, Autonomous Robots 40 (2) (2016) 291–324. doi:10.1007/s10514-015-9453-0.
- 800 [32] S. M. Chaves, A. Kim, E. Galceran, R. M. Eustice, Opportunistic sampling-based active visual slam for underwater inspection, Autonomous Robots 40 (7) (2016) 1245–1265. doi:10.1007/s10514-016-9597-6.

- [33] A. Klesh, P. Kabamba, A. Girard, Optimal path planning for uncertain exploration, in: Proc. of the 2009 American Control Conference, 2009, pp. 2421–2426. doi:10.1109/ACC.2009.5160286.
- [34] I. I. Hussein, Kalman filtering with optimal sensor motion planning, in: Proc. of the 2008 American Control Conference, Seattle, Washington, USA, 2008, pp. 3548–3553. doi:10.1109/ACC.2008.4587043.
- [35] M. Athans, E. Tse, A direct derivation of the optimal linear filter using the maximum principle, IEEE Transactions on Automatic Control 12 (6) (1967) 690–698. doi:10.1109/TAC.1967.1098732.
- [36] M. Athans, P. Falb, Optimal Control: An Introduction to the Theory and Its Applications, Dover Books on Engineering Series, Dover Publications, 2006.
- [37] P. Lourenço, P. Batista, P. Oliveira, C. Silvestre, Towards Uncertainty Optimization in Active SLAM, in: Proc. of the 54th IEEE Conference on Decision and Control, Osaka, Japan, 2015, pp. 3242–3247.
- [38] P. Lourenço, B. J. Guerreiro, P. Batista, P. Oliveira, C. Silvestre, New Design Techniques for Globally Convergent Simultaneous Localization and Mapping: Analysis and Implementation, in: T. I. Fossen, K. Y. Pettersen, H. Nijmeijer (Eds.), Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles, Lecture Notes in Control and Information Sciences, Springer, 2017. doi:10.1007/978-3-319-55372-6\_6.
- [39] P. Batista, C. Silvestre, P. Oliveira, Single range aided navigation and source localization: Observability and filter design, Systems & Control Letters 60 (8) (2011) 665–673.
- [40] P. Batista, C. Silvestre, P. Oliveira, A GES Attitude Observer with Single Vector Observations, Automatica 48 (2) (2012) 388–395.
- [41] P. Batista, C. Silvestre, P. Oliveira, Globally exponentially stable filters for source localization and navigation aided by direction measurements, Systems & Control Letters 62 (11) (2013) 1065–1072.
- [42] J. Nocedal, S. Wright, Numerical Optimization, 2nd Edition, Springer Series in Operations Research and Financial Engineering, Springer, 2006.
- [43] B. D. O. Anderson, Stability properties of Kalman-Bucy filters, Journal of the Franklin Institute 291 (2) (1971) 137–144.
- [44] D. Viegas, P. Batista, P. Oliveira, C. Silvestre, Nonlinear observability and observer design through state augmentation, in: Proc. of the 53rd IEEE Conference on Decision and Control, Los Angeles, USA, 2014, pp. 133–138.
- [45] W. Vetter, Derivative operations on matrices, IEEE Transactions on Automatic Control 15 (2) (1970) 241–244. doi:10.1109/TAC.1970.1099409.

- [46] A. Gelb, Applied Optimal Estimation, MIT Press, 1974.
- [47] H. B. Keller, Numerical Solution of Two Point Boundary Value Problems, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1976.
- 845 [48] L. Shampine, M. Reichelt, J. Kierzenka, Solving boundary value problems for ordinary differential equations in matlab with bvp4c (October 2000). URL [http://www.mathworks.com/bvp\\_tutorial](http://www.mathworks.com/bvp_tutorial)
- [49] H. G. Buck, K. J. Plitt, A multiple shooting algorithm for direct solution of optimal control problems, in: Proc. of the 9th IFAC World Congress, 850 Budapest, Hungary, 1984, pp. 242–247.
- [50] R. Holsapple, R. Venkataraman, D. Doman, A modified simple shooting method for solving two-point boundary-value problems, in: Proc. of the 2003 IEEE Aerospace Conference, Vol. 6, 2003, pp. 2783–2790. doi:10.1109/AERO.2003.1235204.
- 855 [51] A. Trent, R. Venkataraman, D. Doman, Trajectory generation using a modified simple shooting method, in: Proc. of the 2004 IEEE Aerospace Conference, Vol. 4, 2004, pp. 2723–2729. doi:10.1109/AERO.2004.1368069.
- [52] J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence properties of the nelder–mead simplex method in low dimensions, SIAM Journal on Optimization 9 (1) (1998) 112–147. doi:10.1137/S1052623496303470. 860
- [53] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. general considerations, IMA Journal of Applied Mathematics 6 (1) (1970) 76–90. arXiv:/oup/backfile/content\_public/journal/imamat/6/1/10.1093/imamat/6.1.76/2/6-1-76.pdf, doi:10.1093/imamat/6.1.76. 865
- [54] R. Fletcher, A new approach to variable metric algorithms, The Computer Journal 13 (3) (1970) 317–322. arXiv:/oup/backfile/content\_public/journal/comjnl/13/3/10.1093/comjnl/13.3.317/2/130317.pdf, doi:10.1093/comjnl/13.3.317.
- 870 [55] D. Goldfarb, A family of variable-metric methods derived by variational means, Mathematics of Computation 24 (109) (1970) 23–26. doi:10.2307/2004873.
- [56] D. F. Shanno, Conditioning of quasi-newton methods for function minimization, Mathematics of Computation 24 (111) (1970) 647–656.