Visual Navigation in Unknown Environments

Teresa A. Vidal Calleja

Institut de Robòtica i Informàtica Industrial

Universitat Politècnica de Catalunya

Consejo Superior de Investigaciones Científicas

A thesis co-directed by:

Juan Andrade Cetto and Alberto Sanfeliu

A thesis submitted for the degree of

Doctor of Philosophy

Barcelona, 2007

Universitat Politècnica de Catalunya

Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial

PhD program:
Control, Visió i Robòtica

This thesis was completed at:

Institut de Robòtica i Informàtica Industrial, CSIC-UPC

Thesis advisors:
Juan Andrade Cetto
Alberto Sanfeliu

Abstract

Navigation in mobile robotics involves two tasks, keeping track of the robot's position and moving according to a control strategy. In addition, when no prior knowledge of the environment is available, the problem is even more difficult, as the robot has to build a map of its surroundings as it moves. These three problems ought to be solved in conjunction since they depend on each other.

This thesis is about simultaneously controlling an autonomous vehicle, estimating its location and building the map of the environment. The main objective is to analyse the problem from a control theoretical perspective based on the EKF-SLAM implementation. The contribution of this thesis is the analysis of system's properties such as observability, controllability and stability, which allow us to propose an appropriate navigation scheme that produces well-behaved estimators, controllers, and consequently, the system as a whole.

We present a steady state analysis of the SLAM problem, identifying the conditions that lead to partial observability. It is shown that the effects of partial observability appear even in the linear Gaussian case. This indicates that linearisation alone is not the only cause of SLAM inconsistency, and that observability must be achieved as a prerequisite to tackling the effects of linearisation. Additionally, full observability is also shown to be necessary during diagonalisation of the covariance matrix, an approach often used to reduce the computational complexity of the SLAM algorithm, and which leads to full controllability as we show in this work.

Focusing specifically on the case of a system with a single monocular camera, we present an observability analysis using the nullspace basis of the stripped observability matrix. The aim is to get a better understanding of the well known intuitive behaviour of this type of systems, such as the need for triangulation to features from different positions in order to get accurate relative pose estimates between vehicle and camera. Through characterisation the unobservable directions in monocular SLAM, we are able to identify the vehicle motions required to maximise the number of observable states in the system.

When closing the control loop of the SLAM system, both the feedback controller and the estimator are shown to be asymptotically stable. Furthermore, we show that the tracking error does not influence the estimation performance of a fully observable

system and viceversa, that control is not affected by the estimation. Because of this, a higher level motion strategy is required in order to enhance estimation, specially needed while performing SLAM with a single camera. Considering a real-time application, we propose a control strategy to optimise both the localisation of the vehicle and the feature map by computing the most appropriate control actions or movements. The actions are chosen in order to maximise an information theoretic metric. Simulations and real-time experiments are performed to demonstrate the feasibility of the proposed control strategy.

Resum

La navegació en robòtica mòbil implica dues tasques: el coneixement de la posició del robot i el moviment segons una estratègia de control. A més a més, quan no es té coneixement previ de l'entorn, el problema és fins i tot més difícil ja que el robot ha de construir un mapa dels seus voltants mentre es mou. Aquests tres problemes s'haurien de resoldre conjuntament ja que depenen l'un de l'altre.

Aquesta tesi tracta de com, simultàniament, controlar un vehicle autònom mentre es calcula la seva localització i construeix el mapa de l'entorn. L'objectiu principal és analitzar el problema des d'un punt de vista de teoria de control basada en l'implementació d'EKF-SLAM. La contribució d'aquesta tesi és l'anàlisi de les propietats del sistema, tals com l'observabilitat, la controlabilitat i l'estabilitat, que permeten proposar un esquema de navegació apropiat que produeix estimadors ben condicionats, controladors, i conseqüentment, el sistema globalment.

Es presenta una anàlisi de règim permanent del problema d'SLAM, identificant les condicions que porten a l'observabilitat parcial. Es mostra que els efectes d'observabilitat parcial apareixen fins i tot en el cas ideal gaussià i lineal. Això indica que la linealització només no és l'única causa d'inconsistència de l'SLAM, i que l'observabilitat s'ha d'aconseguir com a pre-requisit per a tractar els efectes de la linealització. Addicionalment, es mostra que l'observabilitat completa és necessaria durant la diagonalització de la matriu de covariancia, un enfoc que sovint s'utilitza per reduir la complexitat computacional de l'algorisme d'SLAM, i que condueix a la controllabilitat plena com es mostra en aquest treball.

Considerant particularment el cas d'un sistema amb una única càmera, es presenta un anàlisi d'observabilitat que utilitza la base de l'espai nul de la matriu d'observabilitat. El propòsit és aconseguir una millor comprensió del comportament intuïtiu, ben conegut, d'aquest tipus de sistemes, com ara la necessitat de triangular característiques des de posicions diferents per aconseguir estimacions acurades de la posa relativa entre el vehicle i la càmera. Durant la caracterització de les direccions no observables en l'SLAM monocular, es poden identificar els moviments que cal del vehicle per tal de maximitzar el nombre d'estats observables en el sistema.

En tancar el bucle de control del sistema d'SLAM, es mostra que tant el controlador com l'estimador són asimptòticament estables. A més, es mostra que l'error en el

seguiment no influeix en el rendiment d'estimació d'un sistema plenament observable, i viceversa, que l'estimació no afecta el rendiment del control. A causa d'això, per millorar l'estimació cal una estratègia de moviment de més alt nivel, especialment en el cas d'SLAM amb una única càmera. Considerant una aplicació en temps real, es proposa una estratègia de control per optimitzar tant la localització del vehicle com el mapa de característiques computant les accions de control o moviments més apropiats. Les accions s'escullen per maximitzar una mètrica basad en teoria de l'informació. Es realitzen simulacions i experiments en temps real per demostrar la viabilitat de l'estratègia de control proposada.

Resumen

La navegación en robótica móvil requiere de dos tareas principalmente: el conocimiento de la posición del robot y el movimiento según una estrategia de control. Además, cuando no se conoce previamente el entorno, el problema se vuelve más difícil, ya que el robot debe construir un mapa de su alrededor mientras se mueve. Estos tres problemas deben ser tratados en conjunto ya que dependen unos de los otros.

Esta tesis trata de resolver simultáneamente, el control de un vehículo autónomo mientras se calcula su localización y se construye un mapa del entorno. El objetivo principal es analizar el problema desde un punto de vista de teoría de control basada en la implementación del EKF-SLAM. La contribución de esta tesis es el análisis de las propiedades del sistema, tales como la observabilidad, la controlabilidad y la estabilidad, que permiten proponer un esquema de navegación apropiado, produciendo así estimadores, controladores y como consecuencia sistemas bien condicionados.

En este trabajo se presenta un análisis en estado estacionario del problema de SLAM, identificando las condiciones que producen observabilidad parcial. Se muestra que los efectos de la observabilidad parcial aparecen incluso en el caso ideal lineal y gaussiano. Esto indica que la linealización no es la única causa de la conocida inconsistencia del SLAM y que la observabilidad se debe garantizar como pre-requisito para tratar los efectos de la linealización. Adicionalmente, se muestra que la observabilidad total es necesaria durante la diagonalización de la matriz de covarianza, un enfoque que comúnmente utilizado para reducir la complejidad computacional del algoritmo de SLAM, que a su vez produce controlabilidad total como aquí se muestra.

Considerando particularmente el caso de un sistema con una sola cámara, se presenta un análisis de observabilidad que utiliza las bases del espacio nulo de la matriz de observabilidad. El propósito es comprender mejor los comportamientos intuitivos, bien conocidos, de este tipo de sistemas, como la necesidad de triangular características desde posiciones diferentes para conseguir estimaciones acotadas de las posiciones relativas entre el vehículo y la cámara. Caracterizando las direcciones no observables del SLAM monocular, se pueden identificar los movimientos que requiere el vehículo para maximizar el número de estados observables del sistema.

Al cerrar el lazo de control del sistema de SLAM, se muestra que tanto el controlador como el estimador son asintóticamente estables. Además, se muestra que el error

de seguimiento de la trayectoria deseada no influye en el comportamiento del estimador de un sistema completamente observable, y viceversa, que la estimación no afecta el comportamiento del controlador. Debido a esto, para mejorar la estimación se requiere una estrategia de movimiento de más alto nivel, especialmente cuando el único sensor es una cámara. Se considera una aplicación en tiempo real y se propone una estrategia de control para optimizar tanto la localización del vehículo como el mapa de características visuales, calculando las acciones más apropiadas. Estas acciones se escogen al maximizar una métrica basada en teoría de la información. Se realizan simulaciones y experimentos en tiempo real para demostrar la viabilidad de la estrategia propuesta.

Acknowledgments

First and foremost I would like to thank both Juan Andrade-Cetto and Alberto Sanfeliu for being such a good team as supervisors, for the excellent discussions and their invaluable support over all these years. To Andrew Davison and David Murray for giving me an outstanding overview of the applied theory in the Active Vision Lab of the University of Oxford. To Salah Sukkarieh for hosted me in ACFR at Sydney University.

This work was funded by the Spanish Ministry of Education and Science under the CICYT project DPI2001-2223.

Many thanks to the entire people of the IRI for your support and company. To Carme, Fede and Jordi for being such a good head. To Juan again, More, Guillem, Porta, Alejandro, Jose Luis, Mirats, Sergi and Viorela for our endless lunches, good moments and nice discussions. To Porta for your comments to this work.

I would like to thank to my reviewers David Murray and Jose Santos-Victor for taking the time of reading this thesis and their contributions to the manuscript.

Thanks also to Anna, with who I shared a really nice experience, together with Fred, Teo, Dave, Manjari, Kim, Lisa and Monti for making my life easier in Oxford. To the people of ACFR: Bart, Roman, Mitch, Sharon, Marie, Eva, Tobie, Moriama and Susana for the good trips and parties in Sydney. Specially to Alex Mak for making of my stay a great experience, without forgetting your suggestions to make this work more readable.

I am grateful to Teresa, Jacinto, Cacho e Isela for their unconditional support. To my friends Paz, Claire, Nancy, Alberto, Jorge, Jordi, Brau and the Tijuana crew for the good moments in Barcelona. To all my friends from Mexico who I always feel close. And last but not least to Iñaki for sharing this amazing experience with me.

Contents

List of Figures

List of Examples

Nomenclature

Notation

$\mathbf{x}$	SLAM state vector
$\mathbf{X}$	set of state vectors
$\mathbf{x}_r$	vehicle state
$\mathbf{x}_f$	feature state
$\mathbf{e}$	state estimation error
$\mathbf{P}$	total covariance matrix
$\mathbf{P}_r$	vehicle covariance matrix
$\mathbf{P}_f$	feature covariance matrix
$\mathbf{P}_{rf}$	cross covariance between vehicle and landmark
$\mathbf{u}$	control input
$\mathbf{v}$	process noise
$\mathbf{Q}$	process noise covariance matrix
$\mathbf{z}$	measurement vector
$\mathbf{w}$	measurement noise
$\mathbf{R}$	measurement noise covariance matrix
$\mathbf{Z}$	set of measurement vectors
$\mathbf{f}(\cdot)$	process function
$\mathbf{h}(\cdot)$	measurement function
$\mathbf{F}$	process Jacobian w.r.t the state
$\mathbf{G}$	process Jacobian w.r.t the input
$\mathbf{H}$	measurement Jacobian w.r.t the state
$\mathbf{S}$	innovation covariance matrix
$\mathbf{K}$	Kalman gain
$\mathcal{O}$	observability Gramian
$\mathcal{Q}$	observability matrix
$\mathcal{C}$	controllability matrix
$\mathcal{Q}_{\texttt{TOM}}$	total observability matrix
$\mathcal{Q}_{\texttt{SOM}}$	stripped observability matrix

$\mathcal{N}(\cdot)$	null space
$\mathbf{C}$	orientation matrix
$\mathbf{E}$	orientation rate matrix
$\mathbf{p}$	position vector
$\mathbf{\Psi}$	orientation expressed as Euler angles
$[\phi, \theta, \psi]^\top$	Euler angles
$\boldsymbol{\upsilon}$	linear velocity
$\boldsymbol{\omega}$	angular velocity
$\mathbf{f}$	linear acceleration
$\boldsymbol{\alpha}$	angular acceleration
$\mathbf{f}^s$	linear acceleration in sensor frame
$\mathbf{r}$	relative position between a feature and the sensor
$\mathbf{r}^s$	relative position between a feature and the sensor in sensor frame
$\mathbf{q}$	orientation expressed in quaternion form
$\boldsymbol{\mathcal{P}}$	quaternion transition matrix
$\mathbf{\Omega}$	angular velocity expressed in quaternion form
φ, ϑ	azimuth and elevation angles
$[u, v]$	point image projection
d	radial distortion term
f	focal legth
$[u_0, v_0]$	camera principal point
k_u, k_v	pixel densities
$\mathcal{J}$	control performance index or cost fuction
$\mathbf{Q}_i$	control gain
$\mathbf{L}$	LQG control gain
$\mathbf{e}_c$	tracking error
$H(X)$	entropy
$H(X \mid Z)$	conditional entropy
$I(X; Z)$	mutual information
$\boldsymbol{\mathcal{F}}$	Fisher information matrix
$\mathbf{x}_k$	state in discrete time t_k
$\mathbf{x}_{k+1\|k}$	apriori state estimate in t_{k+1}
$\mathbf{x}_{k+1\|k+1}$	aposteriori state estimate in t_{k+1}

Acronyms

DOF	degrees of freedom
EIF	extended information filter
EKF	extended Kalman filter
FI	Fisher information
FL	feedback linearitsation
GUI	graphical user interface

KF	Kalman filter
LQG	linear quadratic Gaussian
MI	mutual information
pdf	probability density function
psd	positive semi-definite
SLAM	simultaneous localisation and mapping
SOM	stripped observability matrix
TOM	total observability matrix
UKF	unscented Kalman filter
UT	unscented transformation
VS	visual servoing
2D	two-dimensional
3D	three-dimensional

Chapter 1

Introduction

This thesis is about simultaneously controlling an autonomous vehicle, estimating its location and building a feature map. Navigation in unknown environments for autonomous vehicles using a reduced number of sensors requires the vehicle to perform three main tasks: construct its own representation of the environment, localise on it, and compute its own control to reach some destination or goal.

The emphasis is placed on understanding the theoretical limitations of the estimation process and on the relationship between estimation and control. The analysis of the system properties observability, stability and contollability allows us to understand the system. Moreover, it allow us to propose an appropriate navigation scheme to achieve desired properties of the estimator, the controller and consequently the system.

Of special interest is the case of navigation using a single monocular camera because only orientation information can be measured from it. Some of our analysis though will consider other sensors which are able to provide information about depth and orientation.

In this chapter we present the motivation, objectives and contribution of this work.

1.1 Motivation

Many mobile robotic applications need to navigate in unknown environments. Some examples are: the BR 700 industrial robots for transportations, the HELPMATE of Pyxis Corp. also used for transportation task particularly in hospitals, the iRobot Corp. ROOMBA vacuum cleaner, the TOURBOT used as a tour-guide robot in museums or EXPOS, the SOJOUNER used during Pathfinder missions to explore Mars, other robots designed to move wood out of forests, for agricultural applications, airduct inspection, the K-Team's KEPHERA robot, the ActivMedia Robotics PIONEER, and the iRobot Corp. B21 all for education and research, etc. Despite the broad set of applications, all have something in common: the integration of many different fields, such as dynamics and control to solve locomotion problems, signal analysis and computer vision to create robust perceptual systems, computer algorithms, information theory, artificial intelligence, and probability theory for localisation and navigation tasks.

Solving the navigation problem involves answering questions like *where am I?*, *what the world looks like?* and *how to get there?*. The well-known approach that answers the first two questions is SLAM: Simultaneous Localisation and Mapping. SLAM solves the problem of vehicle localisation and environment mapping based on perception information, and it does that by filtering, i.e. by predicting what is going to happen based on stochastic models and correcting with real measurements. There is still one question to answer; *how to get there?*, based on the information provided by the SLAM algorithm. This is what this thesis is about.

Autonomous vehicles must be able to plan their own motion, i.e. to decide automatically their control commands to achieve a specified task. Commonly, it is assumed that the vehicle has complete and exact knowledge of its environment, of course this assumption is not always realistic. If uncertainty in the prior knowledge is small, it is reasonable to anticipate all possible contingencies and to generate sensor-based motion plans that can deal with them. Sensing is used to guide the motions and monitor their execution. On the other hand, if the autonomous vehicle has no prior knowledge about its environment, it is necessary first to learn about it.

Maps are commonly used for mobile robot navigation [12]. To acquire a map, vehicles must carry sensors that enable them to perceive the outside world. However,

almost all sensor measurements are subject to errors, often referred to as measurement noise. More importantly, most vehicle sensors are subject to range limitations. These range limitations make it necessary for a vehicle to navigate through its environment while, at the same time, building a map of it. The motion commands or controls issued during navigation carry important information for building maps, since they are used to get information about the locations at which different sensor measurements were taken. Vehicle motion is also subject to errors, and the controls alone are therefore insufficient to accurately determine a vehicle's pose (location and orientation) relative to its environment.

There are many sources of uncertainty in mobile robotics, in essence, both the vehicle localisation and the map are uncertain, and by focusing just on one or the other introduces systematic noise. Thus, estimating both unknowns at the same time has the pleasing property that both the measurement and the control noise are independent with regards to the properties that are being estimated. Mobile robots operate in environments that are either partially or completely unknown. Often the environment is changing with time in an unknown manner; hence, sensors that can enable the vehicle to navigate in these environments are well motivated.

Sensors that are commonly used in navigation include cameras, range finders using sonar, laser, and infrared technology, radar, tactile sensors, compasses, and GPS. Initially sonars were used as an *intelligent* sensor. However, based on the fact that sonars have very low spatial bandwith capabilities and are subject to noise due to wave scattering, the use of laser range sensing has increased. Although laser sensors have a much higher bandwidth, they are still subject to noise. Moreover, lasers have a limited field-of-view, unless intricate mechanics such as rotating mirrors are incorporated in the sensor design. Given the shortcomings of laser and sonar-based navigation, researchers are now interested on the use of camera-based systems (vision sensors).

Vision sensors can have wide field-of-view, can have millisecond sampling rates, and thus can be easily used for control. However, some disadvantages of vision include lack of depth information, image occlusion, low resolution and the requirement for extensive data interpretation (recognition). Cameras can be cheaper in comparison to other sensor such as laser range scanners. Even with its own advantages and disavantages, the choice of monocular cameras as a sensor for navigation results in a competitive

selection. Visual feedback loops can be used to perform different navigation tasks, for example to correct the position of a robot, to detemine the pose of the robot relative to a target or to follow a predefined path. Several schemes have been proposed to visually control a robot. Visual servoing is a common approach used for this purpose, which has seldom been treated in a localisation and mapping context. Even though in this thesis we are not strictly applying visual servoing techniques, our active control scheme has several similarities to this visual feedback control.

Autonomous navigation requires real-time performance as well. Using modest hardware imposes severe restrictions on the volume of computation that can take place in a time step. In order to get a fully autonomous vehicle working in a partially unknown environment, image processing, control decision making, full estimation of vehicle location and map updating should be done at video-rates, 16 or 33 [ms].

Figure 1.1 illustrates the challenges facing a robot solving efficiently the problem of autonomous navigation in partially unknown environments. Three main tasks are identified and the overlapping areas of this control, localisation and mapping tasks show the combined problems to deal with. As a summary, the control task must deal with the kinematics constraints of the vehicle, usually nonholonomic, with nonlinear models and take into account the well-known dead reckoning typical of mobile robots. The control law must be computed in real-time to get autonomy of the vehicle and it must consider that the robot is navigating in a dynamic environment as well. The localisation task must concern about noisy sensors, nonlinear models, dead reckoning and that the environment could be partially of fully unknown. The mapping task mostly has to deal with data association, noisy sensors, dynamic environments and the amount of features that the system can work with in a real-time application.

This work intends to cover most of these problems in order to navigate with vision in partially unknown environments. In this thesis we mainly focus on wheeled mobile robot such as the ActivMedia's PIONEER to perform the analysis, simulations and experiments, eventhough we analyse as well different type of autonomous systems, like a single hand-held camera that considers a 6-DOF constant velocity model.

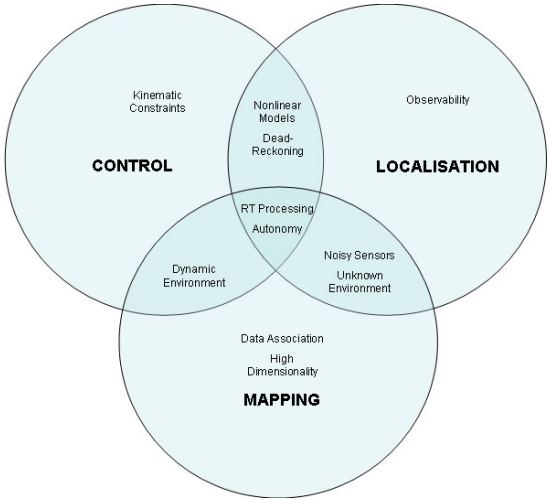


Figure 1.1: Challenges that need to be solved by an autonomous vehicle while navigating. The overlapping areas represent combinations of the mapping, localisation, and control problems.

1.2 Objectives

A mobile platform should be able to *navigate autonomously* in its environment; to do that it has to construct a map of the environment, in order to self-localise and to decide where to go.

A study, including description, analysis and implementation, of an *integrated system* that considers control, perception, and learning for mobile robots in urban, laboratory or compatible industrial surroundings, is the primary goal of this thesis.

The subobjectives are identified as follows:

- To study and analyse in depth the simultaneous localisation and map building algorithms, in order to enrich the navigation and also to guarantee a stable estimation of the vehicle location and the model of the environment.
- Monocular vision is one case in which a single measurement step can only provide incomplete information for the reconstruction of the state space, with the consequence that feature locations cannot be estimated from a single image, and must be computed from the tracking of features over multiple views. To study it in depth, using the tools from control theory such as an *observability analysis*.

- One of the open challenges in the simultaneous localisation and map building problem is *to close the control loop* for the estimated system. To study all the implications of closing the control loop, and the entire system as a whole, including the estimator and the controller, focusing both on theoretical and practical issues.
- The problem of autonomous navigation is handled using mainly artificial vision. To design a *vision control* while simultaneously performing localisation and mapping in real-time.
- One particular challenge of navigation with mobile robots is to localise it in 3D non-flat terrains. It is necessary to provide good models of the real robot, including the mobile platform and the different sensors used for localisation and for navigation control.
- To evaluate of the resulting algorithms both in simulations and in *real-time experiments* with one of our wheel mobile robots or a hand-held camera.

1.3 Contributions

The main contributions of this dissertation, making reference to respective publication, are summarised as follows:

- An analysis of the state estimation error dynamics for a linear system within the Kalman filter based approach to SLAM. We demonstrate that such dynamics is marginally stable and the necessary modifications required in the observation model, in order to guarantee zero mean stable error dynamics [138] [5].
- We provide tight constraints in the amount of decorrelation possible, during the covariance inflation methods to speed up the Kalman filter based SLAM algorithm. The idea is to guarantee convergence of the steady-state error covariance to obtain a suboptimal filter that is both linear in time, and stable, at the same time [137]. Moreover, we show that the decorrelation method is closely related with full controllability condition.
- An algorithm to reduce the affects caused by linearisation in the typical EKF approach to SLAM [6].

- An observability analysis for different bearing-only SLAM systems. The characterisation of the unobservable directions is made using the nullspace basis of the stripped observability matrix. This allow us to identify which vehicle motions are required to maximise the number of observable states in the system, which in turn affects the accuracy in the estimation process. The analysis is performed by modelling the system in the continuous time domain as piecewise constant [139]. In addition nonlinear observability analysis for a vehicle moving in 2D is also performed.

- A single and multi-vehicle control scheme that uses the state estimates generated from the output of the SLAM algorithm as input to a vehicle controller. Given the separability between optimal state estimation and regulation, we show that the tracking error does not influence the estimation performance of a fully observable EKF based SLAM implementation, and vice versa, that estimation errors do not undermine controller performance [4].

- A simulated environment in Simulink/Matlab in order to test the proposed control strategies. This toolbox consists of different blocks that represent motion models such as: planar differential steer vehicle model, 3D differential steer vehicle model, single camera constant velocity model. Besides blocks that represent the measurement models such as: laser-range scanner, full perspective projection camera model, wide-angle camera model, stereo system, and blocks for filtering such as the EKF (see Appendix B).

- An active control strategy to optimise both the localisation of the sensor and building of the feature map by computing the most appropriate control actions or movements. Considerating a single hand-held camera performing SLAM at real-time with generic six degrees of freedom (6-DOF) motion. The actions belong to a discrete set (e.g. go forward, go left, go up, turn right, etc), and are chosen so as to maximise the mutual information gain between posterior states and measurements. Given that our system is capable of producing motion commands for a real-time 6DOF visual SLAM [140].

- A control Class in C++ for real-time monocular SLAM (see Appendix C).

- An active motion control strategy and a motion model for a mobile robot navigating in uneven terrains. Using measurements from only one camera the system autonomously builds a visual feature map while at the same time optimises its localisation within this map. The technique chooses the most appropriate commands maximising the expected information gain between prior states and measurements, while performing 6-DOF bearing-only SLAM at real-time. The proposed motion model is used to infer the position of the vehicle in a non-flat terrain after some steps, in order to evaluate the mutual information for all possible actions at the same time. To validate the approach, simulations over sinusoidal terrains and experimental tests with our synchrodrive mobile robot platform with a wide-angle camera have been performed [141].

1.4 Thesis Structure

Our intention in this work is to understand the problem of controlling a SLAM system from an automatic control theory perspective. We analyse the estimated system, doing a stability analysis of a simple linear case and performing an observability analysis for more complicated cases as the monocular SLAM. These analyses are the base of our control scheme, not only for low-level control law, but also for the active control strategy of a single camera performing SLAM in real-time and of course our vehicle moving in 3D environments as well. Throughout the examples presented we look at the insights of different models, a linear SLAM system, a planar mobile robot, and 6-DOF system with constant velocity models, using range and bearing information and the most crucial for this work, bearing-only information. The examples must be seen as part of the contribution of the work.

This work is organised as follows. This chapter served as an introduction of the themes we are going to treat further. Namely, visual navigation in mobile robotics, localisation and mapping. Also, the motivation, objectives and contributions are presented here.

Chapter 2 presents a survey on the state-of-the-art of simultaneous localisation and map building and on the visual navigation and control of mobile robots. Specifically, we present some of the challenges of simultaneous localisation and mapping,

paying more attention to the ones we treat in this work, observability in SLAM and closing the control loop.

From this point of view, the thesis is divided naturally into two parts: Estimation (Chapters 3 and 4) and Control (Chapters 5 and 6).

Chapter 3 presents an analysis of the state estimation error dynamics for a linear system within the Kalman filter based approach to Simultaneous Localisation and Map building (SLAM). Because of the inherent lack of observability, we demonstrate that such dynamics is marginally stable producing non-zero mean estimation errors. Moreover, we analyse a controllable SLAM system. Again the system is partially controllable and we propose to achieve this condition by adding pseudo noise to the map states. This addition produces inflated covariance matrices, as in the well-known method to reduce time complexity in SLAM decorrelating the covariance matrix. The estimated system could be unstable depending on whether the system is observable or not.

In **Chapter 4** the aim is to obtain a better understanding of the well known intuitive behaviour of visual SLAM systems, such as the need for triangulation to features from different positions in order to get accurate relative pose estimates. We characterise the unobservable directions using the nullspace basis of the stripped observability matrix. This approach helps us identify which vehicle motions are required to maximise the number of observable states in the system which, in turn, affects the accuracy of the estimation. Four different systems using a single camera are analysed.

Chapter 5 is devoted to the control-theoretic analysis of the closed loop system, i.e. System - Estimation - Controller. We propose two different feedback control laws in order to simultaneously estimate vehicle location, build a feature map and control the system to follow any trajectory or go to a certain point. The separation theorem of the closed loop system guarantees that the estimated system and the controlled system are stable if the system is observable, and the conclusion is that a low level controller cannot affect the estimation allowing us to propose the active control strategy of Chapter 6.

An active control strategy is presented in **Chapter 6** to reduce uncertainty during estimation in SLAM. The objective of this strategy is to optimise both the localisation of the sensor and building of the feature map by computing the most appropriate control actions or movements. Combining the nonlinear control law proposed in Chapter 5 and

this strategy we guarantee a better estimation and a stable controlled system for a mobile robot navigating with only one camera.

Chapter 7 presents the general conclusions of this thesis.

Our main sensor will be a single camera, eventhough we work as well with range and bearing measurements in some analysis. The main approach for analysis, simulations and real experiments is the EKF-SLAM based on the work done by Davison in [36].

In order to validate the strategies and control laws proposed in this thesis we developed a simulated environment in Simulink/Matlab that is explained in detail in Appendix B.

Experiments with hand-held cameras and mobile robots where performed based on the code developed by Andrew Davison at Robotics Research Group of Oxford University and extended for our purposes in this work. Appendix C presents the application and its libraries.

Chapter 2

Navigation in Unknown Environments

This chapter presents concepts, tasks and a survey related to robotic navigation. There are three main tasks which a robot must accomplish while navigating in an unknown environment: localisation, mapping, and control. The chapter is organised as follows. Section 2.1 describes some concepts about mobile robot navigation. Section 2.2 presents the state-of-the-art in SLAM and the typical EKF-SLAM algorithm used in following chapters. Section 2.3 describes related work in mobile robot control for navigation with a primary focus on visual sensors. Finally, a summary of the literature related to SLAM and control together is presented in Section 2.4.

2.1 Autonomous Mobile Robots

Campion *et al* in [24] defined an autonomous mobile robot as "... a vehicle which is capable of autonomous motion (without an external human driver) because it is equipped, for its motion, with motors that are driven by an embarked computer". We should add to this definition that the vehicle is also equipped with sensors to perceive its environment.

There are some assumptions that should be taken into account to model autonomous

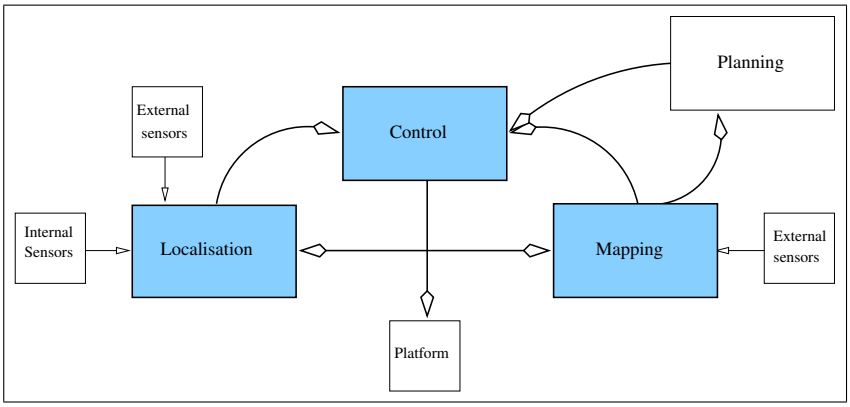


Figure 2.1: Relationship between components of an autonomous mobile robot. The term *navigation*, as used in this thesis, includes the three highlighted tasks of localisation, mapping and control.

mobile robots: the wheeled mobile robots are made up of a rigid frame with non deformable wheels, and sometimes they are moving on a horizontal plane. The property of autonomy is understood as the ability to independently make intelligent decisions as the situation changes.

Some wheeled mobile robots, such as the ones we use in this work, are a class of mechanical systems characterised by kinematic constraints that are not integrable [84]. These constraints cannot be eliminated from model equations, thus the standard planning and control algorithms developed for under-constrained robotic manipulators are difficult to apply. For instance, a vehicle with two forward drive wheels and two back side wheels cannot move sideways. That is, this type of vehicle is a nonholonomic system.

For wheeled vehicles, dead reckoning using internal sensors is not enough to achieve precise motion control because of slippage. Therefore, usage of external sensors is indispensable. Some of these sensors (e.g. cameras) can be mounted on active mechanisms such as pan and tilt necks, active heads or even manipulators. Sensors must be modelled also. The Denavit-Hartenberg convention [112] is commonly used to obtain direct and inverse kinematics of such active mechanisms.

Figure 2.1 shows a diagram of the relationship between the different functional components of an autonomous vehicle based on [43]. These components are:

- **Localisation** or position determination provides estimates of the location, altitude, velocity or acceleration of the vehicle. This ability can often proceed independently from the other system components. Errors in internal sensors usually require filtering to get better estimates of the state.
- **Mapping** is concerned with the acquisition of external sensed information. The mapping function takes input from sensors observing the operational environment. It must use this information to create an internal representation of the environment.
- **Control** responses to a desired action and could stabilise the system along a point or a trajectory. It could also generate desired actions using the information provided by estimates of the localisation and mapping tasks.
- **Mission and task planning** also refers to navigation but in large, functionally generated trajectories, behaviours or way points for the system as a whole. It has no direct links with either sensory input or controller output. However, it clearly must use an understanding of these, in conjunction with maps and defined mission objectives, to produce appropriate navigation commands.

In the literature, the term navigation is frequently applied to different concepts. When used in this thesis, navigation refers to the task of combining localisation, mapping and control. Specifically we focus on the problem where the three tasks are performed simultaneously. Therefore, navigation can be thought of as *SLAM with control*.

2.2 Simultaneous Localisation and Mapping

In the last twenty years there has been a considerable progress in the construction of local terrain maps for use in navigation and localisation. Notable is the work at CMU and JPL in stereo terrain reconstruction on grids [81], and a later work on triangular-tessellated terrain models. These methods have potential to provide quite general terrain models for both navigation and motion planning tasks. Currently however,

such methods require independent knowledge of platform location and so have limited application in localisation.

Simultaneous Localisation and Map Building in mobile robotics has been an active research topic for over fifteen years. SLAM is the process of building a map of the environment while simultaneously using this partially built map to provide vehicle localisation. Typical SLAM algorithms work by generating estimates of the vehicle and landmarks locations. It can be shown that the precision of these estimates increases monotonically and that the vehicle location estimate becomes bounded [40]. This means that a vehicle can start at an uncertain location in an unknown environment and incrementally build a convergent map while maintaining bounds on platform localisation error. Seminal work by Smith and Cheesman [122] suggested that as successive landmark observations take place, the correlation between the estimates of the location of such landmarks in a map grows continuously. It has also been shown that the absolute accuracy of the map reaches a lower bound defined only by the initial vehicle uncertainty [78].

2.2.1 Estimation in SLAM

Optimal state estimators (observers) such as the Kalman filter (KF) and its extension for nonlinear systems, the extended Kalman filter (EKF), are commonly used to solve the SLAM problem. These approaches permit to analyse convergence properties of the filtered system; at least, for the linear case. Observability and controllability are important notions to understand such systems. In a dynamic system, such as SLAM, where both the state and measurement dynamics are corrupted by noise, it is important to know whether it is possible or not to reconstruct the entire state space from output measurements. Unfortunately, the state space constructed by appending the robot pose and the landmark locations is only partially observable [2], producing a constant bounded state estimate, dependant on the initial filter conditions. Consequently, when building an absolute map, and at the same time estimating the absolute vehicle location purely from sensor measurements and odometry, the results of the Kalman filter-based approach to SLAM will be subject to the error produced at the very first iteration [104]. The issue of marginal filter stability in SLAM, previously addressed as a partial observability problem only [3], is now formalised in this thesis in Chapter 3.

The computational complexity of the SLAM estimation problem is at least $O(n^2)$, n being the total number of landmarks in the map. Further, the structure of the SLAM problem is characterised by monotonically increasing correlations between landmark estimates. Thus the state space can not be trivially decoupled. For these reasons, there has been a significant drive to find computationally effective SLAM algorithms. This has been achieved through the development and use of the KF and EKF. In these developments, simplification in the time update step and locality in the observation update step have resulted in algorithms that can process thousands of landmarks in real time on PC level architectures [57, 89]. To speed up the performance of the algorithm, some authors have proposed the use of covariance inflation methods for the decorrelation of the state error covariance [58], subject to suboptimality of the filter. However, full decorrelation of a partially observable system might lead to covariance divergence [69]. Our views on decorrelation to speed up SLAM are covered in Chapter 3.

The Kalman filter (KF) approach has a number of limitations. These include: difficulty in representing complex environment or feature models, difficulty of faithfully describing highly skewed or multi-modal vehicle error models, the use of linear propagation of covariances, and the inherent complexity of the resulting data association problem [102]. Vehicle and sensor models are usually highly nonlinear, and the effects of linearisation required in the EKF can lead to filter divergence [70].

An alternative approach to vehicle prediction, which overcomes many of these limitations, is to consider navigation as a Bayesian estimation problem [132]. In this method, vehicle motion and feature observations are described directly in terms of the underlying probability density functions and the Bayes theorem is used in a combination of grid-based environment modelling and particle filtering techniques. These Bayesian methods have demonstrated considerable success in some challenging environments [129]. Particle filters approximate the state space by random sampling of the posterior distribution, and may require many samples to accurately model the nonlinear effects in both vehicle and measurement models. A middle ground is to use a deterministic approach for the nonlinear propagation of means and covariances. One such solution is the use of the unscented Kalman filter (UKF) [67, 68, 71]. An unscented transformation is similar to a particle filter in that it samples the probability density function *pdf*, but instead of doing it randomly, a careful selection of deterministic *sigma* points is

made so as to preserve the moments of the distribution. Deterministically choosing the particles is a computationally efficient solution for the nonlinear propagation of means and covariances, but doing so for the full state vector in SLAM may not be appropriate. There is no need to use particles in the computation of the map prior, given its linear nature. Thus, by using the Unscented Transformation (UT) [67] only for the vehicle states we are able to reduce the computational complexity (compared to a full UT), and to produce, at the same time, tighter covariance estimates [6].

2.2.2 Map Representations

World-centric and Sensor-centric

One taxonomy of mapping algorithms, motivated by the reference frame to which estimates are linked, is world-centric versus robot-centric or what we call sensor-centric. World-centric maps are represented in a global coordinate space. The entities in the map do not carry information about the sensor measurements that led to their discovery. Sensor-centric maps, in contrast, are described as the relative respresentation between the map and the vehicle.

At first glance, sensor-centric maps might appear easier to build, since no *translation* of robot measurements into world coordinates are needed. However, sensor-centric maps suffer a disadvantage. It is often difficult to extrapolate from individual measurements to measurements at nearby, unexplored places; an extrapolation that is typically straightforward in world-centric approaches. In other words, there is usually no obvious geometry in measurement space that would allow for such extrapolation. For this reason, the dominant approaches to date generate world-centric maps even though we will see further in this thesis there is an inherent lack of observability in this representation.

Geometric and Topological Maps

Another common map representation in SLAM has regard to the types of information that link nearby landmark estimates. These can be either geometric and topological maps. A geometric map represents objects according to their absolute geometric relationships. It can be a grid map, or a more abstract map, such as a line map [87] or a polygon map [56]. Geometric maps derived from a sensor must be matched against past sensed landmarks in global coordinates. This is often of great difficulty because

of the robot's accumulated position error. The simplest way to represent a geometric map is the occupancy grid-map. The first such map representation that appeared in the literature (in conjunction with mobile robots) was the Certainty Grid developed by Moravec and Elfes [99]. In the Certainty Grid approach, sensor readings are placed into the grid by using probability profiles that describe the algorithm's certainty about the existence of objects at individual grid cells. Borenstein and Koren [13] refined the method with the Histogram Grid, which derives a pseudo-probability distribution out of the motion of the robot. Scan matching is also applied to build maps in SLAM in [51] .

By contrast, the topological approach is based on recording the geometric relationships between observed features rather than their absolute position with respect to an arbitrary coordinate frame of reference [28, 50]. The resulting presentation takes the form of a graph where the nodes represent the observed features and the edges represent the relationships between the features. Unlike geometric maps, topological maps can be built and maintained without estimates for the position of the robot. This means that the errors in this representation will be independent of any errors in the estimates for the robot position [127]. Information Filters are commonly used to build this kind of maps [133].

The representation adopted in this work is geometrical, considering 2D or 3D points as the landmarks of our maps.

2.2.3 Challenges

According to [130], a key challenge in SLAM arises from the problems of modelling measurement noise. If the noise in different measurements is statistically independent the mapping problem will be easy to solve. A robot could simply take more and more measurements to cancel out the effects of noise. Unfortunately, the measurement errors are statistically dependent. This is because errors in control accumulate over time, and they affect the way future sensor measurements are interpreted. A filter should be capable of overcoming these problems, and also to deal with problems arising from the nonlinearities of the process and measurement models and finally with the fact that the system is not completely observable.

Despite the large amount of success in developing SLAM systems, the issue of consistency of the EKF-SLAM algorithm is still a challenge [26, 70]. We know that in the long run, variations in Jacobian linearisation introduce spurious information that will eventually produce an optimistic filter, and that this inconsistency is especially relevant in the case of rotations [8]. And in the simplest case of a stationary robot, the EKF-SLAM will produce optimistic estimates because the Jacobians are being evaluated at different estimation values of the same state [63].

The second complicating aspect of the SLAM problem arises from the high dimensionality of the statistical map of the features. If the robot confines itself to the description of major topological features, such as corridors, intersections, rooms and doors, a few dozen features might suffice. A detailed two-dimensional floor plan, which is an equally common representation of robotic maps, often requires thousands of them. But a detailed 3D visual map of a building (or of an ocean floor) may easily require millions of features. From a statistical point of view, each such number is a dimension of the underlying estimation problem. Thus, the mapping problem can be extremely high dimensional and there have recently nice atempt to tackle it [52]. Submapping approaches can help to manage the huge amount of data [15, 45].

A third and possibly the hardest problem in SLAM is the correspondence problem, also known as the data association problem. The correspondence problem is the problem of determining if sensor measurements taken at different points in time correspond to the same physical object in the world. When a robot attempts to map a large cyclic environment and it tries to close the cycle, the robot has to find out where it is relative to its previously built map. This problem is complicated by the fact that at the time of loop closing, the robot's accumulated pose error might be unboundedly large. The correspondence problem is difficult, since the number of possible hypotheses can grow exponentially over time. Neira *et al* propose the joint compatibility test to solve the data association using a greedy algorithm using branch and bound strategies to solve wrong associations [102].

Fourth, environments change over time. Some changes may be relatively slow, such as the change of appearance of a tree across different seasons, or the structural changes that most office buildings are subjected to over time. Others are faster, such as the change of door status or the location of furniture items, such as chairs. Even faster may

be the change of location of other agents in the environment, such as cars or people. The dynamism of robot environments creates a big challenge, since it adds yet another way in which seemingly inconsistent sensor measurements can be explained. This becomes evident in the case of a robot facing a closed door that previously was modeled as open. Such an observation may be explained by two hypotheses, namely that the door status changed, or that the robot is not where it believes to be. Unfortunately, there are almost no mapping algorithms that can learn meaningful maps of dynamic environments [51]. Instead, the predominant paradigm relies on a static world assumption in which the robot position is the only time-variant quantity (and everything else that moves is just noise). Consequently, most techniques are only applied in relatively short time windows, during which the respective environments are static.

A fifth challenge arises from the fact that robots must choose their way during mapping. The task of generating robot motion in the pursuit of building a map is commonly referred to as autonomous exploration, addressed to navigation control and planning [108, 125, 134]. In order to build a map, the robot must explore its environment to map uncharted areas. Typically it is assumed that the robot begins its exploration without having any knowledge of the environment. Then, a certain motion strategy is followed which could aim for example, at maximising the amount of charted area in the least amount of time. Such a motion strategy is called exploration strategy, and it depends strongly on the kind of sensors used. While optimal robot motion is relatively well-understood in fully modeled environments, exploring robots have to cope with partial and incomplete models. Hence, any viable exploration strategy has to be able to accommodate contingencies and surprises that might arise during map acquisition. For this reason, exploration is a challenging planning problem, which is often solved sub-optimally via simple heuristics.

Many researchers believe that no single sensor modality alone adequately captures all relevant features of a real environment. To overcome this problem, it is necessary to combine data from different sensor modalities, a process known as *sensor fusion*, the sixth challenge. Buchberg *et al.* [20] and [72] developed a mechanism that utilises heterogeneous information obtained from a laser-radar and sonar system in order to construct a reliable and complete world model. Nieto *et al* [105] proposed what they called DenseSLAM, in which they enrich their geometric maps with the information of a

radar. Courtney and Jain [32] integrated three common sensing sources (sonar, vision, and infrared) for sensor-based spatial representation. They implemented a feature-level approach to sensor fusion from multisensory grid maps using a mathematical method based on spatial moments and moment invariants.

Today, mapping is largely considered the most difficult perceptual problem in robotics. Progress in SLAM is bound to impact a much broader range of related perceptual problems, such as sensor-based manipulation and interaction with people.

In this thesis we will focus mainly in first and fifth of the above challenges, tackling them from a control theory viewpoint without disregarding the probabilistic approach.

2.2.4 Sensors and Techniques

Visual sensing is the most information-rich modality for navigation in everyday environments. However, recent advances in simultaneous localisation and map building for mobile robots have been made using sonar and laser range sensing to build maps in 2D that have been largely overlooked in the vision literature. Durrant-Whyte implemented systems using a wide range of vehicle and sensor types [44] and are currently working on ways to ease the computational burden of SLAM. Chong and Kleeman achieve nice results using advanced tracking sonar and accurate odometry combined with a submapping strategy [27].

Thrun *et al* [128] have produced some demonstrations of robot navigation in real environments using a laser range finder and mapping with some vision. In a different work [131], they used scan matching of a laser range finder mounted in a pan head. Castellanos [25] also uses a laser range finder and a mapping strategy called the SPmap. This group is also working with a strategy that they called Hierarchical SLAM [45].

Leonard and Feder [88], worked primarily with underwater robots and sonar sensors, have recently proposed submapping ideas, breaking a large area into smaller regions for more efficient map-building [14]. The philosophy of *constant time* SLAM, also proposed by Leonard and Newman [86], is to maintain the consistency, look for global convergence and an algorithm computationally more efficient using local submaps. More recently the well-known information form of the EKF, also known as the extended information filter (EIF) has been used. This filter maintains an inverse representation of the normal

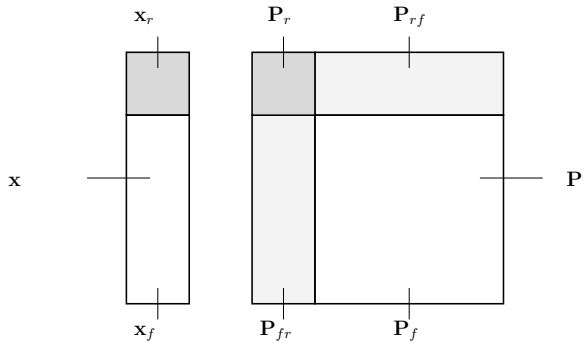


Figure 2.2: The EKF-SLAM state vector and covariance matrix representation.

pdf, in which the covariance matrix is substituted by its inverse, the information matrix, and the mean vector by the information vector. Thrun *et al* in [133] proposed an EIF that maintains a sparse information matrix. The sparsity has important ramifications on the computational properties of solving SLAM problems. Eustice and Leonard presented the exact SEIF using a delayed-state framework [46].

Murray and Davison made the first application of active vision to real-time, sequential map-building within a SLAM framework [37]. They showed that active visual sensing is ideally suited to the exploitation of sparse *landmark* information required in robot map-building. Posterior works by Davison concentrate in localisation and mapping with one wide-range camera for any kind of robot or even carried by a person [36]. Kim and Sukkarieh used also vision and inertial measurements to localise an aerial vehicle and build a terrain map [76].

More common is to find applications using laser range finder, but the tendency is now to solve the problem using vision or more than one sensor in order to obtain 3D representations of the environment.

2.2.5 SLAM solution using the EKF

The SLAM problem is commonly solved using EKF techniques. Its theoretical meaning and its simplicity makes it advantageous over other filters or estimators. This is the approach we take in the rest of the thesis. This section presents models and EKF for the case of the SLAM problem.

System model

In the general SLAM case, the motion of the vehicle and the measurement of the map features are governed by the discrete-time state transition model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{2.1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \,. \tag{2.2}$$

The state vector $\mathbf{x}_k \in \mathbb{R}^{m+dn}$ contains the position of the vehicle $\mathbf{x}_{r,k} \in \mathbb{R}^m$ at time step k, and a vector of n stationary d-dimensional map features $\mathbf{x}_f \in \mathbb{R}^{dn}$,

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{r,k} \\ \mathbf{x}_f \end{bmatrix} . \tag{2.3}$$

The input vector $\mathbf{u}_k \in \mathbb{R}^l$ is the vehicle control command and $\mathbf{v}_k \in \mathbb{R}^l$ is a Gaussian random vector with zero mean and a covariance matrix $\mathbf{Q} \in \mathbb{R}^{l \times l}$, representing unmodeled vehicle dynamics and thence system process noise. The function $\mathbf{f} : \mathbb{R}^{m+dn} \rightarrow \mathbb{R}^{m+dn}$ is a possibly nonlinear difference equation that describes the motion of the vehicle.

The Gaussian random vector $\mathbf{w}_k \in \mathbb{R}^{dn}$ represents both the inaccuracies of the also possibly nonlinear measurement model $\mathbf{h} : \mathbb{R}^{dn+m} \rightarrow \mathbb{R}^{dn}$ and the measurement noise with zero mean and covariance matrix $\mathbf{R} \in \mathbb{R}^{dn \times dn}$.

Provided that a set of measurements $Z^i = \{\mathbf{z}_1, \ldots, \mathbf{z}_i\}$ was available for the computation of the current map estimate $\mathbf{x}_{k|k}$, the expression

$$\mathbf{x}_{k+1|k} = \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0}) \tag{2.4}$$

gives an *a priori* noise-free estimate of the new locations of the robot and map features after the vehicle control command $\mathbf{u}_k$ is input to the system. Similarly,

$$\mathbf{z}_{k+1|k} = \mathbf{h}(\mathbf{x}_{k+1|k}) + \mathbf{0} \tag{2.5}$$

constitutes a noise-free *a priori* estimate of sensor measurements.

The EKF approach to SLAM requires linearisation of both the plant and measurement models. Such linearisations are formulated as Taylor series approximations with the higher order terms dropped,

$$\mathbf{x}_{k+1} \approx \mathbf{x}_{k+1|k} + \mathbf{F}(\mathbf{x}_k - \mathbf{x}_{k|k}) + \mathbf{G}\mathbf{v}_k \tag{2.6}$$

$$\mathbf{z}_{k+1} \approx \mathbf{z}_{k+1|k} + \mathbf{H}(\mathbf{x}_{k+1} - \mathbf{x}_{k+1|k}) + \mathbf{w}_{k+1}\,. \tag{2.7}$$

The Jacobian matrices $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ contain the partial derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$ and the noise $\mathbf{v}$, and of $\mathbf{h}$ with respect to $\mathbf{x}$, respectively:

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0})} \tag{2.8}$$

$$\mathbf{G} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right|_{(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0})} \tag{2.9}$$

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{(\mathbf{x}_{k+1|k}, \mathbf{0})}\,. \tag{2.10}$$

Given that the landmarks are considered stationary, their *a priori* estimate is simply

$$\mathbf{x}_{f,k+1|k} = \mathbf{x}_{f,k|k}\,. \tag{2.11}$$

Thus, the elements of the non-stationary linear state transition model of the vehicle and map dynamics in (2.6) and (2.7) take the form

$$\begin{bmatrix} \mathbf{x}_{r,k+1} \\ \mathbf{x}_f \end{bmatrix} \approx \begin{bmatrix} \mathbf{x}_{r,k+1|k} \\ \mathbf{x}_{f,k|k} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{F}_r & \\ & \mathbf{I} \end{bmatrix}}_{\mathbf{F}} \begin{bmatrix} \tilde{\mathbf{x}}_{r,k|k} \\ \tilde{\mathbf{x}}_{f,k|k} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{G}_r \\ \mathbf{0} \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \mathbf{v}_k \\ \mathbf{0} \end{bmatrix}, \tag{2.12}$$

$$\mathbf{z}_{k+1} \approx \mathbf{z}_{k+1|k} + \underbrace{\begin{bmatrix} \mathbf{H}_r & \mathbf{H}_f \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} \tilde{\mathbf{x}}_{r,k+1|k} \\ \tilde{\mathbf{x}}_{f,k+1|k} \end{bmatrix} + \mathbf{w}_{k+1}\,. \tag{2.13}$$

In the remaining of this thesis, extensive use of KF related notation will be used. The Kalman filter algorithm is described in Appendix A.1. The EKF requires Gaussian

representations for all the random variables that form the map (the vehicle pose and all landmark's positions). Moreover, their variances need to be small to be able to approximate all the non linear functions with their linearised forms.

In terms of covariance matrices the Kalman filter steps are: The *a priori* state covariance prediction,

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^{\top} + \mathbf{Q}_k, \tag{2.14}$$

the Innovation covariance

$$\mathbf{S}_{k+1} = \mathbf{R}_{k+1} + \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^{\top} \tag{2.15}$$

and finally the filter gain

$$\mathbf{K} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^{\top}\mathbf{S}_{k+1}^{-1} \tag{2.16}$$

The update of the state covariance is computed with

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{S}_{k+1}\mathbf{K}^{\top} \tag{2.17}$$

The steady state value for the covariance matrix is given by the solution of the Riccati equation

$$\mathbf{P} = \mathbf{F}(\mathbf{P} - \mathbf{P}\mathbf{H}^{\top}(\mathbf{H}\mathbf{P}\mathbf{H}^{\top} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P})\mathbf{F}^{\top} + \mathbf{Q}. \tag{2.18}$$

Such solution to the Riccati equation will converge to a steady state covariance only if the pair $\{\mathbf{F}, \mathbf{H}\}$ is *completely observable* [9]. This condition is not satisfied in general in SLAM, and for the linear case, the solution of (2.18) is a function of $\mathbf{P}_{r,0|0}$, $\mathbf{Q}$, $\mathbf{R}$, and the total number of landmarks n. Note however that, for the nonlinear case, the computation of the Jacobians $\mathbf{F}$ and $\mathbf{H}$ will in general also depend on the steady state value of $\mathbf{x}$.

The objective of SLAM is to estimate $\mathbf{x}_k$ properly using, for instance the EKF that recursively computes a minimum mean square error (MMSE) estimate. Figure 2.3 shows a schematic representation of the SLAM algorithm. The MMSE is designated

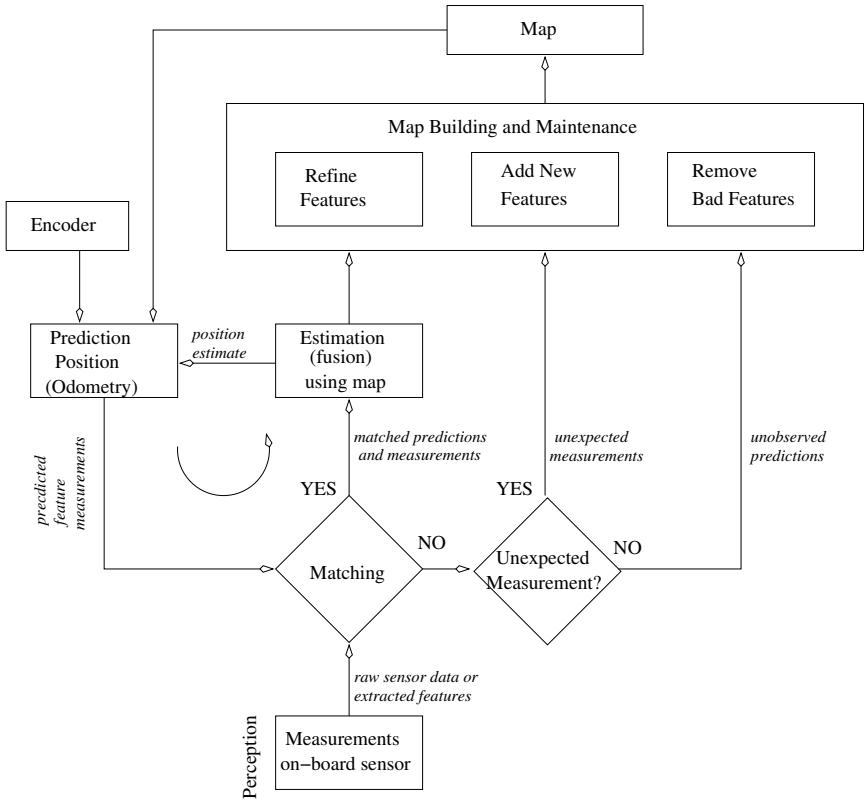


Figure 2.3: A flow chart for a generic SLAM algorithm. This representation does not depend on the type of representation and filter used to store localisation and map information [116].

by the system state estimate $\mathbf{x}_{k|k} = [\mathbf{x}_{r,k|k}, \mathbf{x}_{f,k|k}]^\top$, and its state covariance matrix,

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{r,k|k} & \mathbf{P}_{rf,k|k} \\ \mathbf{P}^\top_{rf,k|k} & \mathbf{P}_{f,k|k} \end{bmatrix}, \tag{2.19}$$

where the mean $\mathbf{x}_{r,k|k}$ and covariance $\mathbf{P}_{r,k|k}$ of a vehicle part of the state estimate, and the mean $\mathbf{x}_{f,k|k} = [\mathbf{x}^1_{f,k|k}, \ldots, \mathbf{x}^i_{f,k|k}]^\top$ and covariance $\mathbf{P}_{f,k|k}$ of the map part of the state estimate, and the cross covariance term $\mathbf{P}_{rf,k|k}$ between vehicle and map are related as illustrated in Figure 2.2.

Landmark Initialisation

Given a set of i measurements $Z^i = [\mathbf{z}_1; \mathbf{z}_2; \ldots \mathbf{z}_i]$ we wish to initialise a new entity $\mathbf{x}^i_f$ (feature) into a stochastic map modelled as a mean $\mathbf{x}_{k|k}$ and covariance matrix $\mathbf{P}_{k|k}$.

Initialisation consits of stacking the new landmark position $\mathbf{x}_f^i$ into the map as

$$\mathbf{x}^+ = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_f^i \end{bmatrix} \tag{2.20}$$

and defining the *pdf* of this new state conditioned on measurement $\mathbf{z}_i$ [103]. Considering the case of any given measurement model a implicit expression that relates a new variable $\mathbf{m}_f$ is defined as,

$$\mathbf{h}'(\mathbf{x}_r, \mathbf{x}_f, \mathbf{m}_f) = 0 \tag{2.21}$$

the measurement associated to a new feature, where $\mathbf{z} = \mathbf{m}_f + \mathbf{v}_k$. Solving $\mathbf{h}'$ for fixed values of $\mathbf{x}_r$ and $\mathbf{m}_f$, in order to get an explicit expression for a new feature

$$\mathbf{x}_f = \mathbf{j}(\mathbf{x}_r, \mathbf{m}_f)\,. \tag{2.22}$$

Note that for range and bearing measurements the solution of this expression is a point. We assume $\mathbf{x}_f$ is approximately a Gaussian random variable with mean and covariances matrices defined by

$$\mathbf{x}_{f,k|k} = \mathbf{j}(\mathbf{x}_{r,k|k}, \mathbf{z}) \tag{2.23}$$

$$\mathbf{P}_f = \mathbf{J}_r \mathbf{P}_r \tag{2.24}$$

$$\mathbf{P}_{ff} = \mathbf{J}_r \mathbf{P}_{rr} \mathbf{J}_r^\top + \mathbf{J}_m \mathbf{P}_{rr} \mathbf{J}_m^\top\,, \tag{2.25}$$

where $\mathbf{P}_r = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rf} \end{bmatrix}$ and $\mathbf{J}_r = \partial \mathbf{j}/\partial \mathbf{x}_r$ and $\mathbf{J}_m = \partial \mathbf{j}/\partial \mathbf{m}_f$. The augmented map (see Figure 2.4) is finally specified by

$$\mathbf{x}_{k|k}^+ = \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{x}_{f,k|k}^i \end{bmatrix} \quad \mathbf{P}_{k|k}^+ = \begin{bmatrix} P_{k|k} & P_{f,k|k}^\top \\ P_{f,k|k} & P_{ff,k|k} \end{bmatrix}\,. \tag{2.26}$$

For bearing-only measurements the explicit solution of (2.22) turns out a line. The feature measurement model cannot be directly inverted to give the position of a new feature given an image measurement and the camera position, since the feature depth is unknown. Estimating the depth of a feature will require camera motion and several measurements from different viewpoints. Image features could be extracted as is

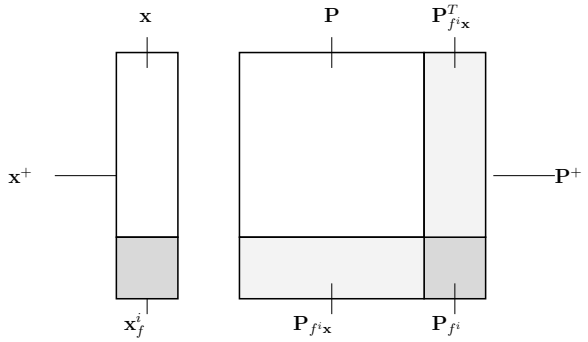


Figure 2.4: Initialisation for EKF-SLAM: state vector and covariance matrix representation.

presented in 2.3.1.

2.3 Control Using Vision

It is very interesting why sensing the environment and subsequently navigating is such a difficult task for mobile robots; after all, it is something which is taken for granted as easy for humans or animals, who have no trouble moving through unfamiliar areas, even if they are perhaps different from environments usually encountered.

Early vision systems developed for mobile robot navigation relied on the geometry of space and other metrical information for driving the vision processes and performing localisation [98]. In particular, interior spaces are represented by CAD models of varying complexity. CAD models were replaced by simpler models, such as occupancy maps, topological maps or even sequences of images. When sequences of images were used to represent space the images taken during navigation were submitted to some kind of appearance-based matching between the perception (actual image) and expectation (goal image or goal images stored in a data base).

Visual Navigation usually has been approached by the use of landmarks or visual features. Several algorithms have been presented in the literature which use visual landmarks for navigation [79], [113], [114]. Visual features are generally represented by three-dimensional (3D) models (set of lines or points) in the environment space, which have to be matched with corresponding two-dimensional (2D) models in the image

plane [90].

Unknown environments require more robust algorithms for navigation such as feedback control methods, with vision used as the primary sensor. The following subsection presents a summary of the most representative approaches for visual control.

2.3.1 Landmarks

Landmarks are distinct features that a robot can recognise from its sensory input. Landmarks can be geometric shapes such as rectangles, lines, circles, etc. and they may include additional information (e.g., in the form of a bar-code). In general, landmarks have a fixed position, relative to which a robot can localise itself. Landmarks are carefully chosen to be easily identifiable; for example, there must be sufficient contrast to the background. Before a robot can use landmarks for navigation, the characteristics of the landmarks must be known and stored in the robot's memory. The main task in localisation is then to recognise the landmarks and to calculate the robot's position with respect to them. In the same way in mapping, the robot should calculate the landmark position with respect to a fixed reference frame or with respect to itself.

If the robot's pose was known all along, building a map would be quite simple. Conversely, if we already had a map of the environment, there exist compositionally elegant and efficient algorithms for determining the robot's pose at any point in time [42, 51, 74, 126]. In combination, however, the problem is much harder.

In order to simplify the problem of landmark acquisition it is often assumed that the current robot pose is known approximately, so that the robot only needs to look for landmarks in a limited area. For this reason good odometry is required and the use of filters to estimate the robot pose, and in the case of mapping, to estimate landmark positions is necessary.

We can address the problem for two types of landmarks: "artificial" and "natural". Natural landmarks work best in highly structured environments such as corridors, manufacturing floors, or hospitals. Borenstein in [12] defined *natural landmarks* as those objects or features that are already in the environment and have a function other than robot navigation; and *artificial landmarks* as specially designed objects or markers that need to be placed in the environment with the sole purpose of enabling robot navigation.

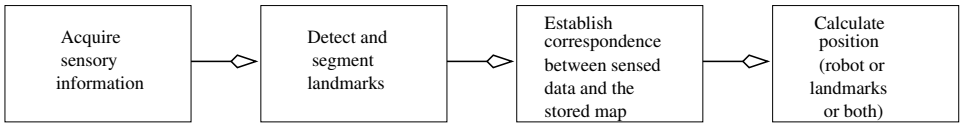


Figure 2.5: Procedure for landmark-based positioning [12].

The general procedure according to Borenstein *et al* [12], for performing landmark-based position is shown in Figure 2.5. If the purpose is map building, the landmark position is also calculated. They use sensors to sense the environment and then extract distinct structures that serve as landmarks for navigation in the future.

Information to do path planning, obstacle avoidance, and other navigation task must also be easily extractable from the spatial representation used. It is possible to use one or more landmarks, or even to create maps of landmarks to perform the desired task. For example, line maps could be useful for wall-following [91] and in this case the walls are natural landmarks. There are many examples of natural landmarks: trees, lampposts, furniture, lamps and so on. These landmarks are commonly chosen depending on the environment (e.g. indoor or outdoor). The main challenge is to extract those landmarks from the entire scene in a robust way in order to generate a spatial representation.

Image features

Salient points in an image are those that are locally distinguishable, that possess some strong particularity that make them unique in some sense. These points are usually called features, and the process of identifying them in an image is normally referred to as feature detection. In gray-level images a feature may correspond to a point where the luminosity variation in all directions is locally maximal. The analysis of the image derivatives is at the heart of most of the best known feature detectors. The task of finding the same salient point in another image is usually called feature matching. From the above statements, the uniqueness of a feature is determined by its close neighbourhood. If we memorise this neighbourhood as the feature's signature, for example in the form of a small rectangular patch like those in [37], we will be able to find it in other images by simply scanning the new image for a similar enough signature.

Discrete features are easier to represent than continuous objects or regions, and

provide unambiguous information for localisation. The spatial density of features used can differ depending on how well it is required to represent the world, and individual ones can easily be added or deleted as necessary.

The features which will make up the sparse map used for localisation in this work need to be landmarks for the vehicle: they must be reliably and repeatedly detectable and measurable, and preferably from a wide range of vehicle positions. This differs slightly from the requirements of the features used in structure from motion algorithms where it is not usually intended that the same features (usually "corners" or line segments) be detected and matched over long periods, indeed this is typically not possible because with a passive camera features can go out of view very quickly. They must be features present in a normal environment so that no artificial interference such as positioning of beacons is required. They must be stationary in the scene (or potentially have known motions, in work beyond the scope of this thesis). They must be easily identifiable from a variety of ranges and angles, and not frequently be occluded.

That is, without neither detecting nor recognising objects in the images, a vehicle must be able to select interesting points in its surrounding 3D world and track them for a certain number of images. In later chapters in this thesis we will see how to use this power to localise these points in the 3D space, building with them a sort of map, while simultaneously using every new observation to get self-localised in this map.

To detect good features automatically in an image, one of the first works was the Harris corner detector [60]. To identify a salient point in a grayscale image the idea is to select those points where the gray level spatial derivatives are locally maximal, based on the eigenvalues of the image Hessian. This is a corner if this rate of change is maximum in all directions. Or an edge when this rate of change is important only in one direction.

A more confident criterion is proposed by Shi and Tomasi in [115], this feature detector is very similar to the Harris corner detector, but Shi and Tomasi apply it to patches. They derive the operator showing that it finds features which will be intrinsically easy to track using their gradient-based search method. The idea is to find image patches which will make good features due to their high variation in intensity, these stand out well from their surroundings. By taking a uniform square mask, the Shi and Tomasi measure optimises the selectivity of correlation-based matchings with

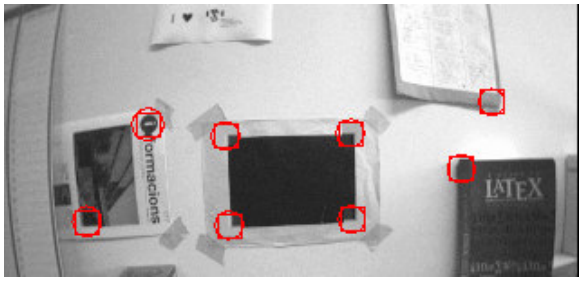


Figure 2.6: Image square patches (11 × 11 pixels) used as long-term landmarks features. Using MonoSLAM software from Andrew Davison, Imperial College.

patches of the same size of the mask. That is, in order to optimise matching, the weighting mask is chosen to be uniform and of the same size as the feature patch descriptor.

Feature matching may be performed by exploiting different principles (geometric, appearance, object-recognition, etc.). We concentrate on the use of appearance to feature matching, as they are robust yet easy to define and fast to compute. The feature's appearance could be described by a medium-sized rectangular patch in the vicinity of the corner pixel detected by the feature detector mentioned above, and shown as the red squares in Figure 2.6 [37]. In consecutive images, every pixel is assigned a patch of the same size to that of the reference patch. A similarity measure may be assigned to each pixel in the new image. This measure is computed from the appearances of the reference patch and the pixel's associated patch. The pixel that originated a particular patch is named the base pixel of the patch, which is normally its central pixel.

Different appearance-based similarity measures may be defined between two equally-sized patches, we will focus on a search using a normalised sum-of-squared-difference: the patch detection algorithm is not used on the new image. This differs from many structure from motion systems where the feature detector is run on each new image: features are found as local maxima of the particular feature-finding operator, and then correlation matching is carried out between the features found in the current and previous images, all of which are local maxima. Some authors [29] have found that requiring all patches matched to be maxima like this is restrictive, and features can be missed which still give good correlation matches. This is a good point when applied to an

active system, like the control we propose in this thesis, since it is important to keep track of the same features for as long as possible. Since a search in a particular image is only for one feature at a time, it makes sense to move straight to correlation matching.

One advantage of applying the feature detector to all images is that most detectors return feature locations to sub-pixel accuracy in a natural way. It is more difficult to see how to obtain sub-pixel accuracy from the correlation search used in Davison work [37]: interpolation can be used, but the form of the function to fit to image values depends on the image itself.

2.3.2 Visual Motion Control

The problem of navigation to a specific pose or trajectory has generally been approached from two directions; the open loop strategies and the feedback control strategies. Open loop strategies seek to find a bounded sequence of control inputs, driving the vehicle from an initial position to some arbitrary position, usually working in conjunction with a motion planner [83, 100]. Reactive, feedback control systems, on the contrary, use the environment itself for navigation. However, due to the limitations presented by Brockett in [17], there is no smooth, continuous control law which can locally stabilise closed loop nonholonomic systems to a point. These limitations can be overcome by either relaxing the constraints on the desired pose, i.e. stabilising to a point without a guarantee on orientation [82, 110], using discontinuous control techniques [31], or by using time-varying control (see [110]).

Similarities between the open loop strategies and those more rigorous control based visual servoing algorithms are abundant. However, the closed loop control-based strategies are still in their infancy with regard to effective use of a vision system.

Vision-based robot control (generally called visual servoing) concerns several fields of research including computer vision, robotics and automatic control. Visual servoing can be useful for a wide range of applications and it can be used to control many different dynamic systems (manipulator arms, mobile robots, aircraft, etc.). Visual servoing systems are generally classified depending on the number of cameras, on the position of the camera with respect to the robot or on the design of the error function to minimise in order to reposition the robot.

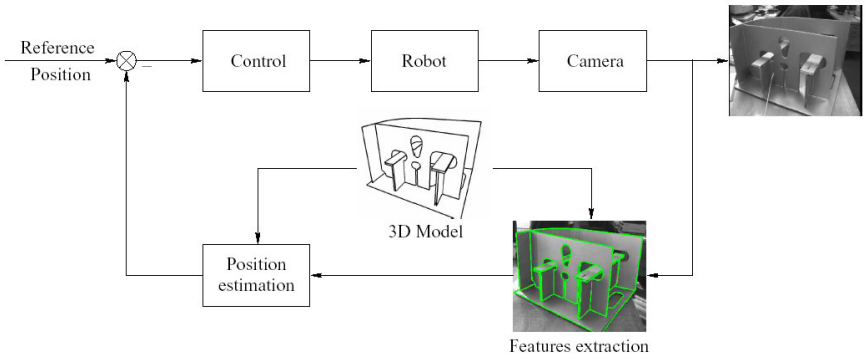


Figure 2.7: 3-D Visual servoing [94].

According to the position of the camera there exist two configurations: end-effector mounted (eye-in hand), or fixed in the workspace. There also exist hybrid systems where one camera is in-hand and another camera is stand-alone observing the scene, but these are less common.

Existing eye-in hand approaches are typically classified in different categories: position based, image based, hybrid based and motion based control systems. In a position based control system, the input is computed in the three-dimensional Cartesian space. The pose of the target with respect to the coordinate system of the camera is estimated from image features corresponding to the perspective projection [48] of the target in the image. In this approach knowledge of a perfect geometric model of the object is necessary (see Figure 2.7).

Recently, a 3D visual servoing scheme which can be performed without knowing the 3D structure of the target, has been proposed in [10]. The rotation and the direction of translation are obtained from the essential matrix [65]. The essential matrix is estimated from the current and reference images of the target [48] . However, as for the previous 3D visual servoing, such a control vector does not ensure that the considered object will always remain in the camera field of view, particularly in the presence of important camera or robot calibration errors.

Visual Servoing [66] is suitable for control of a mobile robot based on an on-board camera, because it can avoid the dead reckoning problem and reduce the cost to identify the absolute location. A lot of visual servo controllers had been developed by the

research community.

Several examples of visual servoing systems can be found in [61] and in the special issues on visual servoing which have been published in international journals. The first one, published in the IEEE Transactions on Robotics and Automation, October 1996, contains a detailed tutorial [66]. A special issue on visual servoing appeared in 2003 in the International Journal of Robotics Research [80]. In this issue a table with most of the authors dedicated to VS appears, displaying the specific problem dealt with, degrees of freedom and kind of VS applied. Malis presents an excellent survey in [95].

Based on the approaches mentioned above, a lot of work for mobile robotics has been published, or for a similar underactuated kinematics as the case of [93, 145, 146]. Mahony and Hamel developed a semi-global asymptotic visual servoing result for unmanned aerial vehicles that tracked parallel co-planar linear visual features and Zhang and Ostrowsky used a vision system to navigate a blimp.

Pissard-Gibollet and Rives analysed in [107] an image based visual servoing for a hand-eye system mounted on a mobile robot. Years later, they developed a visual servoing for the same system using time-variant control [135]. In [59], a monocular vision system is mounted on a pan and tilt to generate image-Jacobian and geometry based controllers by using different snapshots of the target and an epipolar constraint. A drawback of this method is stated in [21], the system became numerically ill-conditioned for large pan angles. Specifically, they used a teaching and replay phases to facilitate the estimation of the unknown object height parameter in the image-Jacobian by solving a least-squares problem. Burschka and Hager in [22] used a spherical image projection of a monocular vision system to overcome the above limitation. They are also working on building maps whilst controlling the robot using vision [23]. Ma *et al* [91] considered the problem of estimation from vision measurements and the design of the control strategy jointly. Steering control was used by Murray and Davison for visual fixation of a target in [37].

There is a lot of research which has focused in the use of vision system for navigating a wheeled mobile robot. Kim *et al* [75] proposed a tracking based on monocular visual feedback using consecutive image frames and an object data base. This was achieved with the use of an EKF. Das *et al* in [33] also used EKF techniques and feedback from a monocular omnidirectional camera system to enable wall following, follow-the-leader

and position regulation tasks. In [136] a method to stabilising a wheeled mobile robot to a target pose based on the discrepancies between a target view of the landmarks in the workspace and the robot's current view, is presented. The method combines non-linear control theory with research derived from hypothesis on insect navigation. Song and Huang [124] and use spatiotemporal apparent velocities obtained from an optical flow of successive images of an object to estimate the depth and time-to-contact to develop a monocular vision guided robot. Dixon *et al* in [41] used feedback from an uncalibrated, fixed (ceiling-mounted) camera to develop an adaptive tracking controller for a vehicle that compensated for the parametric uncertainty in the camera and the wheeled mobile robot dynamics. In [143], the authors exploited a rigid body transformation to develop a visual servoing mobile robot tracking controller (the regulation problem was not solved due to restrictions on the reference trajectory) that adapted for the constant, unknown height of an object moving in a plane. Structure from motion algorithms have also been used for vehicle guidance (e.g. [142]).

In this work we use a control scheme similar to the one in Figure 2.7. Only a single camera is considered and there is no 3D model as a reference. The combination of SLAM and control allows the visual navigation in unknown environments to be more robust to feature occlusions in contrast with visual servoing techniques.

2.4 SLAM with Control

Actively controlls the robot while mapping has recently received more attention, in contrast to the purely estimation process that received much attention during the past 20 years. Noteworthy, particularly given the probabilistic nature of the Bayesian approach to the solution of the SLAM problem, entropy reduction has recently gained popularity as a map building strategy for driving a robot during a SLAM session in order to minimise uncertainty [16, 49, 117]. Previous approaches to closing the control loop in SLAM were by incorporating visual servoing techniques [23], or by implementing a PD controller over an A* searched trajectory [134].

Action evaluation with respect to information gain has been implemented for SLAM systems, but little to no effort has been expended on the real-time constraint. One such approach makes use of Rao-Blackwellized particle filters [125]. When using particle

filters for exploration, only a very limited number of actions can be evaluated due to the complexity in computing the expected information gain. The main bottleneck is the generation of the expected measurements each action sequence would produce, which is generated by a ray-casting operation in the map of each particle. In contrast, measurement predictions in a feature-based EKF implementation can be computed much faster, having only one map posterior per action to evaluate, instead of the many a particle filter requires. Moreover, in [125] the cost of choosing a given action is subtracted from the expected information gain with a user selected weighting factor.

Sim has addressed decision making for the robot exploration problem, as an optimisation problem for a restricted hand-crafted set of exploratory policies [118], as a sequential decision making problem (POMDPs) [120], and by updating an information surface in a SEIF implementation [119]. These contributions, however, only test the strategies for very small planar point-based simulated environments, and remain to be tested in real-world applications. In order to avoid local maxima, the approach presented in [119] explicitly avoids loop closing by discarding repeated poses during trajectory search.

In [64] the authors analyse the possibility and necessity of multi-step look-ahead trajectory planning in SLAM. The problem addressed is similar to that of [49] and [16]. The objective of trajectory planning is to minimise the estimation error spanning in a finite time horizon. It is shown that multi-step look-ahead is possible when the current estimation error is small because it optimises the mean value of the performance metric and the most interesting issue is the optimisation strategy for the trajectory planning problem using a variant of the Model Predictive Control (MPC) with the optimality proven.

Other approaches include, for example, a multirobot stereo-vision occupancy grid-based SLAM system [109], with best single-step look ahead chosen on the basis of overall map entropy reduction. In such a discrete representation of the map posterior, overall map entropy is computed as the sum of individual entropies for each grid cell. Bryson and Sukkarieh on the other hand, present simulated results of the effect different vehicle actions have with respect to the entropic mutual information gain [18]. The analysis is performed for a 6-DOF aerial vehicle equipped with one camera and an inertial sensor,

for which landmark range, azimuth, and elevation readings are simulated, and data association is known.

In this thesis we want to focus on action decision problem in terms of uncertainty reduction considering real-time constraints. Our experience with real-time vision-based SLAM has shown us that short loop closing is essential for consistent bearings-only mapping. Whaite *et al.* suggests that a gradient strategy for uncertainty reduction would not falter on top of a local maximum [144]. The reason being that the information surface being ascended is continuously changing as new data are added. Maximally informative posteriors come from locations with large variance, and when measurements iterate over the same states, the prediction variance will be reduced to the level of sensor noise, flattening the information surface with the effect of *pushing* the robot away from that location. Consequently, in this work, we propose concentrate in a greedy real-time steepest descent approach to entropy reduction for a monocular SLAM systems, rather than on planning for large sequences of actions.

Chapter 3

Steady-State Behaviour of EKF-SLAM

This chapter presents an analysis of the EKF-SLAM problem from the point of view of control theory. A block diagram of the system under consideration is shown in Figure 3.1.

For any estimation algorithm it is important to know what limits the achievable state estimation quality. For the linear KF-SLAM algorithm, Dissanayake et al. in [40] showed that the map uncertainty reaches a lower bound determined by the initial uncertainty in the vehicle location. The nonlinear EKF-SLAM problem can exhibit two types of problems. As in any EKF estimator, the approximations due to linearisation of the system and measurement equations can lead to filter divergence [9]. And, in the specific case of a stationary vehicle equipped with a range-bearing sensor, it has been shown that the EKF algorithm can produce an inconsistent map [70].

Another undesirable filter property is non-zero steady state error. When building a map, and at the same time estimating the absolute vehicle location purely from local sensor measurements and odometry, the results of the EKF-based approach to SLAM will depend on the error produced at the very first iteration [104]. This dependency

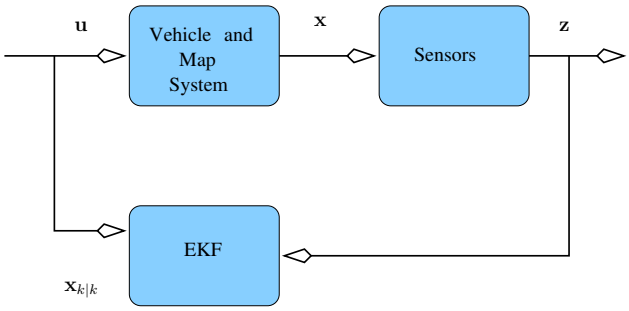


Figure 3.1: Block diagram of the closed loop system for estimation in SLAM process.

is produced by the partial observability inherent to absolute localisation (referred to as world-centric maps in [26]). The link between the finite steady-state error and partial observability was identified by Andrade-Cetto in [3]. This work showed that the SLAM problem with range and bearing measurements of point landmarks is partially observable and characterised the null space basis of the observability matrix both for linear and nonlinear vehicles.

In this chapter our aim is to analyse the steady state behaviour of the SLAM system in Figure 3.1. We show that partial system observability, in addition to the linearisation errors, affects the estimation performance in the SLAM problem, producing a marginally stable filter. We apply the same analysis to the SLAM system made fully observable by using the method proposed by [3]. We consider both linear and nonlinear vehicle and measurement models.

On the other hand, a controllability condition analysis is also performed to show limitations of the EKF, when used for tracking landmark states by considering them stationary and having only vehicle process noise as the input to the system. Controllability condition states that the process noise enters into each state component and prevents the Kalman gain and the covariance matrix from converging to zero. If this condition is achieved the steady-state covariance matrix converges to a unique positive definite matrix, independent of the initial covariance. This does not typically happen in the SLAM problem because landmarks are considered static, with no process noise

entering to the map state.

The rest of the chapter is organised as follows. In Section 3.1 the steady-state behaviour of the KF is illustrated with an example of a simple linear one-dimensional mobile robot, that we call *monobot*, performing the SLAM process. Then, we show the behaviour of the more realistic case of a planar mobile robot with range and bearing measurements. The aim is to show how partial observability hinders full reconstructibility of the state space, making the final map estimate dependent on the initial measurements. The stability analysis of the EKF-SLAM problem is performed from the point of view of control theory .

Section 3.2 is devoted to show the behaviour of the filter when the system becomes fully observable, an example for a linear case shows the estimator is asymptotically stable. A linear observability analysis for the planar mobile robot shows that an anchored system is fully observable. The behaviour of this anchored system is shown with an example using real data.

In Sections 3.1 and 3.2 it is also shown that, as a consequence of partial controllability, updates do not change the landmark state estimates after just a small number of iterations. This happens because the corresponding Kalman gain terms tend to zero.

Section 3.3 considers the case when artificial positive definite process noise is added to the map state. The result can be equivalent to decorrelation of the state error covariance matrix. This technique is commonly used in Kalman filtering to reduce computational complexity. In this section we show that it can also produce a controllable system. In this same section, we introduce the covariance decorrelation method which preserves stability of the filter and, at the same time, reduces the algorithm's complexity by avoiding decorrelation of the vehicle part of the state vector.

In Section 3.4 full observability is combined with full controllability. The result is a stable solution of the filter even though the covariance decorrelation method is used.

3.1 Effects of Partial Observability and Controllability

There are three important convergence properties for the EKF-SLAM algorithm according to [40]: (1) the determinant of any submatrix of the map covariance matrix decreases monotonically as observations are successively made; (2) in the limit as the number of observations increases, the landmark estimates become fully correlated; and (3) in the limit, the covariance associated with any single landmark location estimate reaches a lower bound determined only by the initial covariance in the vehicle location estimate at the time of the first sighting of the first landmark.

In a dynamic system where both state and measurements are corrupted by noise (a typical SLAM model), it is important to know whether it is possible or not to reconstruct the entire state space from output measurements, i.e. whether the system is *observable*. This is important because the asymptotic stability of state estimation is guaranteed only if the system is fully observable. Moreover, convergence of the state error covariance matrix to a finite steady state value is equivalent to filter stability in the sense of bounded-input bounded-output. Asymptotic stability of the filter does not require the dynamic system to be stable; only the observability condition is required, even though it alone does not guarantee uniqueness of the solution of the Riccati equation [9].

When a stochastic system is partially controllable, the Gaussian noise sources $\mathbf{v}_k$ do not affect all of the elements of the state space. That is, some states are uncorrupted by the noise. The diagonal elements of $\mathbf{P}$ corresponding to these incorruptible states will be driven to zero by the Kalman filter, and once this happens, these estimates will remain fixed and further observations will not alter their values.

The solution of the Riccati equation (2.18) for a time-invariant system with process noise converges asymptotically to a finite steady-state covariance if the pair $(\mathbf{F}, \mathbf{H})$ is completely observable and the pair $(\mathbf{F}, \mathbf{G}\mathbf{Q}^{1/2})$ is completely controllable.

In summary, the observability condition guarantees a steady flow of the information

about each state component, and prevents the uncertainty (state error covariance) from becoming unbounded. Moreover, the controllability condition allows the process noise to enter into each state component and prevents the covariance matrix from becoming singular [9].

Next, we derive the equations for the linear SLAM case, in order to show the behaviour of the filter. We are able to see in the end, that the estimated system has a pole on the unit circle of the z-plane, making the filter marginally stable. The state estimation error $\mathbf{e}_k$ is defined as

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}_{k|k} \,. \tag{3.1}$$

Then, with the appropriate substitutions using (2.12), (2.13) and the EKF (Appendix A.2) as the estimator we obtain the expression for the linearised discrete-time error dynamics,

$$\mathbf{e}_{k+1} = (\mathbf{F} - \mathbf{KHF})\mathbf{e}_k + (\mathbf{I} - \mathbf{KH})\,\mathbf{v}_k - \mathbf{K}\mathbf{w}_{k+1} \,, \tag{3.2}$$

where $\mathbf{K}$ is the Kalman filter gain. In order to guarantee stability of the filter, the close-loop transition matrix $(\mathbf{F} - \mathbf{KHF})$ must be stable. That is, its eigenvalues must lie inside the unit circle of the z-plane.

Unfortunately, the state vector formed by appending the vehicle position estimate with the landmark location estimates is not fully observable, which results in one eigenvalue of the matrix $(\mathbf{F} - \mathbf{KHF})$ being equal to one. Modifying appropriately the measurement model or the state-space, the estimated system becomes asymptotically stable, i.e., it has a zero mean steady state estimation error.

This means that the matrix $(\mathbf{F} - \mathbf{KHF})$ is marginally stable as a consequence of the partial observability. The pole equal to one corresponds to a state estimate signal corrupted with constant amplitude. However, the solution to the Riccati equation still converges, in the linear case, to a positive semi definite steady state covariance in terms

of $\mathbf{P}_{r,0|0}$, $\mathbf{V}$, $\mathbf{W}$, and the total number of landmarks n [54]. Notice however, that for the nonlinear case, the computation of the Jacobians $\mathbf{F}$ and $\mathbf{H}$ will in general also depend on the steady state value of $\mathbf{x}$.

On the other hand, the process model assumes that the landmarks are stationary, with no state transition function and no process noise. Therefore, their associated covariance (its determinant) will asymptotically approach zero as more observations are made [40]. The filter gain for the landmark states will also tend to zero.

Let us look at the controllability matrix for this system. Given that

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_r & \\ & \mathbf{I} \end{bmatrix}$$

and

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_r \\ \mathbf{0} \end{bmatrix},$$

the controllability matrix is

$$\mathcal{C} = \left[\; \mathbf{G}\mathbf{Q}^{1/2} \;\middle|\; \ldots \;\middle|\; \mathbf{F}^{\dim \mathbf{x}-1}\mathbf{G}\mathbf{Q}^{1/2} \; \right]. \tag{3.3}$$

Consequently, the dimensionality of the controllable subspace, spanned by the column space of $\mathcal{C}$ is $\operatorname{rank} \mathcal{C} = \dim \mathbf{x}_r$, regardless of the number of landmarks in the map. Obviously, the only controllable states are the ones associated with the vehicle motion.

A semi-definite covariance matrix means that the filter believes that it has perfect knowledge of some state components. For these states the Kalman filter gain evaluates to zero, and the innovations are not considered in the revision of the state estimate. To avoid this situation, a standard practice is to add a positive definite pseudo-noise covariance to those uncontrollable states [9].

The following examples demonstrate that in partially observable and partially controllable SLAM systems one of the eigenvalues of the close-loop estimation matrix

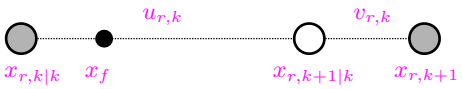


Figure 3.2: Monobot, one-dimensional mobile robot.

makes the system marginally stable and the Kalman gain is equal to zero.

Example 3.1

Consider a one-dimensional robot (monobot) with a one-dimensional landmark as illustrated in Figure 3.2. At time t_k, the vehicle location is $x_{r,k}$ and the motion command is u_k. The vehicle position error dynamics is modeled with the additive term v_k, and the discretised system model is simply

$$x_{r,k+1} = x_{r,k} + u_k + v_k \tag{3.4}$$

$$x_{f,k+1} = x_{f,k} \tag{3.5}$$

The map produced by this system is a single static landmark x_f. The measurement model for such landmark is

$$z_{k+1} = x_{f,k+1} - x_{r,k+1} + w_k\,, \tag{3.6}$$

where w_k is the landmark measurement error.

In the notation of (A.1.1) and (A.1.2), the matrices of this simple system are

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \mathbf{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad \mathbf{H} = \begin{bmatrix} -1 & 1 \end{bmatrix}\,. \tag{3.7}$$

For a Kalman gain $\mathbf{K} = [k_1, k_2]^\top$ in this simplest SLAM configuration, the eigenvalues of the matrix

$$\mathbf{F} - \mathbf{KHF} = \begin{bmatrix} k_1 + 1 & -k_1 \\ k_2 & -k_2 + 1 \end{bmatrix} \tag{3.8}$$

are $\{1, k_1 - k_2 + 1\}$. *Thus, one closed-loop eigenvalue is always equal to one regardless of the Kalman filter gain.*

As a special case we consider the steady state and find the closed-loop eigenvalues of the system. We assume that the initial monobot position variance is equal to zero and the landmark variance is equal to one. After solving the Riccati equation the steady-state covariance matrix $\mathbf{P}$ *can be written as*

$$\mathbf{P} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2/(1+\sigma_\omega^2) \end{bmatrix}, \tag{3.9}$$

assuming zero initial covariance. The variance associated with the monobot process noise is σ_v *and the variance associated with the sensor noise is* σ_ω. *The Kalman gain is computed according to (A.1.9)*

$$\mathbf{K} = \frac{1}{s^2} \begin{bmatrix} -\sigma_v^2 \\ \frac{\sigma_\omega^2}{1+\sigma_\omega^2} \end{bmatrix},$$

where s *is the innovation variance defined as*

$$s^2 = \sigma_v^2 + \sigma_w^2 + \frac{\sigma_\omega^2}{1+\sigma_\omega^2}. \tag{3.10}$$

Substituting the above result in equation (3.8),

$$\mathbf{F} - \mathbf{KHF} = \frac{1}{s^2} \begin{bmatrix} s^2 - \sigma_v^2 & \sigma_v^2 \\ \frac{\sigma_\omega^2}{1+\sigma_\omega^2} & \frac{s(1+\sigma_\omega^2)-\sigma_\omega^2}{1+\sigma_\omega^2} \end{bmatrix} \tag{3.11}$$

the eigenvalues of the filter dynamics matrix for a one landmark monobot are $\{1, \frac{1}{s^2}(\sigma_v^2 - \sigma_w^2)\}$. *Therefore, one of the two eigenvalues is always equal to one regardless of the values of the noise variances. Figure 3.3 shows the two eigenvalues as a function of* σ_v *and* σ_ω, *where there is always an eigenvalue equals to one. This means that the closed-loop error dynamics is always marginally stable, resulting in a finite steady-state error. The steady-state value of the estimation error depends on the initial conditions.*

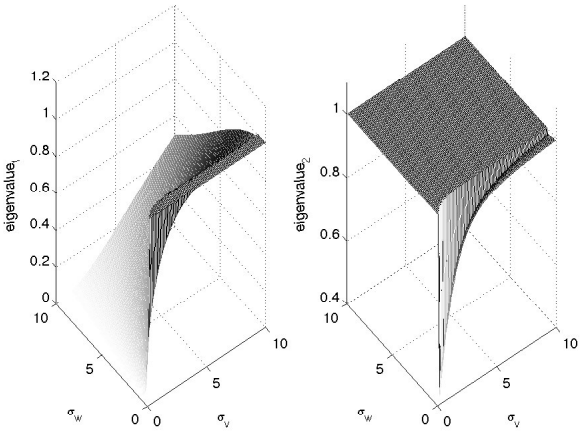


Figure 3.3: Eigenvalues of the one landmark monobot SLAM.

To further illustrate the effects of marginal filter stability in SLAM, a simulation for a vehicle under Brownian motion is shown. The case of one landmark is considered in the left column of Figure 3.4 and the case of two landmarks is shown in the right column of Figure 3.4. In both cases the monobot starts at location $x_{r,0} = -1$. *Both landmarks are located at* $x_f^{(i)} = 1$. *The plots shows: a) full state estimate, b) vehicle estimation error with* 2σ *uncertainty bounds, c) landmark estimation error, also with* 2σ *uncertainty bounds, and d) vehicle and landmark Kalman gains. The finite steady-state error for the vehicle and landmark locations can be seen in plots (b) and (c). This value of the steady-state error is smaller to the initial conditions when a large number of landmarks is used.*

The uncertainty in localising stationary landmarks with no process noise approaches zero. And so does the associated filter gain. Figure 3.4 show the evolution of the localisation errors for both the monobot and the landmarks, and the reduction to zero value of the landmark part of the Kalman gain, due to the uncontrollability of the system.

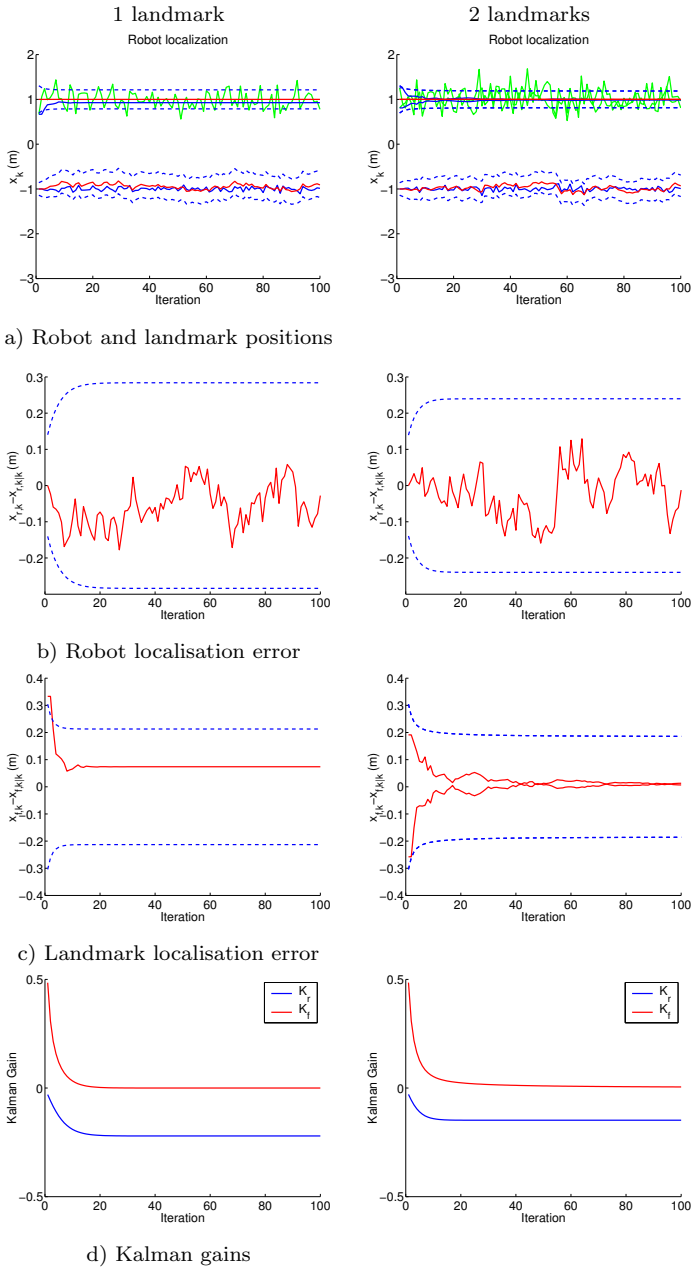


Figure 3.4: Partially observable SLAM for a monobot during Brownian motion with 100 iterations.(Partially observable - Partially controllable).

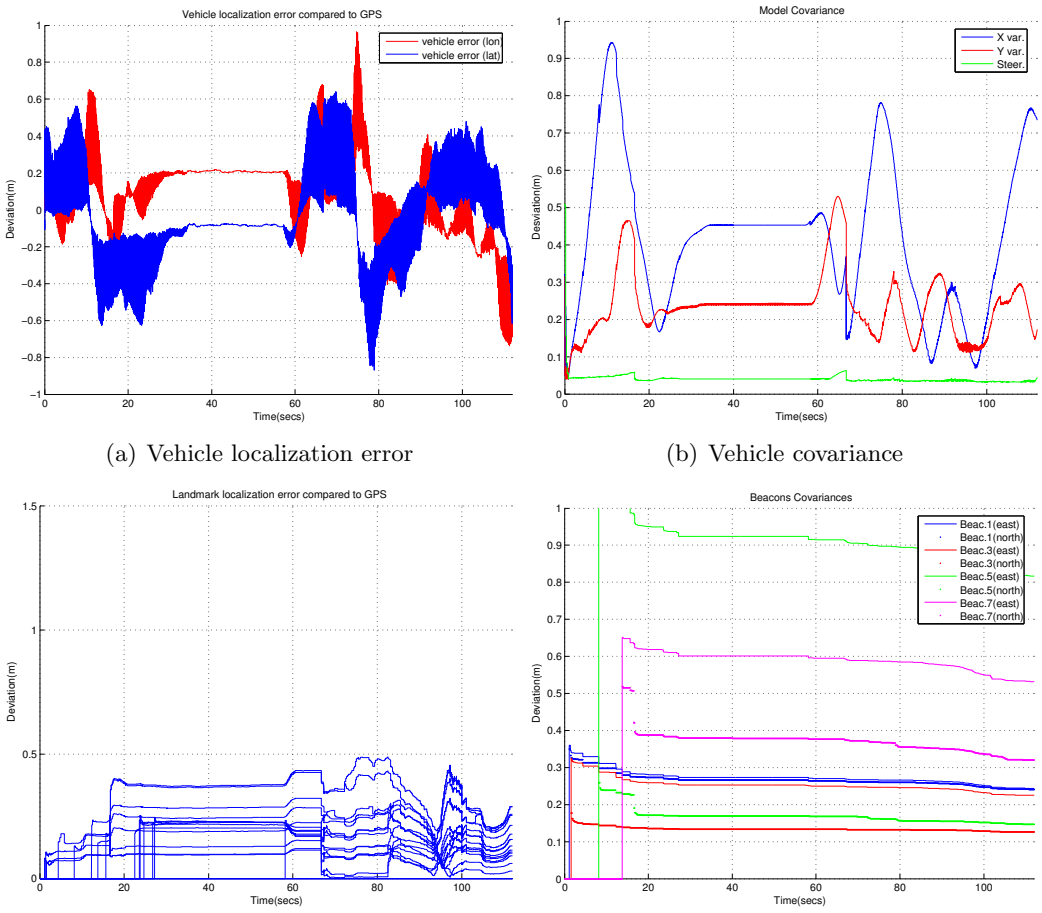
We showed how in the linear case of SLAM, the solution of the Riccati equation produces a marginally stable solution for the estimation error.

The results from previous example can be extended to more complicated motion vehicle models, for example, the nonlinear differential steer mobile robot model. For nonlinear cases the Extended Kalman Filter (EKF) is commonly used; the *a priori* estimates are evaluated as described in Section 2.2.5. Given that the estimator uses a linearisation of the system models, an observability and controllability analysis of the linearised system is appropriate. This analysis for a planar vehicle and range-bearing measurement models during SLAM shows that the system is partially observable [2]. Moreover, a nonlinear observability analysis using Lie algebra shows similar results [85]. For nonlinear models it is important to guarantee this behaviour since linearisation errors will always be present in the case of the EKF. In the next example we show the behaviour of the filter using real data.

Example 3.2

This example presents results of a simulation using an experimental dataset from the ACFR - University of Sydney database [101]. Nonlinear vehicle and measurement models are used. The dataset was obtained by driving a car through a car park. The landmarks used are tree trunks, as measured with a laser range finder. The reconstructed maps are compared to GPS ground truth to assess accuracy.

This first experiment corresponds to a typical partially observable partially controllable SLAM run showing the effects of marginal stability on the filter performance. Figure 3.6 plots results on this run, shows the actual vehicle path and landmark location estimates recovered by the algorithm, compared to GPS ground truth for the beacons. Figure 3.5 frames b) and d) show the covariances both for the vehicle and landmark state estimates. Note that even when the relative *map is internally consistent, it is slightly rotated and shifted from the actual beacon locations. The amount of this shift*



(a) Vehicle localization error

(b) Vehicle covariance

(c) Landmark localization errors

(d) Landmark covariance

Figure 3.5: Partially observable SLAM for a car-like vehicle using the University of Sydney Car Park dataset.

depends on the initial vehicle uncertainty, i.e., the initial filter conditions, and can be seen in Figure 3.5, frame c).

By performing the stability analysis, we have demonstrated that using the typical range and bearing measurement model for SLAM system in a world-centric representation, it is not possible to obtain a zero mean state error estimate. Instead, one obtains a constant bounded state estimate as in the linear case. The reason being, that the filter

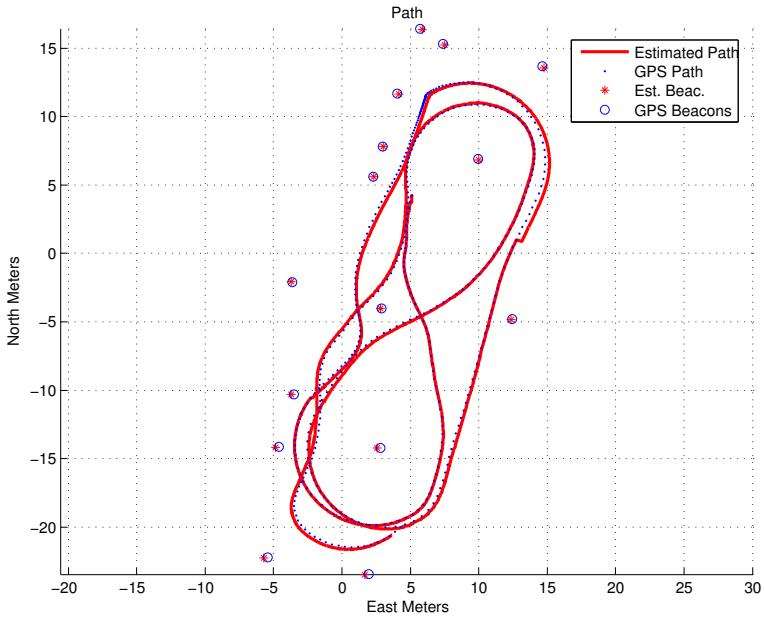


Figure 3.6: Partially observable SLAM run. Vehicle path and landmark location estimates, compared to GPS ground truth.

used, one in which the absolute vehicle and landmark position estimates stacked in the same state vector, is marginally stable. However, a perfect relative map is produced by the algorithm [40].

Changing the model it is possible to make the system observable and in such way guaranteeing the stability of the filter as we will show in the following section.

3.2 Fully Observable System

Modifying appropriately the measurement model or the state-space, the estimated system becomes asymptotically stable, i.e., it has a zero mean steady state estimation error.

Full observability can be achieved by using several methods proposed in the past [2], [11] and [76]: 1) fixed global references; *anchors* or *markers*, 2) the use of an external sensor or 3) the use of the relative position of the landmark with respect to the position

of the vehicle instead of global positioning [26].

The effect of full observability on the steady-state behaviour of SLAM system is demonstrated using the *anchoring* method [2]. In the first example we analyse the case of monobot SLAM with linear KF. Next, we present an example that shows the full observability of an anchored nonlinear system. The third example shows a simulation using the experimental vehicle data.

Example 3.3

We continue the Example 3.1, but now with an anchor added at the origin. The measurement model becomes

$$\begin{bmatrix} z_k^{(0)} \\ z_k \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} w_k^{(0)} \\ w_k \end{bmatrix} \tag{3.12}$$

No map state is needed for the anchor. The zero-th superscript in the measurement vector is used for the consistent indexing of landmarks and measurements with respect to the original model. It can be easily shown that the observability matrix for this augmented measurement model is full rank [2].

The stability of the matrix $(\mathbf{F} - \mathbf{KHF})$ *depends on the values of the Kalman gain*

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}. \tag{3.13}$$

As shown next, the eigenvalues of this matrix lie inside the unit circle of the z-plane.

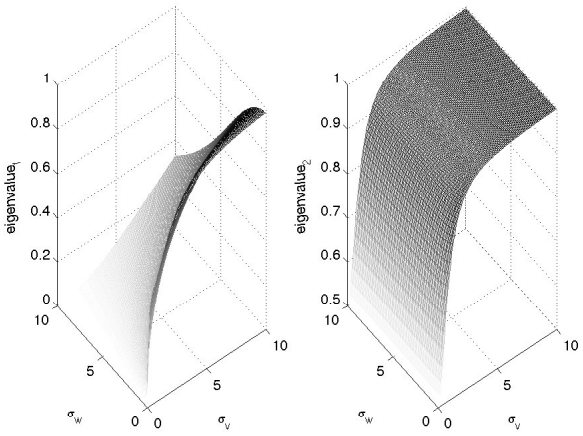


Figure 3.7: Eigenvalues of the observable one landmark monobot SLAM.

Substituting for the fully observable measurement model we find that,

$$\begin{aligned} k_{11} &= -\frac{\sigma_\omega^4 + \sigma_v^2\sigma_\omega^4 + 4\sigma_v\sigma_\omega^2 + 2\sigma_\omega^2 + 3\sigma_v^2}{\gamma} \\ k_{12} &= -\frac{\sigma_\omega^4 + \sigma_v^2 + 3\sigma_v^2\sigma_\omega^2 + \sigma_v^2\sigma_\omega^4}{\gamma} \\ k_{21} &= -\frac{2\sigma_\omega^2 + \sigma_v^2\sigma_\omega^2 + 2\sigma_v^2}{\gamma} \\ k_{22} &= \frac{\sigma_\omega^4 + 2\sigma_\omega^2 + \sigma_v^2\sigma_\omega^2 + 2\sigma_v^2}{\gamma} \end{aligned} \tag{3.14}$$

where

$$\gamma = \sigma_\omega^6 + 2\sigma_v^2\sigma_\omega^4 + 6\sigma_\omega^4 + 7\sigma_v^2\sigma_\omega^2 + 4\sigma_\omega^2 + 4\sigma_v^2 \tag{3.15}$$

The eigenvalues of the matrix $(\mathbf{F} - \mathbf{KHF})$ *are shown in Figure 3.7 as a function of the variances of the process and the measurement noises. The values never reach the limits of zero or one, therefore the system is fully observable.*

In general terms, this shows how, in the fully observable case, the observability matrix has full rank, for any values of $\mathbf{Q}$ *and* $\mathbf{R}$*, the eigenvalues of* $(\mathbf{F} - \mathbf{KHF})$ *will always be inside the unit circle of the z-plane.*

The effects of marginal stability discussed in Section 3.1 are eliminated when observability is guaranteed. Some simulation results in which partial observability has been revised are presented in Figure 3.8 under the same conditions as in Example 3.1. The vehicle and landmark estimation errors do converge to a zero mean signal (plots (b) and (c)) and we observe a reduction in vehicle and landmark variance compared to those in Figure 3.4.

The results of the previous example can be extended to more complicated models. In the next example we develop the equations of the nonlinear planar vehicle and range a bearing measurement model using anchors, to show how full osbervability is achieved for a SLAM system.

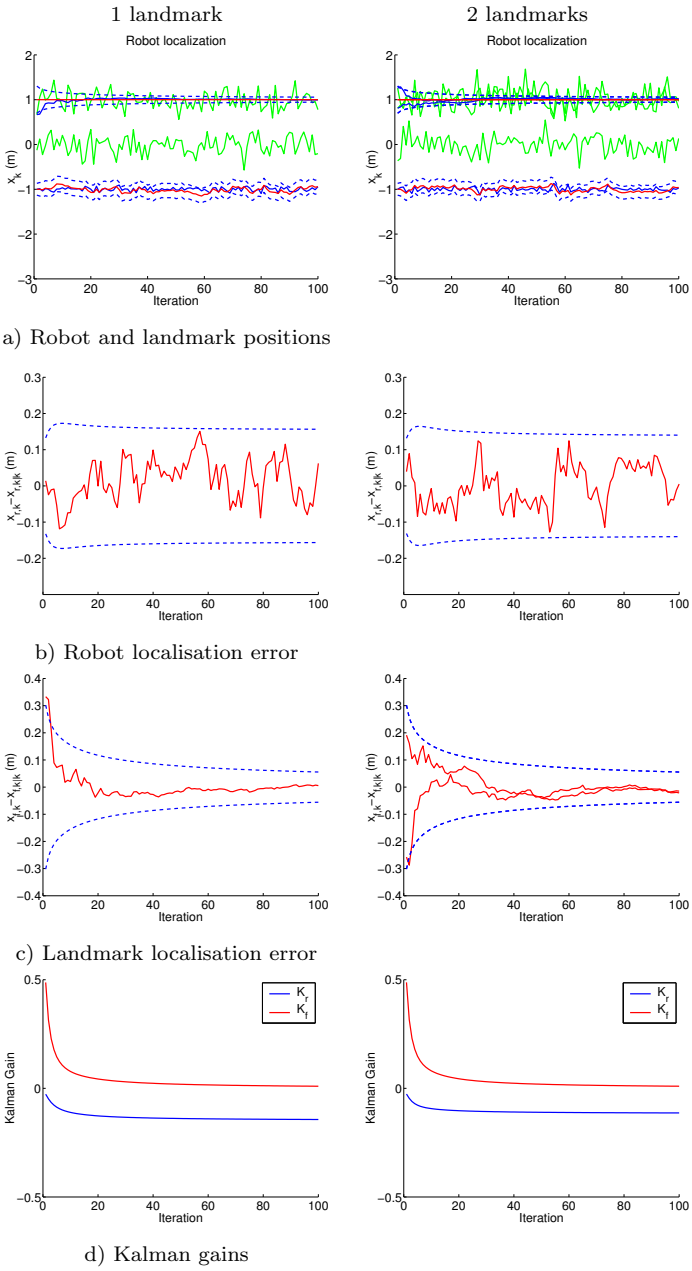
Example 3.4

Let us consider the SLAM system of the planar vehicle of the Figure 3.9 mapping two-dimensional landmarks $\mathbf{x}_f = (x_f^i, y_f^i)^\top$*.*

The planar vehicle is modelled as a wheeled mobile robot with differential steering. Its state space is three-dimensional: two Cartesian coordinates and the angular orientation. The vehicle is controlled by a linear velocity v *and a angular velocity* ω*. The process model used to predict the trajectory of the centre of projection of the sensor, considering input noises, is given by,*

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + ((v_k + v_{v,k})\cos\theta_k - l(\omega_k + v_{\omega,k})\sin\theta_k)\,\Delta t \\ y_k + ((v_k + v_{v,k})\sin\theta_k + l(\omega_k + v_{\omega,k})\cos\theta_k)\,\Delta t \\ \theta_k + \Delta t(\omega_k + v_{\omega,k}) \end{bmatrix} \tag{3.16}$$

where l *is the distance from the centre of the wheel axle to the location of the centre*



a) Robot and landmark positions

b) Robot localisation error

c) Landmark localisation error

d) Kalman gains

Figure 3.8: Fully observable SLAM for a monobot during Brownian motion with 100 iterations.

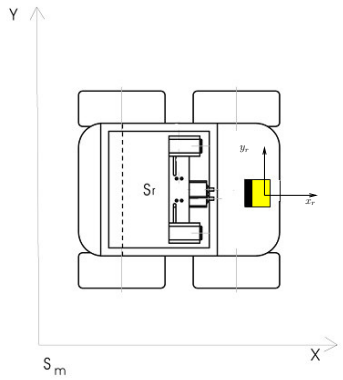


Figure 3.9: Differential steering nonlinear vehicle. The Cartesian coordinates of the control point (x_r, y_r) are located on the base of the laser range scanner, and were chosen not coincident with the vehicle axle centre.

of projection for any given sensor[1]*,* Δt *is discrete time step, and* v_v, v_ω *are zero mean Gaussian input noises.*

The Jacobian matrices $\mathbf{F}_r = \frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}$ *and* $\mathbf{G}_r = \frac{\partial f(\mathbf{x},\mathbf{u})}{\partial \mathbf{v}}$ *are obtained by differentiating (3.16) with respect to states and noises. That is,*

$$\mathbf{F}_r = \begin{bmatrix} 1 & 0 & -(v_k \sin\theta_k - l\omega_k \cos\theta_k)\Delta t \\ 0 & 1 & (v_k \cos\theta_k - l\omega_k \sin\theta_k)\Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

and

$$\mathbf{G}_r = \begin{bmatrix} \cos\theta_k \Delta t & -l \sin\theta_k \Delta t \\ \sin\theta_k \Delta t & +l \cos\theta_k \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.18}$$

The laser range scanner is considered as a typical sensor producing range and bear-

[1] We will also call it *control point*, because in further chapters it will be use for the controller.

ing observations. The measurement model is given by

$$\begin{bmatrix} z_{r,k} \\ z_{\beta,k} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_f^i - x_k)^2 + (y_f^i - y_k)^2} + w_{r,k} \\ \tan^{-1}\left(\frac{(y_f^i - y_k)}{(x_f^i - x_k)}\right) - \theta_k + \frac{\pi}{2} + w_{\beta,k} \end{bmatrix} \tag{3.19}$$

with z_r and z_β the range and bearing of an observed point landmark with respect to the laser centre of projection. The absolute coordinates of such landmark are x_f^i and y_f^i and the zero mean Gaussian measurement noises are w_r and w_β.

The Jacobian matrix for this model is,

$$\mathbf{H}_i = \begin{bmatrix} -\frac{x_f^i - x_k}{d_i} & -\frac{y_f^i - y_k}{d_i} & 0 & \dots & \frac{x_f^i - x_k}{d_i} & \frac{y_f^i - y_k}{d_i} & \dots \\ \frac{y_f^i - y_k}{d_i^2} & -\frac{x_f^i - x_k}{d_i^2} & -1 & & -\frac{y_f^i - y_k}{d_i^2} & \frac{x_f^i - x_k}{d_i^2} & \end{bmatrix} \tag{3.20}$$

with $d_i = \sqrt{(x_f^i - x_k)^2 + (y_f^i - y_k)^2}$.

The results from the previous example are easily extensible to this new vehicle model. Thus, the measurement model of a global reference fixed at the origin, for the nonlinear vehicle is

$$\mathbf{h}^{(0)} = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} + w_{r,k} \\ \tan^{-1}\left(\frac{y_k}{x_k}\right) - \theta_k + \frac{\pi}{2} + w_{\beta,k} \end{bmatrix} \tag{3.21}$$

and its corresponding Jacobian is

$$\mathbf{H}_0 = \begin{bmatrix} \frac{x_k}{\sqrt{x_k^2+y_k^2}} & \frac{y_k}{\sqrt{x_k^2+y_k^2}} & 0 & 0 & 0 & \dots \\ -\frac{y_k}{x_k^2+y_k^2} & \frac{x_k}{x_k^2+y_k^2} & -1 & 0 & 0 & \end{bmatrix} \tag{3.22}$$

The new measurement model results,

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_i \end{bmatrix} \tag{3.23}$$

and it produces an observability matrix[2]*,*

[2]For more details about the observability matrix please refer to Chapter 4.

$$\mathcal{Q} = \begin{bmatrix} \mathbf{H} \\ \mathbf{HF} \\ \mathbf{HF}^2 \\ \mathbf{HF}^3 \\ \mathbf{HF}^4 \end{bmatrix} \tag{3.24}$$

$$= \begin{bmatrix} \frac{x_k}{\sqrt{x_k^2+y_k^2}} & \frac{y_k}{\sqrt{x_k^2+y_k^2}} & 0 & 0 & 0 \\ -\frac{y_k}{x_k^2+y_k^2} & \frac{x_k}{x_k^2+y_k^2} & -1 & 0 & 0 \\ -\frac{x_f-x_k}{d} & -\frac{y_f-y_k}{d} & 0 & \frac{x_f-x_k}{d} & \frac{y_f-y_k}{d} \\ \frac{y_f-y_k}{d^2} & -\frac{x_f-x_k}{d^2} & -1 & -\frac{y_f-y_k}{d^2} & \frac{x_f-x_k}{d^2} \\ 0 & 0 & -\frac{(x_f-x_k)v_k\sin\theta_k+(y_f-y_k)v_k\cos\theta_k}{d} & 0 & 0 \\ 0 & 0 & -\frac{(x_f-x_k)v_k\cos\theta_k-(y_f-y_k)v_k\sin\theta_k}{d^2} & 0 & 0 \\ 0 & 0 & \frac{x_k v_k\sin\theta_k+y_k v_k\cos\theta_k}{x_k^2+y_k^2} & 0 & 0 \\ 0 & 0 & \frac{x_k v_k\cos\theta_k+y_k v_k\sin\theta_k}{x_k^2+y_k^2} & 0 & 0 \end{bmatrix} \tag{3.25}$$

which can be verified to be full-rank. Notice that for simplicity $l = 0$, but even with $l \neq 0$ the full-rank condition does not change. That is, for the linearised nonholonomic velocity-controlled planar mobile robot, simultaneous measurement of one anchor as global reference, and the estimation of the position of any other landmark, full observability of the SLAM system is guaranteed.

For this particular vehicle model it is shown in [3] that visibility of a two-dimensional anchor, guarantees full observability. This example showed that at least a two-dimensional anchor with orientation is needed or, if anchors without orientation are used, two 2D anchors can be used instead [85].

Steady state zero mean error is guaranteed for the SLAM algorithm by the certain knowledge of some landmarks, for example we assume the first two landmarks are anchored and the vehicle is localised with respect to them. To show this, in the next

example we present a nonlinear fully observable SLAM simulation using experimental data.

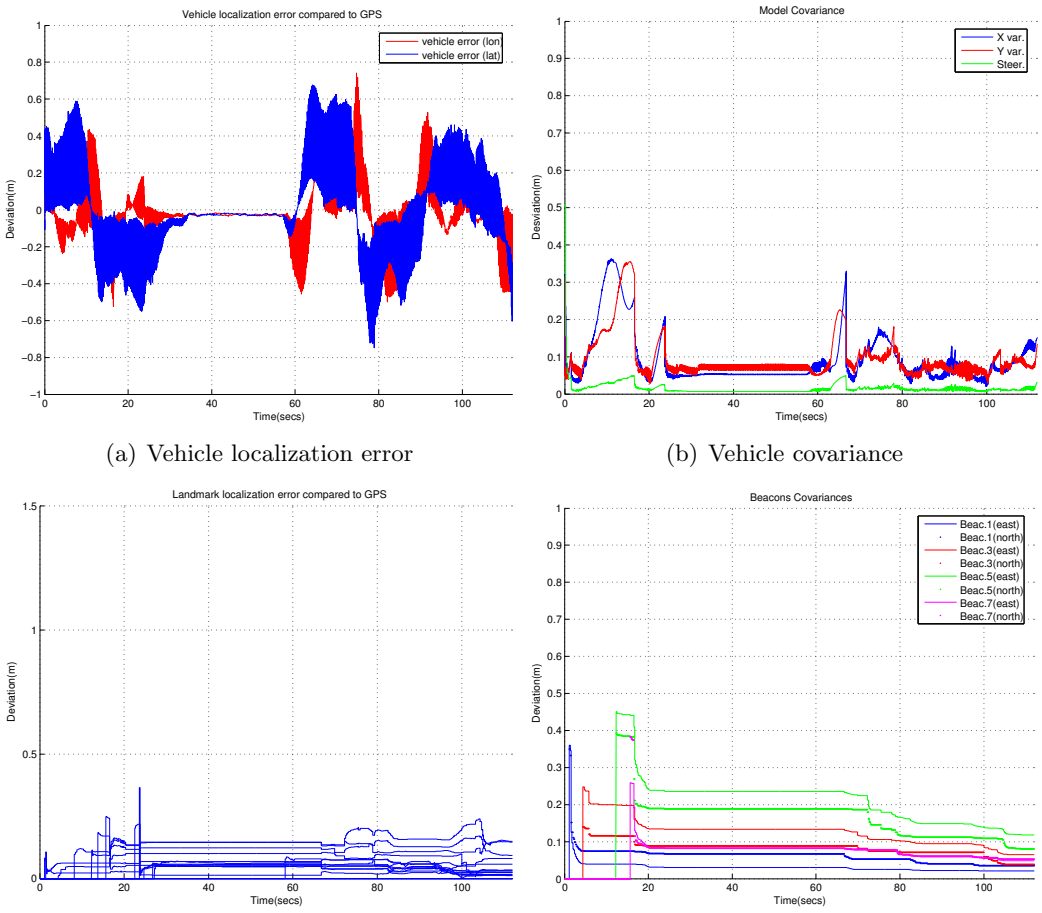
Example 3.5

The second experiment corresponds to a completely observable SLAM run (using the GPS measures for two anchors located at $(2.8953, -4.0353)$ *and* $(9.9489, 6.9239)$ *). In this case, the vehicle and landmark covariance estimates do not depend on the initial filter conditions, and thus are significantly reduced. This is shown in frames b) and d) in Figure 3.10. The absolute landmark error estimate is also significantly reduced, as shown in frame c). Figure 3.11 shows the actual vehicle path and landmark estimates as recovered by the filter. The beacons, shown as empty blue circles in the centre of the plot, are used as anchors for the map, and no state estimate is computed for them.*

3.3 Fully Controllable System

To speed up the performance of the algorithm, some authors have proposed the use of covariance inflation methods for the decorrelation of the state error covariance [58], subject to suboptimality of the filter. Adding pseudo-noise covariance to the landmark states is equivalent to making the system controllable. However, full decorrelation of a partially observable system might lead to filter instability [69]. In this section we want to show how to diagonalise only part of the state error covariance to obtain a suboptimal filter that is controllable, linear in time, and stable at the same time.

A clever way to add pseudo-noise to the model is by diagonalising the state error covariance matrix [57, 58, 69]. The result is a suboptimal filter that will compute inflated estimates for the vehicle and landmark covariances, that has the computational advantage of being uncorrelated. The addition of a covariance term $\mathbf{\Delta P}$ to the a priori



(a) Vehicle localization error

(b) Vehicle covariance

(c) Landmark localization errors

(d) Landmark covariance

Figure 3.10: Fully observable SLAM for a car-like vehicle using the University of Sydney Car Park dataset.

state covariance estimate

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top + \Delta\mathbf{P} \tag{3.26}$$

is equivalent to providing a new form for the plant noise Jacobian $\mathbf{G}' = \begin{bmatrix} \mathbf{G} & \mathbf{I} \end{bmatrix}$

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^\top + \mathbf{G}' \begin{bmatrix} \mathbf{Q} & \\ & \Delta\mathbf{P} \end{bmatrix} \mathbf{G}'^\top \tag{3.27}$$

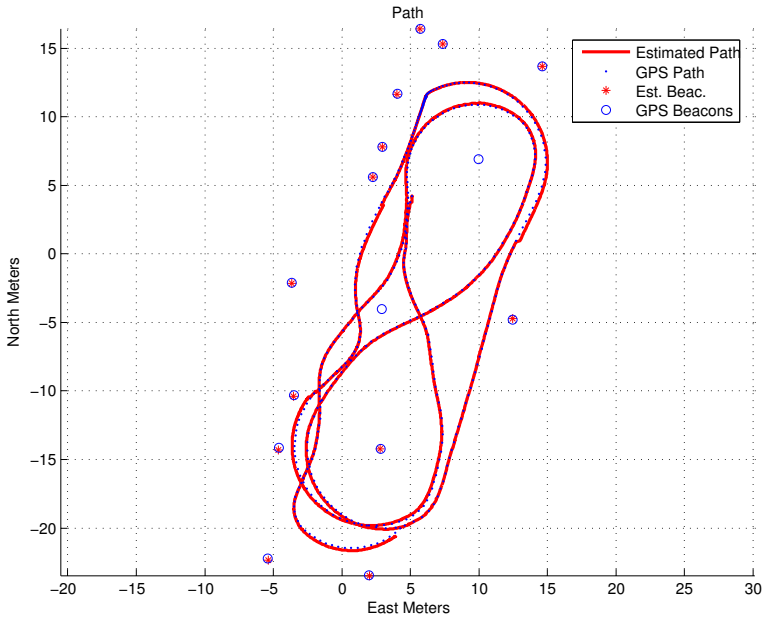


Figure 3.11: Fully observable SLAM run. Vehicle path and landmark location estimates, compared to GPS ground truth.

$\Delta\mathbf{P}$ may be chosen, for example, such as to minimise the trace of a resulting block diagonal $\mathbf{P}$ in (3.26) (see [69]).

Example 3.6

Continuing with Example 3.1, for the linear monobot case, choosing a full rank $\Delta\mathbf{P}$ *is equivalent to having noise input to more states than those that can be observed with the filter. In this case, because of partial observability, both vehicle and landmark variance estimates become unbounded. Figure 3.12 shows this for the same monobot as in Example 3.1. This phenomena was first observed in [69] using relative maps.*

Not only both the vehicle and landmark state estimation variances become unbounded. The addition of pseudo-noise should be performed only at most, in the amount of states equal to the dimension of the observable subspace. Then, the linear system is controllable but unstable because the pseudo noise was added to more states than the

uncontrollable ones, i.e. the system is fully decorrelated.

The following example presents a real SLAM system that is fully decorralated.

Example 3.7

Using the same database of Example 3.2, we present a third experiment that corresponds to a typical partially observable SLAM run, in which the entire state error covariance is being decorrelated as discussed in Section 3.3. Figure 3.13 plots results of this run, showing in frames b) and d) unbounded covariances both for the vehicle and landmark state estimates, due to the naïve covariance inflation method used.

It is clear that the vehicle position is highly correlated with every landmark and that the inflation produced by these terms is too large to maintain the covariance bounded producing data association errors (see Figure 3.14).

One solution to the problem of instability during covariance inflation, is to decorrelate only the landmark state estimates, and to preserve all vehicle to landmark correlations.

$$\Delta\mathbf{P} = \begin{bmatrix} \mathbf{0} & \\ & \mathbf{Q}_f \end{bmatrix} \tag{3.28}$$

such that $\mathbf{P}_f + \mathbf{Q}_f$, the map part of the state error covariance, is block diagonal.

Example 3.8

Figure 3.15 shows a partially observable monobot (Example 3.1) under Brownian motion for which only the map part of the state error covariance matrix has been decorrelated. The algorithm does converge to a steady state solution under these circumstances, and still can be implemented in real time. The one landmark case is identical to the original case, since the linear one landmark map is already diagonal (scalar actually).

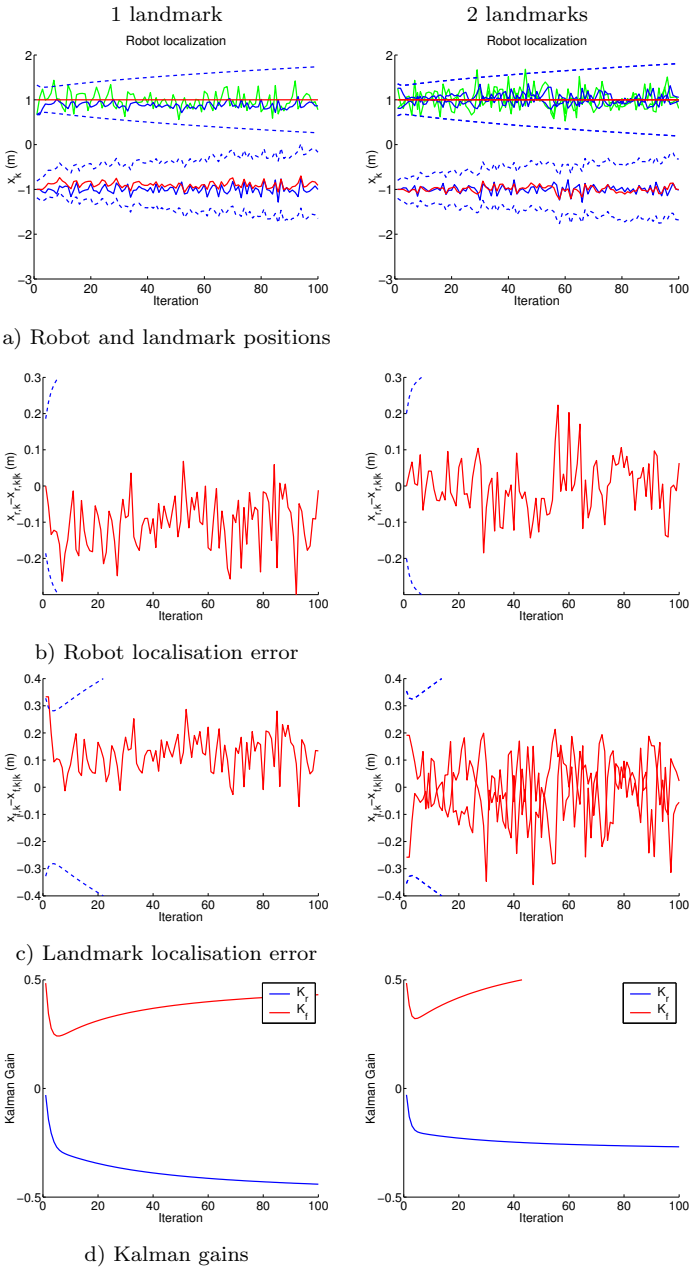
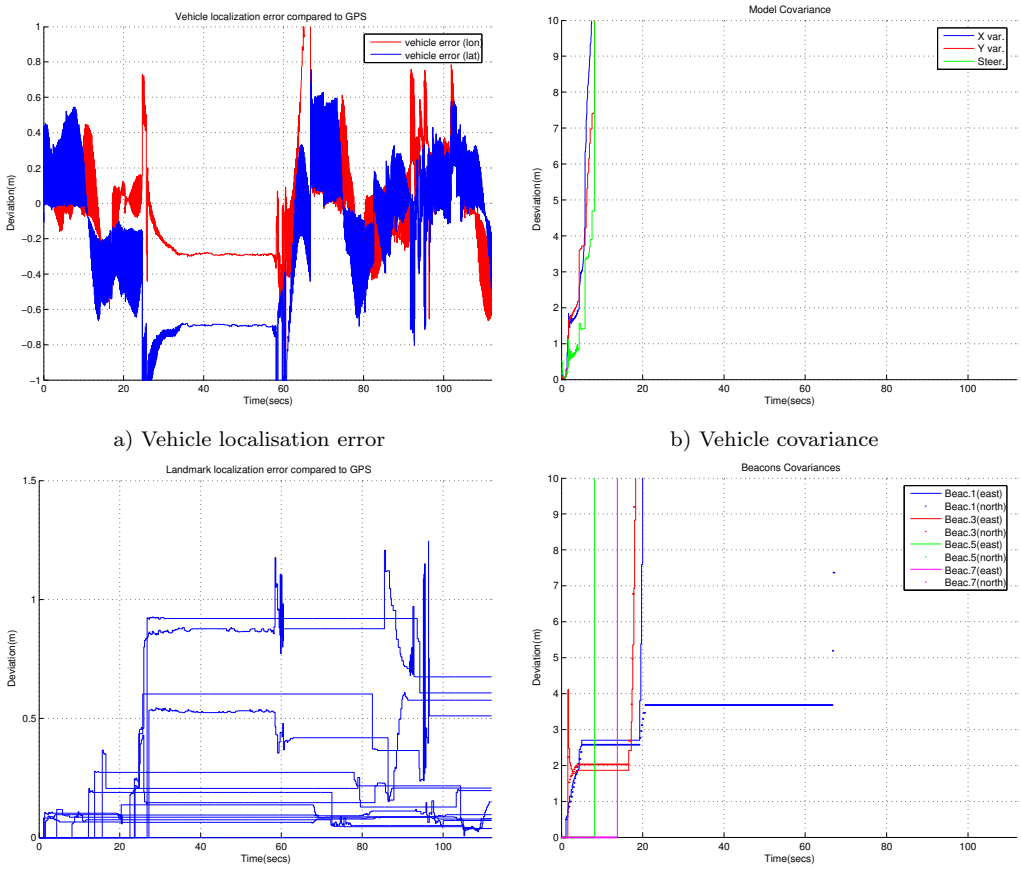


Figure 3.12: Partially observable SLAM for a Brownian motion monobot with 100 iterations. The entire state error covariance is decorrelated with the minimal trace solution [69]. By decorrelating the entire state error covariance matrix, the covariance estimates become unbounded.



a) Vehicle localisation error

b) Vehicle covariance

c) Landmark localisation errors

d) Landmark covariance

Figure 3.13: Partially observable SLAM for a car-like vehicle using the University of Sydney Car Park dataset. The entire state error covariance matrix is decorrelated with the minimal trace solution [69].

For the two-landmarks case, the landmark variance estimate is greater than the non-decorrelated shown in the third row in Figure 3.4 c). That is, the covariance has been inflated during decorrelation. Furthermore, now that the system is controllable, the Kalman gains for the landmark state estimates do not become zero, and they converge to a steady state value.

We can see in all cases, that the covariance inflation suboptimal partially observable

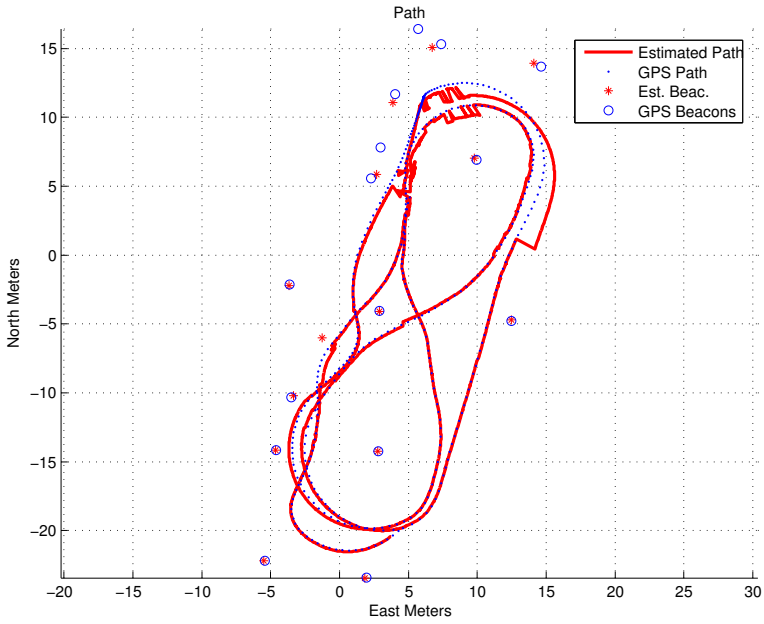


Figure 3.14: Vehicle path and landmark location estimates, compared to GPS ground truth for a partially observable SLAM run decorrelating only the map part of the state error covariance matrix.

SLAM converges only when

$$\text{rank } \Delta\mathbf{P} \leq \text{rank } \mathcal{Q}\,. \tag{3.29}$$

The next example presents the fully controllable version of the partially observable SLAM system using the ACFR database.

Example 3.9

A fourth experiment has been performed using the same data in Example 3.2. A partially observable SLAM system decorrelating the map part of the state error covariance. Adding pseudo-noise to the landmark states during the covariance inflation procedure amounts to making the system controllable; and doing so for as many states as those observable, produces both vehicle and landmark bounded state covariances estimates. This is shown in Figure 3.16, frames b) and d). Figure 3.17 shows the actual vehicle path and landmark location estimates recovered by the algorithm, compared to GPS ground truth for the beacons. There are some data association problems due to the

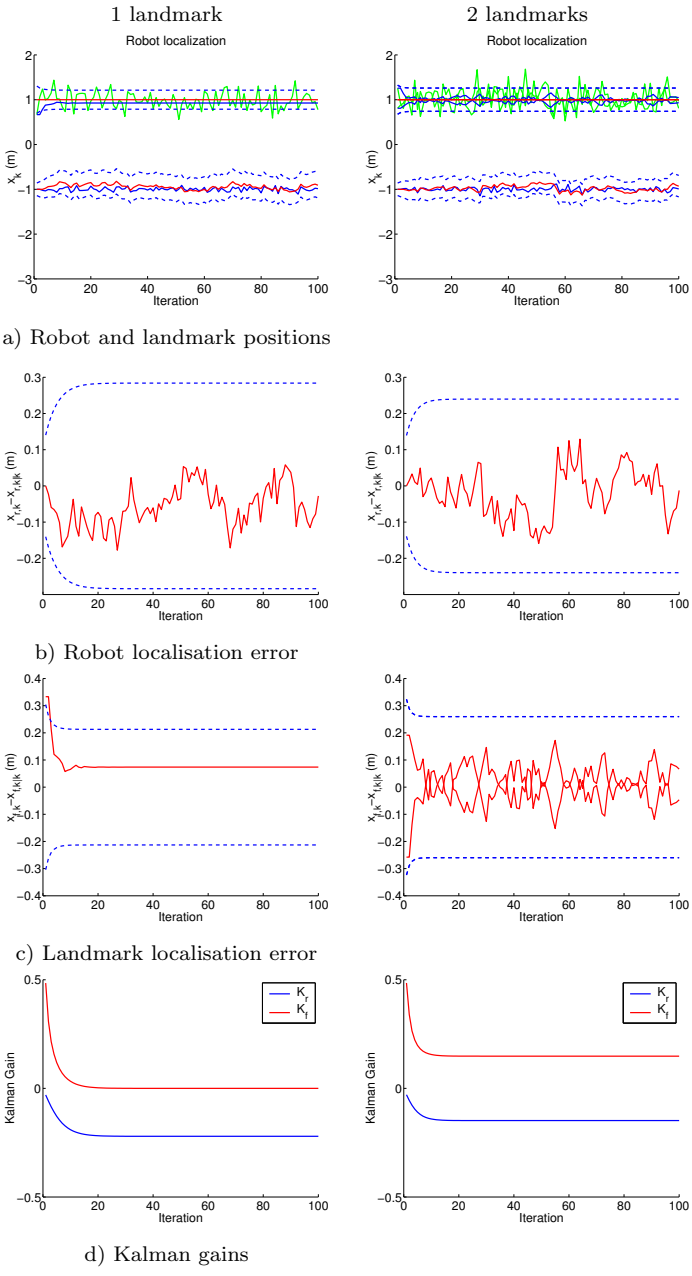
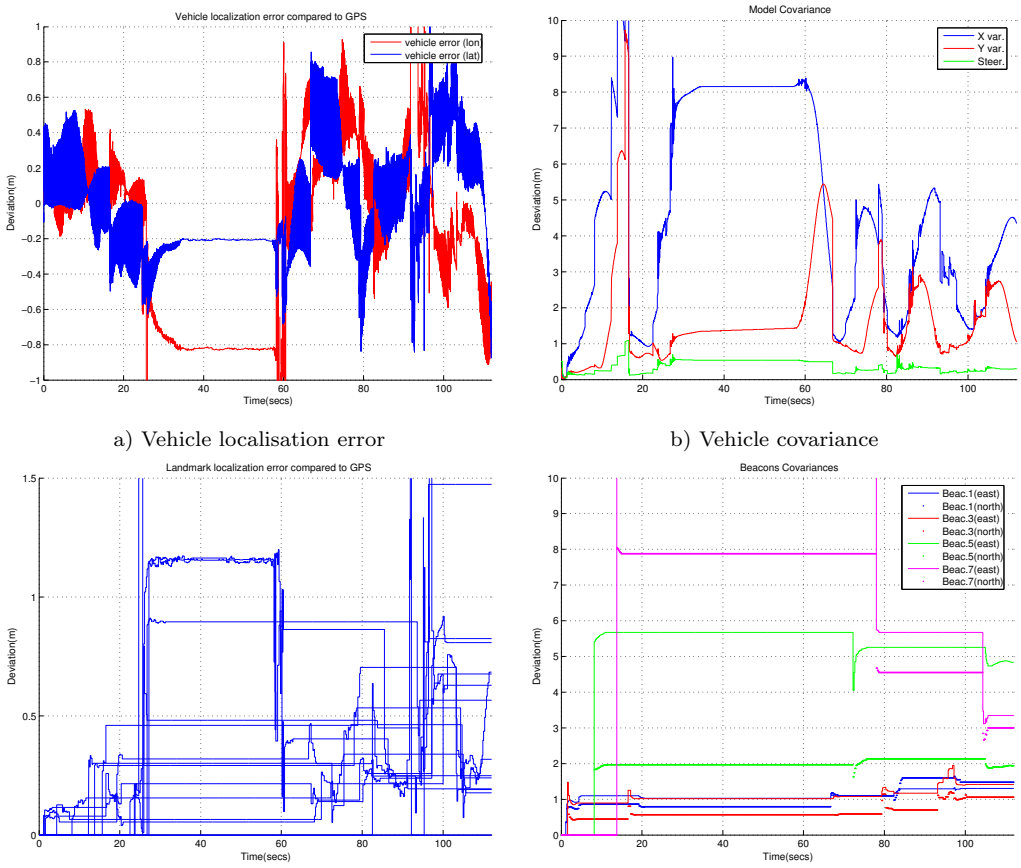


Figure 3.15: Partially observable SLAM for a Brownian motion monobot with 100 iterations. The state error covariance is decorrelated only for the landmark part of the state vector, with the minimal trace solution. By decorrelating only the map part of the state error covariance matrix, we preserve filter stability.



a) Vehicle localisation error

b) Vehicle covariance

c) Landmark localisation errors

d) Landmark covariance

Figure 3.16: Partially observable SLAM for a car-like vehicle using the University of Sydney Car Park dataset. Only the map part of the state error covariance matrix is decorrelated with the minimal trace solution. By adding controllability to as many states as those that are observable, the filter remains stable, and the estimated covariances remain bounded.

large covariances, with a joint compatibility test [102]. Ssome of these misassociations could be avoided, but in this case we wanted to show that landmark association is more complicated because of the decorrelation process.

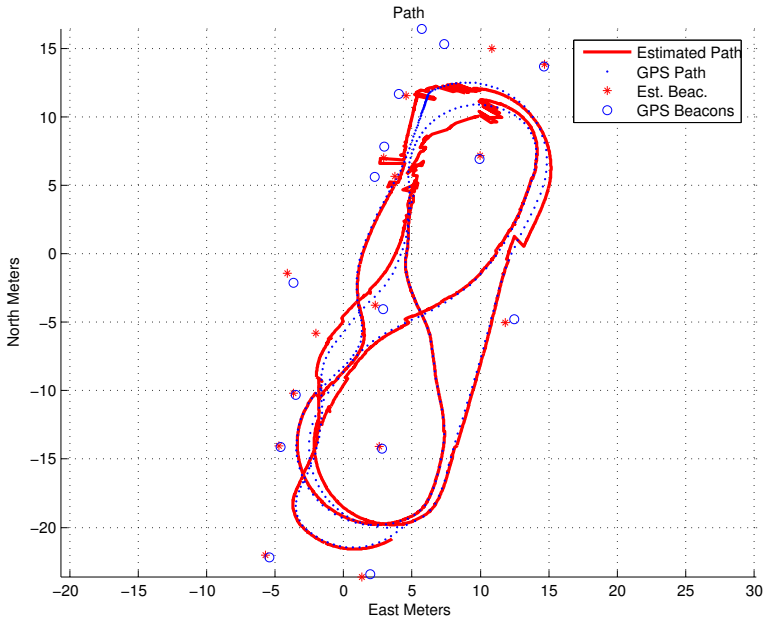


Figure 3.17: Partially observable and fully contrallable SLAM run. Vehicle path and landmark location estimates, compared to GPS ground truth with decorrelation of only the map part of the state error covariance matrix.

3.4 Fully Controllable - Fully Observable System

Considering the fully observable case [2], even if we add pseudo-noise to the vehicle as well as to the landmark states, the covariance will reach a steady-state value, and the Kalman gain will not be zero, at least, in the linear case.

Example 3.10

Continuing with Example 3.3, diagonalising the map state error covariance $\mathbf{P}$ *of the monobot with observable measurements, the state error variances reach lower values than those in the partially observable case and the Kalman gain never reaches the value of zero as shown in Figure 3.18. The solution of the Riccati equation is now independent of the initial covariance estimate* $\mathbf{P}_{0|0}$.

We have observed experimentally however, that with linear vehicle models, it is best

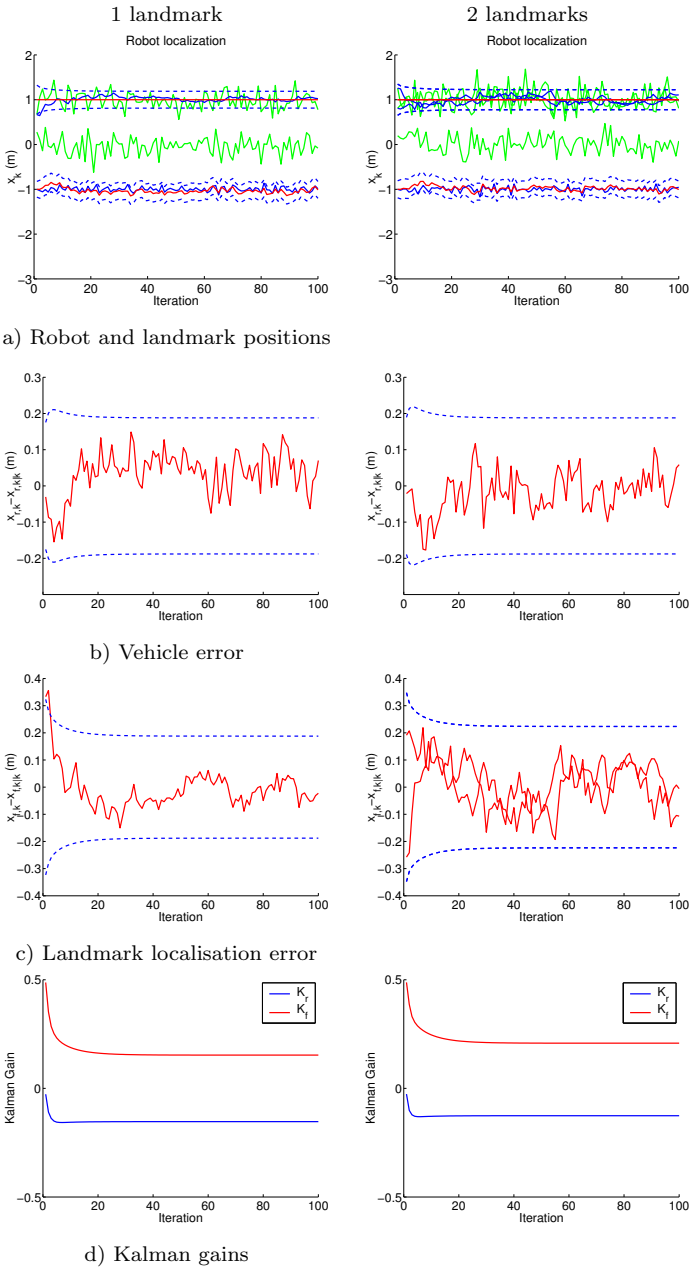


Figure 3.18: Fully observable SLAM for a Brownian motion monobot with 100 iterations. The entire state error covariance is decorrelated with the minimal trace solution. In the linear case, it is possible to decorrelate the entire state error covariance matrix, and still preserve filter stability.

to also decorrelate only the map part of the state error covariance, even in the fully observable case.

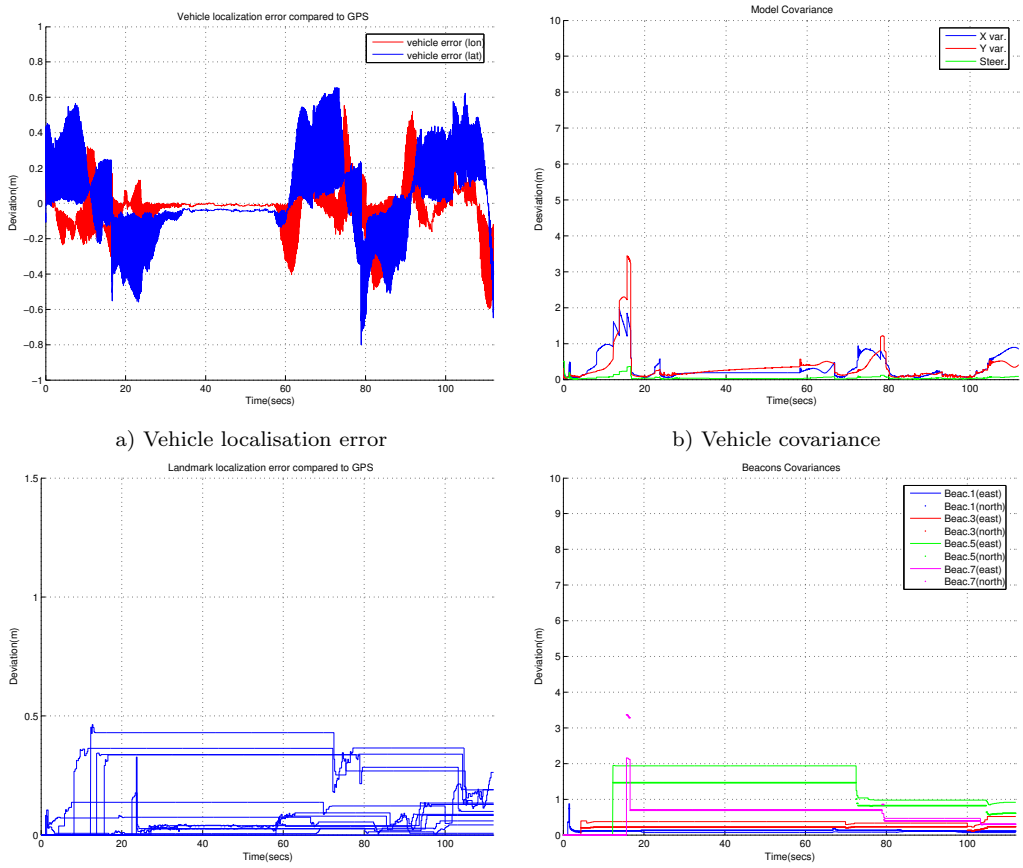
In the next example, nonlinear models with real data are used to show the behaviour of the filter for this fully observable and fully controllable case.

Example 3.11

The last experiment shown, using again the ACFR database of Example 3.2, corresponds to a fully observable SLAM run using the same anchors as before, and also decorrelating only the map part of the state error covariance. In this case, the vehicle and landmark covariance estimates do not depend on the initial filter conditions, and thus are significantly reduced. This is shown in frames b) and d) in Figure 3.19. The absolute landmark estimate error is also significantly reduced, as shown in Figure 3.19, frame c). Figure 3.20 was obtained with a suboptimal linear-time SLAM algorithm that has both bounded covariance estimates, and independence on the filter initial conditions; thus producing a fast and accurate absolute map.

3.5 Conclusion

We have shown that the typical partially observable approach to SLAM generates a unit norm eigenvalue for the matrix $\mathbf{F}-\mathbf{KHF}$, making the state estimation error converge to a non-zero-mean constant bounded value in the linear case. Marginal stability of such partially observable system produces also at least one *psd* solution to the steady state Riccati equation for the covariance error, provided the initial conditions of $\mathbf{P}$ are also *psd*. Partial observability makes the final map dependant on the initial observations. This situation can easily be remedied either by anchoring the map to the first landmark



a) Vehicle localisation error

b) Vehicle covariance

c) Landmark localisation errors

d) Landmark covariance

Figure 3.19: Fully observable SLAM for a car-like vehicle using the University of Sydney Car Park dataset. Only the map part of the state error covariance is decorrelated with the minimal trace solution. Full observability guarantees independence of the filter initial conditions, and an accurate absolute map is obtained, with smaller covariance estimates than its relative counterpart.

observed, by having an external sensor that sees the vehicle at all times, or by using a relative representation of the landmark with respect to the vehicle position that we will refer later in this work, as *sensor-centric* mapping.

In addition, landmarks in the map are assumed to be static. Therefore, the covariance yielded by the EKF will, asymptotically, tend to zero, and consequently the filter gain for these components will tend to zero. This is because there is no process noise entering into these components and thus, not satisfying the controllability condition.

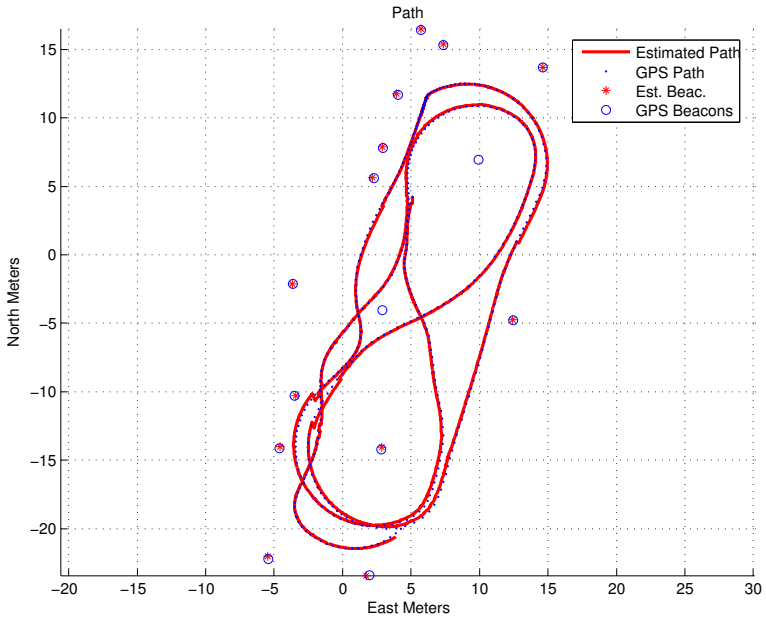


Figure 3.20: Fully observable and fully controllable SLAM run. Vehicle path and landmark location estimates, compared to GPS ground truth with decorrelation of only the map part of the state error covariance matrix.

This can be remedied by assuming an artificial process noise entering into the landmark states. This pseudo-noise can be used to improve the speed of the algorithm, reduced from quadratic to linear, including covariance inflation methods to diagonalise the state error covariance matrix. We observed that this may lead to instability if pseudo-noise is added in a higher state dimensionality than what can be observed. We proposed in this chapter to diagonalise only the map part of the state error covariance, thus guaranteeing convergence of $\mathbf{P}$, and at the same time obtaining in the limit an $O(n)$ algorithm.

Chapter 4

Observability Analysis of Bearing-only SLAM

In this chapter we present an observability analysis of the SLAM system equipped with a bearing-only sensor. Bearing-only SLAM refers to the case in when only the direction to a feature from the sensor is measured, and no range information is available. A common example of bearing-only sensor is a single camera. A block diagram of this system is shown in Figure 4.1.

Bearing-only sensors present a problem to the SLAM algorithm because a single measurement step can only provide incomplete information for the reconstruction of the state space, with the consequence that feature locations cannot be estimated from a single image, and must be computed from the tracking of landmarks over multiple views. From a control theory viewpoint, we can say that one measurement step in bearing-only SLAM renders the system unobservable. The fact that bearing-only SLAM is unobservable, and that it can only become observable by integrating measurements at different time steps once the appropriate motions are made, are known facts within the SLAM community.

Our aim in this chapter is to provide a formal explanation to issues that are intu-

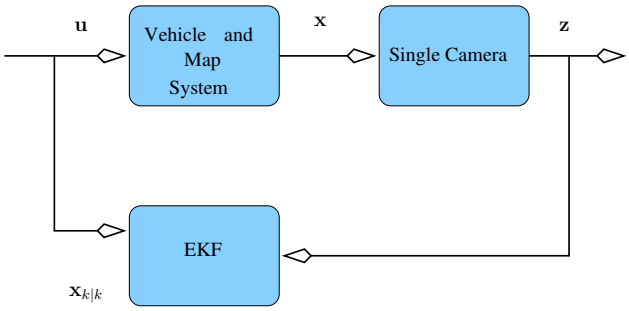


Figure 4.1: Block diagram of the closed loop system for estimation in Visual SLAM process. This system is identical to the one shown in Figure 3.1 except that the general sensor is replaced with a single monocular camera.

itively known or that have been commonly addressed with geometrical concepts. For example, the observability analysis shows why nothing can be said about the location of features that are sensed from a purely rotating bearing-only sensor, or that repeated measurements to a landmark along the line of sight of a forward moving camera do not aid in recovering its range. By providing a formal explanation to these simple issues, we pave the road to understanding more complicated behaviours particular of bearing-only SLAM systems.

Another outcome of our observability analysis is related to sensor trajectories. The performance of the estimator in a bearing-only SLAM system is strongly related to the trajectories followed by the sensor. Ultimately, one would want to drive the system such as to avoid unobservable states. Only then we can guarantee bounded estimation uncertainty. The analysis presented here explains why some bearing-only SLAM systems recover from unobservable conditions.

In the literature, a common observability analysis for the nonlinear systems is performed by using the technique in [62] which involves computing the Lie derivatives over the nonlinear equations of the system. For example, in [106] the authors study the observability of a mobile robot localising itself in a 2D world using bearing-only observations to known features. In [96] the authors study the observability of multiple robots attempting to localise themselves w.r.t the position of other robots. In [11] a

nonlinear observability analysis for the planar robot absolute localisation and mapping problem is performed. Their work concludes that absolute localisation and mapping is observable only after two anchors (markers) have been observed. Other analysis could also be fulfilled by numerically evaluating the observability Gramians.

Instead of performing a nonlinear observability analysis, an alternative explored as well in this chapter is to consider our nonlinear system in its error form which allows the system to be approximated with piecewise constant linear functions. This allows us to use tools in linear systems observability [55] which greatly reduce the complexity of the analysis. These tools also allow us to derive explicit expressions of the unobservable directions in the state space that give insight as to what motions should be made, or what features should be seen so as to improve observability of the entire estimation process. Moreover, by characterising explicitly the form of the unobservable subspaces we get a better understanding of the behaviour of the system.

Following the technique in [55], we analyse here some different SLAM systems, a planar mobile robot, a 6-DOF model with inertial sensors, planar and 6-DOF constant velocity models, all performing SLAM using the measurement modality of bearing-only vision. The approach consists in determining the number of segments needed to obtain a full rank of the so-called stripped observability matrix. The main idea is to model these systems as piecewise constant, focusing mainly in the nullspace basis of this stripped observability matrix finding the unobservable directions in order to enhance observability. A similar methodology was applied to study the observability of SLAM using range and bearing measurements for a UAV with inertial sensors in [77] and [19].

The chapter is organised as follows. In Section 4.1 a brief summary of observability for linear systems is introduced. The rest of the section presents the dynamic observability analysis for different systems; a planar vehicle, the general planar constant velocity model, a 6-DOF model with inertial sensors that reduces the estimated state of the general 6-DOF constant velocity model that we present as well.

In Section 4.2 we present a nonlinear observability analysis for the planar vehicle,

where is shown absolute orientation cannot be recovered for the relative configuration without anchors. Finally, Section 4.3 presents some ideas related with Fisher Information and our problem of observability with bearing-only measurements.

4.1 Observability for Linear Systems

Observability is a notion that plays a major role in filtering and reconstruction of states from inputs and outputs. Together with controllability, observability is central to the understanding of feedback control systems.

Informally, observability answers the question of whether if an initial state $\mathbf{x}(0)$ can be *uniquely* deduced from the history of observations. This requires that the observability Gramian

$$\boldsymbol{\mathcal{O}}(0,t) \triangleq \int_0^t e^{\mathbf{F}^\top \tau} \mathbf{H}\mathbf{H}^\top e^{\mathbf{F}\tau} d\tau \tag{4.1}$$

be nonsingular or, equivalently, that the nullspace of $\boldsymbol{\mathcal{O}}$ is $\mathbf{0} \in \mathbb{R}^n$, where $\mathbf{F}$ and $\mathbf{H}$ are the state transition and measurement matrices, and n is the dimension of the state vector. If this condition is met, then all states are observable.

If $\mathbf{x}(\mathbf{0}) = \xi$ is unobservable over n steps, then it is unobservable over any number of steps. Equivalently, the system is observable if and only if $\text{rank}(\boldsymbol{\mathcal{O}}) = n$ or the dimension of its nullspace $\mathcal{N}$ is equal to zero. The observability condition is an indicator whether or not the system contains all the necessary information to perform the estimation with an error which is bounded. For the case of SLAM, observability implies a bounded error both for localisation of the vehicle and features (see Chapter 3).

For time-invariant systems, the observability condition can be reduced to check whether the matrix

$$\boldsymbol{\mathcal{Q}} \triangleq \begin{bmatrix} \mathbf{H} \\ \mathbf{HF} \\ \vdots \\ \mathbf{HF}^{n-1} \end{bmatrix} \tag{4.2}$$

has rank n or not.

A time-variant system can be approximated by a piecewise constant system with little loss of accuracy and with no loss of the characteristic behaviour of the system [55]. For a piecewise constant system, its j-th segment dynamic model is

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{u}(t) \tag{4.3}$$

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t), \tag{4.4}$$

where,

$$\begin{aligned} \mathbf{F}(t) = \mathbf{F}_1, \mathbf{H}(t) = \mathbf{H}_1 \quad & t_0 < t < t_1 \\ \mathbf{F}(t) = \mathbf{F}_2, \mathbf{H}(t) = \mathbf{H}_2 \quad & t_1 < t < t_2 \\ \vdots \qquad\qquad\quad & \vdots \\ \mathbf{F}(t) = \mathbf{F}_j, \mathbf{H}(t) = \mathbf{H}_j \quad & t_{j-1} < t < t_j \,. \end{aligned} \tag{4.5}$$

The observability matrix (4.2) for each segment is

$$\boldsymbol{\mathcal{Q}}_j = \begin{bmatrix} \mathbf{H}_j \\ \mathbf{H}_j\mathbf{F}_j \\ \vdots \\ \mathbf{H}_j\mathbf{F}_j^{n-1} \end{bmatrix}. \tag{4.6}$$

The system is instantaneously observable at time j if the nullspace of the observability matrix is equal to $\mathbf{0}$. However, to guarantee observability of a piecewise constant system, the full-rank condition must hold for the Total Observability Matrix (TOM)

$$\boldsymbol{\mathcal{Q}}_{\mathtt{TOM},j} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_1 \\ \boldsymbol{\mathcal{Q}}_2 e^{\mathbf{F}_1\tau_1} \\ \vdots \\ \boldsymbol{\mathcal{Q}}_j e^{\mathbf{F}_{j-1}\tau_{j-1}} \dots e^{\mathbf{F}_1\tau_1} \end{bmatrix} \tag{4.7}$$

with τ_j, the time interval between t_j and t_{j-1}.

The Stripped Observability Matrix (SOM), a simplified form of the TOM is defined as

$$\boldsymbol{\mathcal{Q}}_{\mathtt{SOM},j} = \begin{bmatrix} \boldsymbol{\mathcal{Q}}_1 \\ \boldsymbol{\mathcal{Q}}_2 \\ \vdots \\ \boldsymbol{\mathcal{Q}}_j \end{bmatrix}, \tag{4.8}$$

and it can be shown [55] that if

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_j) \subset \mathcal{N}(\mathbf{F}_j), \quad \forall j \tag{4.9}$$

then

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_{\mathtt{SOM}}) = \mathcal{N}(\boldsymbol{\mathcal{Q}}_{\mathtt{TOM}}) \tag{4.10}$$

So, when the condition holds, the analysis is as simple as calculating nullspace of the SOM instead of the TOM. Using the SOM for the study of the observability of piecewise constant systems is much simpler than using the TOM, since the exponentials $e^{\mathbf{F}_j \tau_j}$ need not be computed.

In order to use the SOM method to analyse a nonlinear system, we need to approximate the nonlinear system as piecewise linear. Considering a time step small enough to get a linear relationship as a transition matrix evaluated in the actual time, it is possible to get this kind of approximation. Moreover, considering this piecewise constant system over several segments we are able to perform a sort of a dynamic observability analysis.

4.1.1 Planar Vehicle SLAM

One distinguishing feature of the wheeled mobile robots systems in SLAM is the use of proprioceptive sensor measurements as inputs. This kind of sensor allows the SLAM system to have more information about its own motion. The extereoceptive sensor, for our particular case is a single camera, is used only for correction.

World-centric maps are represented in a global coordinate space in which the state vector is formed by the absolute position of the vehicle and features. Robot-centric or sensor-centric maps, in contrast, are described in measurement space (reference to the robot or sensor). In that case, the state vector has only the distance between vehicle and features.

The world-centric representation is partially observable as in the case of the planar vehicle with range and bearing measurements (see Chapter 3), contrary to the world-centric representation with anchors where the resulting system is fully observable with the constraint of always needing a translational motion.

In the following example we will analyse the consequences, in terms of observability, of having world-centric maps, anchored world-centric maps or relative maps when only odometry and measurements of a single camera are available.

Example 4.1

Let us consider a planar vehicle controlled by linear and angular velocities υ_r and ω_r, respectively. A camera is set on the vehicle centre of rotation. The continuous process model that describes the position and orientation of the centre of projection of the camera is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \upsilon_r \cos\theta \\ \upsilon_r \sin\theta \\ \omega_r \end{bmatrix}. \tag{4.11}$$

In Example 3.4 we used the same process model in a discrete time representation and considered the distance between the centre of rotation of the vehicle and the centre of projection of the camera collinear, $l = 0$.

Assume a measurement model which makes bearing observations to point features.

In the simple planar case, the bearing to a feature is

$$\varphi = \tan^{-1}\left(\frac{r_y}{r_x}\right) - \theta\,, \tag{4.12}$$

where $\mathbf{r} = \mathbf{x}_f - \mathbf{p} = [r_x, r_y]^\top$ *is the 2D distance vector between any given feature and the sensor in global coordinates,* $\mathbf{p} = [x_r, y_r]^\top$ *and* θ *is again the sensor orientation.*

This model, as with most other SLAM models, is nonlinear. We can express the SLAM system in its indirect (error) form where the state contains the vehicle position error $\tilde{\mathbf{p}}$*, the vehicle orientation error* $\tilde{\theta}$*, and the feature position errors* $\tilde{\mathbf{x}}_f^{(1)}, \ldots, \tilde{\mathbf{x}}_f^{(i)}$*. The state error is defined as the difference between the true state and the estimated state.*

World-centric map

The state transition for the j*-th piecewise segment of this error form is*

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \\ \dot{\tilde{\theta}} \\ \dot{\tilde{x}}_f^{(1)} \\ \dot{\tilde{y}}_f^{(1)} \\ \vdots \\ \dot{\tilde{x}}_f^{(i)} \\ \dot{\tilde{y}}_f^{(i)} \end{bmatrix} = \mathbf{F}_j \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \\ \tilde{x}_f^{(1)} \\ \tilde{y}_f^{(1)} \\ \vdots \\ \tilde{x}_f^{(i)} \\ \tilde{y}_f^{(i)} \end{bmatrix}. \tag{4.13}$$

Considering only one feature, we have five components in the state vector (three vehicle pose states and 2 feature position states). The state transition matrix for the

j-th segment of the continuous piecewise linear system is:

$$\mathbf{F}_j = \begin{bmatrix} 0 & 0 & -\upsilon_r, j\sin\theta_j & 0 & 0 \\ 0 & 0 & \upsilon_r, j\cos\theta_j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.14}$$

The measurement model for the j-th piecewise segment, also expressed in this indirect form, is $\tilde{\mathbf{z}} = \mathbf{H}_j\tilde{\mathbf{x}}$*, where*

$$\mathbf{H}_j = \begin{bmatrix} \frac{r_y}{d^2} & -\frac{r_x}{d^2} & -1 & \frac{-r_y}{d^2} & \frac{r_x}{d^2} \end{bmatrix} \tag{4.15}$$

and $d^2 = r_x^2 + r_y^2$*. Note also, that although not explicitly indicated in the notation, the relative feature position* $\mathbf{r}$*, and the distance* d *in* $\mathbf{H}_j$ *are also with respect to the j-th piecewise segment.*

Now, let us consider the first time segment. The observability matrix, from (4.6), is (with $\mathbf{r}, d, \upsilon$*, and* θ *evaluated for the 1st segment)*[1]

$$\boldsymbol{\mathcal{Q}}_1 = \begin{bmatrix} \frac{r_y}{d^2} & -\frac{r_x}{d^2} & -1 & -\frac{r_y}{d^2} & \frac{r_x}{d^2} \\ 0 & 0 & -\frac{r_y}{d^2}\upsilon_r\sin\theta - \frac{r_x}{d^2}\upsilon_r\cos\theta & 0 & 0 \end{bmatrix}. \tag{4.16}$$

The resulting nullspace basis for this instantaneous observability matrix is,

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \left\{ \begin{bmatrix} r_x \\ r_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -r_x \\ 0 \\ 0 \\ 0 \\ r_y \end{bmatrix} \right\}. \tag{4.17}$$

[1] In (4.16) and in the rest of the observability matrices shown throughout the chapter, zero rows are not shown, as they do not contribute to determining the unobservable directions of the state space. These typically come from evaluating the powers of the transition matrix when building them up.

Thus the unobservable modes are $r_x\tilde{x}_r + r_y\tilde{y}_r$, $\tilde{x}_r + \tilde{x}_f$, *and* $-r_x\tilde{x}_r + r_y\tilde{y}_f$. *An important conclusion from (4.17) is that all of the nullspace basis vectors are orthogonal to the vehicle orientation state (i.e. the third element of every nullspace basis vector is zero). This means that the vehicle orientation is the only completely observable state from the original state vector; except for the case of* $v_r = 0$, *in which case the null space basis is of dimension 4, and the orientation is also unobservable as is shown in the following expression,*

$$\mathcal{N}(\mathcal{Q}_1)\,|_{v_r=0} = \left\{ \begin{bmatrix} -r_x \\ -r_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r_x \\ 0 \\ 0 \\ 0 \\ -r_y \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} d^2 \\ 0 \\ -r_y \\ 0 \\ 0 \end{bmatrix} \right\}. \tag{4.18}$$

Note also that the values of r_x *and* r_y *are irrelevant, that is, regardless of where the feature is located with respect to the sensor, there will always be three unobservable modes in one time segment (except of course, for the singular case of a feature on the optical centre of the camera* $r_y = 0$*).*

Needless to say, the nullspace basis for the TOM is exactly the same because the null space of $\mathbf{F}_j$,

$$\mathcal{N}(\mathbf{F}_j) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} \tag{4.19}$$

contains the null space of the SOM.

Extending the analysis for two time segments the SOM takes the form in (4.8), and

computing its nullspace basis we get

$$\mathcal{N}(\mathcal{Q}_{\texttt{SOM}}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{4.20}$$

Adding more features or time segments does not modify the observability conditions. That is, the system will remain partially observable with non-observable modes $\tilde{x}_r + \sum_i \tilde{x}_f^{(i)}$ *and* $\tilde{y}_r + \sum_i \tilde{y}_f^{(i)}$ *as the nullspace basis above shows for the case of only one feature.*

Anchored world-centric map

Consider the same state space as before. Now we change the characteristics of the unknown environment setting features such that their exact position in 2D is known. We call this known features anchors *following the spirit of [3]. Some other authors have called them* markers *[11].*

The observability condition using anchors instead of relative mapping for the planar case of range and bearing have been analysed before [3, 85]. In that case, only one time segment and two anchors are needed. In the nonlinear observability analysis performed by [11] for localisation of a planar vehicle with bearing-only measurements it is suggested that two anchors are needed in order to guarantee full observability. For our linear analysis of the planar bearing-only case, two anchors and two time segments are needed for the system to become fully observable. However, it should be noticed that one anchor seen from two different time steps is equivalent to having two anchors.

The new measurement matrix for the j-th segment is,

$$\mathbf{H}_j = \begin{bmatrix} \frac{r_y}{d^2} & -\frac{r_x}{d^2} & -1 & -\frac{r_y}{d^2} & \frac{r_x}{d^2} \\ \frac{a_y^{(1)}-y}{(da^{(1)})^2} & -\frac{a_x^{(1)}-x}{(da^{(1)})^2} & -1 & 0 & 0 \\ & & \ddots & & \\ \frac{a_y^{(a)}-y}{(da^{(a)})^2} & -\frac{a_x^{(a)}-x}{(da^{(a)})^2} & -1 & 0 & 0 \end{bmatrix}, \tag{4.21}$$

where $da^{(a)} = (a_y^{(a)} - y)^2 + (a_x^{(a)} - x)^2$ and a is the number of anchors.

Notice that more measurements of the relative position of the vehicle are available because of the anchor, therefore the instantaneous observability matrix is larger than in the world-centric with no anchors case.

Performing the same analysis as above, the null space basis of the instantaneous observability matrix using the world-centric model with two anchors and one unknown feature is,

$$\left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ -r_{x,1} \\ -r_{y,1} \end{bmatrix} \right\}. \tag{4.22}$$

Moreover $\mathcal{N}(\mathcal{Q}_{\texttt{SOM}}) = 0$, and the analysis shows that the planar vehicle with one feature and two anchors is completely observable in two time segments. Again $\mathcal{N}(\mathcal{Q}_j) \subset \mathcal{N}(\mathbf{F}_j)$ therefore the TOM is observable for one or more anchors so the condition 4.10 is set.

Relative map

Alternatively, let us consider as the new state the relative position error of the vehicle with respect to to the features rather than separate global vehicle and map feature errors. In other words, let us consider a relative (considering the case when the vehicle's centre of rotation corresponds to the camera's centre of projection) model, where $\tilde{\mathbf{r}}^{(i)}$ is again

$\tilde{\mathbf{x}_f}^{(i)} - \tilde{\mathbf{p}}$. *The state vector becomes*

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{r}}^{(1)} \\ \tilde{\mathbf{r}}^{(2)} \\ \vdots \\ \tilde{\mathbf{r}}^{(i)} \\ \tilde{\theta} \end{bmatrix}. \tag{4.23}$$

Notice that the measurement model observes directly the relative angle between the new state but we still want to know the orientation of the vehicle with respect to an absolute frame, then we will still consider the measurement model as in (4.12) and

$$\mathbf{H}_j^i = \begin{bmatrix} \frac{r^(1)_y}{(d^{(1)})^2} & -\frac{r^(1)_x}{(d^{(1)})^2} & -1 & \cdots & 0 & 0 & 0 \\ & & & \ddots & & & \\ 0 & 0 & -1 & \cdots & \frac{r^(i)_y}{(d^{(i)})^2} & -\frac{r^(i)_x}{(d^{(i)})^2} & 0 \end{bmatrix}. \tag{4.24}$$

The dimension of the state vector is reduced by two, consequently the nullspace basis of the instant observability matrix is reduced to one vector for each feature, i.e.

$$\mathcal{N}(\mathcal{Q}_1) = \left\{ \begin{bmatrix} r_{x,1}^{(1)} \\ r_{y,1}^{(1)} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ r_{x,1}^{(2)} \\ r_{y,1}^{(2)} \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ r_{x,1}^{(i)} \\ r_{y,1}^{(i)} \\ 0 \end{bmatrix} \right\}. \tag{4.25}$$

In the case of having only one feature, the nullspace dimension of the instantaneous observability matrix is only one, and again, the observable state is the orientation.

As the number of features increases, the orientation is the only state that remains observable. According to this analysis, the actual orientation of the vehicle referred to the global coordinate frame is being recovered because of the linearisation performed to get the error (indirect) form, but this is not necessarily true[2]*. In this case having the reference frame on the first sensor position, the initial orientation has been implicitly considered equal to zero, and because of the use of the proprioceptive sensors (encoders) it is always possible to get the relative orientation of the vehicle.*

For the particular case of having a feature aligned exactly in front of the camera, $(r_y = 0)$*, the unobservable mode for the one time segment observability matrix is* $\tilde{r}_x$*. This means that having only such measurement available, the relative position between the vehicle and the feature is unrecoverable, but the orientation can be recovered. A similar situation happens when the feature is aligned with the image plane,* $(r_x = 0)$*, and the unobservable mode is* $\tilde{r}_y$*. Although it is unlikely that such case will be detected by the sensor, unless an omnidirectional camera is used.*

Opposite to global localisation or world-centric model with no anchors, this SLAM system becomes fully observable in two time-segments, as can be verified from the SOM and the TOM becoming full-rank.

This result is of special importance, as it gives a theoretical grounding to what has become common practice in SLAM systems. That is, on the use of relative maps as opposed to world-centric maps.

As in the one time segment case, the system will remain unobservable when a feature is aligned with the field of view of the sensor, $r_y = 0$*, with an unobservable mode* $\tilde{r}_x$*. The same does not happen however, for a feature aligned along the* y *axis, as the difference* r_x *will move away from zero during the second segment, and the SOM and TOM become full rank.*

Moreover, when there is no translation (pure rotation), the nullspace basis augments

[2] A further nonlinear analysis using Lie differentiation, following the approach in [96], shows that actually the orientation with respect to a global frame is not observable for this relative representation.

by one vector. Therefore, neither the position nor the orientation can be fully recovered (only a combination of the two). In this case, the SOM and TOM are not full rank, and consequently there is no way to estimate the whole state for bearing-only SLAM when $v_r = 0$. On the other hand, when there is no rotation, the system is completely observable in two time segments, except again, for the unfortunate case when a feature is aligned along the line of sight of the sensor.

4.1.2 Inertial SLAM

6-DOF inertial systems, for example an unmanned aerial vehicle, have again proprioceptive sensors that give to the estimator more information about the state. Moreover, in these systems not only the inputs are consider as proprioceptive, but also the measurements provided by the inertial sensors, reducing this way the size of the state vector to be estimated by the filter.

World-centric representations are partially observable as in any SLAM case. In order to consider a world-centric with anchors representation more than two anchors must be considered to get the system completely observable. Again relative representations can be considered to model this kind of systems, producing a fully observable bearing-only SLAM system.

Example 4.2 ———————————————————

Let us consider a non-planar motion model with 3D features. For example a model that requires six degree of freedom equations to predict the position, velocity and altitude of the vehicle in an inertial navigation framework. The continuous process model is

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{C}^{\top}\mathbf{f}^{s} + \mathbf{g} \\ E\boldsymbol{\omega} \end{bmatrix} . \tag{4.26}$$

where $\mathbf{f}^s$ *and* $\boldsymbol{\omega}$ *are the sensor-frame referenced vehicle accelerations and rotation rates as provided by inertial sensors on the vehicle, and* $\mathbf{g}$ *is the acceleration due to gravity.*

The direction cosine matrix $\mathbf{C}$ *is given by*

$$\mathbf{C} = \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\theta s_\phi \\ c_\psi s_\theta c_\phi + s_\psi s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi & c_\theta c_\phi \end{bmatrix}, \tag{4.27}$$

and the rotation rate is

$$\mathbf{E} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}. \tag{4.28}$$

As with our previous model, this inertial SLAM system can also be casted in indirect form [77], with the states of the system being the vehicle pose errors and feature location errors. The system can be considered as piecewise linear constant, and its SOM can be used to analyse its observability properties. This is done in [19] for the range and bearing case. They show this system to be completely observable in two time segments for the relative model and partially observable for the world-centric model in n-landmarks. Our aim is to perform here the same analysis for the more challenging bearing only case.

In a three-dimensional world, two bearing angles are typically measured, the azimuth φ*, and the elevation* ϑ*, usually coming from a vision camera with 3D absolute position* $\mathbf{p}$*, and absolute Euler angles* $\boldsymbol{\Psi}$*. In this case, the measurement model is*

$$\mathbf{z} = \begin{bmatrix} \varphi \\ \vartheta \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{r_y^s}{r_x^s}\right) \\ \tan^{-1}\left(\frac{r_z^s}{\sqrt{(r_x^s)^2+(r_y^s)^2}}\right) \end{bmatrix}, \tag{4.29}$$

where $\mathbf{r}^s = \mathbf{C}(\mathbf{x}_f - \mathbf{p})$ *is the relative position of the feature with respect to the sensor.*

World-centric map

The inertial SLAM system in the indirect form with zero input, because it does not affect the analysis, is

$$\begin{bmatrix} \dot{\tilde{\mathbf{p}}} \\ \dot{\tilde{\mathbf{v}}} \\ \dot{\tilde{\Psi}} \\ \dot{\tilde{\mathbf{x}}}_f \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{v}} \\ [\times \mathbf{f}]\,\tilde{\Psi} \\ \mathbf{C}^{\top}\tilde{\boldsymbol{\omega}} \\ \mathbf{0} \end{bmatrix}. \tag{4.30}$$

The segment transition matrix of this example is

$$\mathbf{F}_j = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{3\times 3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\times \mathbf{f}_j] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.31}$$

For range and bearing measurements it is possible to find a linear relationship between the measurements and the relative states using the indirect form [77]. Transforming into Cartesian coordinates the information contributed by range and bearing errors they end up with an expression equivalent to

$$\tilde{\mathbf{r}}_m = \tilde{\mathbf{H}}_m \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{v}} \\ \tilde{\boldsymbol{\Psi}} \\ \tilde{\mathbf{x}}_f \end{bmatrix}, \tag{4.32}$$

where $\tilde{\mathbf{H}}_m = \begin{bmatrix} -\mathbf{I}_{3\times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}] & \mathbf{I}_{3\times 3} \end{bmatrix}$ *is the measurement matrix for range and bearing observations and* $[\times \hat{\mathbf{r}}]$ *is the skew symmetric matrix of the estimate of the relative position between the feature and the vehicle positions. For bearing-only measurements the above expression must be pre-multiplied by the partial derivative of (4.29) with respect to the position of the feature in sensor coordinates,* $\mathbf{r}^s$*. Then, the measurement*

matrix for our bearing-only case becomes

$$\mathbf{H}_j = \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} \tilde{\mathbf{H}}_m \,, \tag{4.33}$$

where

$$\frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} = \begin{bmatrix} -\frac{r_y^s}{(d^s)^2} & \frac{r_x^s}{(d^s)^2} & \mathbf{0}_{2\times 3} \\ -\frac{r_x^s r_z^s}{d^s((d^s)^2+(r_z^s)^2)} & \frac{r_y^s r_z^s}{d^s((d^s)^2+(r_z^s)^2)} & \frac{d^s}{(d^s)^2+(r_z^s)^2} \end{bmatrix}, \tag{4.34}$$

with $(d^s)^2 = (r_x^s)^2 + (r_y^s)^2$.

Again as in the previous systems the world-centric model is partially observable, the non-observable directions correspond to the sensor position and the landmark position according to the null space of the SOM,

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_{\mathtt{SOM}}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} = \left\{ \begin{bmatrix} \mathbf{i} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{i} \end{bmatrix}, \begin{bmatrix} \mathbf{j} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{j} \end{bmatrix}, \begin{bmatrix} \mathbf{k} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{k} \end{bmatrix} \right\}, \tag{4.35}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ *are the unitary vectors in the* x, y *and* z *directions respectively.*

Our conclusion of this observability analysis is that, for a world-centric inertial system performing SLAM with only one camera. The system is not fully observable, only linear velocity and misalignments (related with the orientation of the vehicle) are recovered.

Anchored world-centric map

Let us consider now the part corresponding to the anchor in the measurement model:

$$\mathbf{H}_a = \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} \tilde{\mathbf{H}}_a \,, \tag{4.36}$$

where

$$\tilde{\mathbf{H}}_a = \begin{bmatrix} -\mathbf{I}_{3\times 3} & \mathbf{0} & [\times \hat{\mathbf{r}}] & \mathbf{0} \end{bmatrix} , \tag{4.37}$$

and the new measurement matrix for this particular case is formed by appending (4.33) with (4.36) for each 3D anchor considered in the environment.

The analysis is shown here for one feature only, but the result holds for more features. The null space basis of the instantaneous observability matrix with one or two anchors gives a partially observable system in two time segments and for three or more anchors and one unknown feature the nullspace of the observability matrix for the first segment is

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \left\{ \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{C}\hat{\mathbf{r}} \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{r}} \times \mathbf{f} \\ \mathbf{0}_{3\times 3} \\ \mathbf{f} \\ \mathbf{0}_{3\times 3} \end{bmatrix} \right\} . \tag{4.38}$$

That is, in one time step the system can recover only linear velocities, because it has some fixed references to the world reference frame. On the other hand, using three or more anchors, the SOM is full rank in two time segments.

For the particular case of having no linear acceleration ($\mathbf{f} = \mathbf{0}$) *the dimension of* $\mathcal{N}(\boldsymbol{\mathcal{Q}}_1)$ *is equal to four, linear velocities are still recovered in one step, but the null space basis of the SOM results of dimension equal to one recovering the position of the feature plus the velocity but not the vehicle's position and orientation.*

Relative map

Let us change the state vector in which the error in the relative position of each

feature with respect to the vehicle position is considered rather than the separate global vehicle and the map feature positions,

$$\mathbf{x} = \begin{bmatrix} \tilde{\mathbf{r}} \\ \tilde{\mathbf{v}} \\ \tilde{\Psi} \end{bmatrix} \tag{4.39}$$

with

$$\tilde{\mathbf{r}}_m = \begin{bmatrix} -\mathbf{I}_{3\times3} & \mathbf{0} & [\times\hat{\mathbf{r}}] \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{r}} \\ \tilde{\mathbf{v}} \\ \tilde{\Psi} \end{bmatrix}. \tag{4.40}$$

The observability matrix for one time segment is

$$\boldsymbol{\mathcal{Q}}_1 = \begin{bmatrix} \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} \times \hat{\mathbf{r}} \\ \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}^s} \times \mathbf{f} \end{bmatrix}. \tag{4.41}$$

There are only six independent rows given that the partials $\frac{\partial \mathbf{z}}{\partial \mathbf{r}^s}$ *are of size* 2×3*. The nullspace basis of this matrix is*

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \left\{ \begin{bmatrix} \mathbf{C}\hat{\mathbf{r}} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\zeta}(\mathbf{C}, \hat{\mathbf{r}}, \mathbf{f}) \\ \mathbf{0}_{3\times3} \\ \mathbf{f} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{C}\hat{\mathbf{r}} \\ \mathbf{0}_{3\times3} \end{bmatrix} \right\}, \tag{4.42}$$

where

$$\boldsymbol{\zeta}(\mathbf{C}, \hat{\mathbf{r}}, \mathbf{f}) = \begin{bmatrix} \zeta_x \\ 0 \\ \zeta_z \end{bmatrix}. \tag{4.43}$$

For the sake of clarity we only show the form of this function. The full expression is a long relation between the linear acceleration, the estimated relative position, and the orientation misalignment angles. Noteworthy, regardless of its final form, it has

one zero along $\tilde{r}_x$ and nonzero elements in the other two directions.

Then, from a visual inspection of 4.42 we can conclude that the relative position between vehicle and feature errors, and the misalignment can only be partially reconstructed during one time segment in our bearing-only case.

We analyse now the case of two time segments for the inertial bearing-only SLAM. The corresponding SOM is

$$\boldsymbol{\mathcal{Q}}_{\mathtt{SOM},2} = \begin{bmatrix} \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} \times \hat{\mathbf{r}}_1 \\ \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} \times \mathbf{f}_1 \\ \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} \times \hat{\mathbf{r}}_2 \\ \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} \times \mathbf{f}_2 \end{bmatrix}, \tag{4.44}$$

this becomes full rank (as well as the TOM), although $\mathcal{N}(\boldsymbol{\mathcal{Q}}_j) \nsubseteq \mathcal{N}(\mathbf{F}_j)$.

Considering the particular case of setting to zero the linear acceleration, i.e. $\mathbf{f} = \mathbf{0}$. The systems adopts a constant velocity form, for which only position, linear velocity, and misalignment is estimated. The rank of the instantaneous observability matrix $\boldsymbol{\mathcal{Q}}_1$ becomes four; thus, when the system does not accelerate it loses observability. The interesting issue is that for two time segments the corresponding SOM is still not full rank. This system becomes in a sort of constant velocity model. The only observable state is the linear velocity error with a nullspace basis of rank two.

Constant velocity models are of great value when vehicle dynamics are difficult to obtain. In the following subsection we devote some time to the observability analysis of such a model for monocular SLAM.

4.1.3 Planar Constant Velocity SLAM

Constructing a motion model for an agile camera which is carried by a person, mobile robot or other moving platform may seem to be fundamentally different to modelling the motion of a wheeled mobile robot, or an aerial vehicle with inertial sensors: the key difference is that in the mobile robot and the aerial vehicle cases we can use the inputs driving the motion or the inertial information to predict motion estimates, whereas there is not such prior information about an agile camera's movements, i.e. it does not consider proprioceptive sensing.

Example 4.3

Let us consider a planar constant velocity motion model that is linear and constant. Nonetheless, we will use the same error form as before to perform the analysis because the measurement model is still nonlinear, and in order to be in the same context as with the planar vehicle model. The system model with no landmarks is given by

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ \omega_z \\ f_x \\ f_y \\ \alpha_z \end{bmatrix}, \tag{4.45}$$

where $[v_x, v_y]$, ω_z, $[f_x, f_y]$ and α_z are the linear velocity, the angular velocity in the orthogonal direction, linear acceleration and angular acceleration, respectively.

The world-centric models with no anchors for constant velocity systems performing SLAM are partially observable. In such case $dim(\mathcal{N}(\boldsymbol{\mathcal{Q}}_1)) = 6$ recovering no-directions, and $dim(\mathcal{N}(\boldsymbol{\mathcal{Q}}_{\texttt{SOM}})) = 2$ only recovers the angular and linear velocities.

Anchored world-centric map

The state space consists of two-dimensional position, orientantion and two-dimensional

linear velocity, angular velocities and unknown features:

$$\mathbf{x} = [x_r, y_r, \theta, v_x, v_y, \omega_z, x_f^{(1)}, y_f^{(1)}, \dots, x_f^{(i)}, y_f^{(i)}]^\top ,$$

where the constant transition matrix for the case of one unknown feature is,

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4.46}$$

The measurement model used for this analysis is the same as in (4.12), adding the corresponding measurements to the anchors in the Jacobian results (4.21). Obviously, both the instantaneous observability matrix and the SOM change depending on the number of anchors set in the environment.

For the observability analysis using only one anchor, the rank of the instantaneous observability matrix is equal to four. The linear and angular velocities are recovered in the first time segment. The next time segment gives us $dim(\mathcal{N}(\boldsymbol{\mathcal{Q}}_{\mathtt{SOM},2})) = 1$ *and no more directions are being recovered.*

Moreover, the constant velocity model with two anchors is fully observable in two time segments. However the instantaneous observability matrix it has rank six, recovering only linear and constant velocities in one step.

In addition, considering more than two anchors, the null space basis of the instan-

taneous observability matrix is,

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -r_{x,1}^{(1)} \\ -r_{y,1}^{(1)} \end{bmatrix}. \tag{4.47}$$

Therefore, the unobservable direction is only along the feature in the first time segment, that is we are able to recover position, orientation and velocities in the first time segment only if the camera is translating.

The null space basis of $\mathbf{F}_j$ *is*

$$\mathcal{N}(\mathbf{F}_j) = \left\{ \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}, \tag{4.48}$$

given that $\mathcal{N}(\boldsymbol{\mathcal{Q}}_j)$ *is contained in the null space basis of* $\mathbf{F}_j$*, the condition (4.10) is met and the used of the SOM is valid.*

In two time segments, the null space basis of the SOM is empty as in the case with two anchors. Consequently the TOM is full rank in two time segments for more than two anchors.

Relative map

Using the relative position between the features and camera, the system with only one feature becomes

$$\begin{bmatrix} \dot{\tilde{r}}_x \\ \dot{\tilde{r}}_y \\ \dot{\tilde{\theta}} \\ \dot{\tilde{v}}_x \\ \dot{\tilde{v}}_y \\ \dot{\tilde{\omega}} \end{bmatrix} = \mathbf{F} \begin{bmatrix} \tilde{r}_x \\ \tilde{r}_y \\ \tilde{\theta} \\ \tilde{v}_x \\ \tilde{v}_y \\ \tilde{\omega} \end{bmatrix}, \tag{4.49}$$

with the constant transition matrix

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.50}$$

The instantaneous observability matrix using the measurement matrix in (4.24) considering one features results in ($\mathbf{r}$ *and* d *evaluated for the 1st time segment)*

$$\boldsymbol{\mathcal{Q}}_1 = \begin{bmatrix} \frac{r_y}{d^2} & -\frac{r_x}{d^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{r_y}{d^2} & -\frac{r_x}{d^2} & 0 \end{bmatrix}, \tag{4.51}$$

and the nullspace basis of this matrix is

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \left\{ \begin{bmatrix} r_x \\ r_y \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ r_x \\ r_y \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{4.52}$$

For this system, none of the state components is completely observable in one step, not even the orientation, and it has four unobservable modes, as can be seen from the above instantaneous observability matrix nullspace basis (4.52).

When the analysis is made for two or more time segments, two unobservable modes spanned the null space where nor the orientation, nor the angular velocity can be recovered. The nullspace basis for the two time segments SOM is

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_{\texttt{SOM}}) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{4.53}$$

The addition of more features to the state space does not change this result. Nevertheless, there is one particular constraint, the velocity must be nonzero to at least recover the relative positions.

The main difference between this system and the planar vehicle, is that the constant velocity system is uniquely provided with extereoceptive sensors. This is the reason because no relative representations in constant velocity models achieve the rank condition

to be fully observable.

4.1.4 6-DOF Constant Velocity SLAM

A stochastic constant linear velocity, constant angular velocity model does not assume that the camera moves at a constant velocity over all time, but that the statistical model of its motion in a time step is that on average, except for undetermined accelerations that occur with a Gaussian profile. The implication of this kind of models is that certain smoothness on the camera motion in any direction can be modelled, i.e. very large accelerations are relative unlikely.

As we have confirmed for all the previous models, world-centric representation with anchors is the only formulation that lets systems with extereoceptive sensors only be fully observable it there exists some translation. For such reason, in the reminder of this section, we will only analyse the case of the global representation.

Example 4.4 ———————————————————————

A constant velocity motion model with 6 degree of freedom assumes that the camera can be attached to any mobile platform and is free to move in any direction in $\mathbb{R}^3 \times SO(3)$*. A smooth unconstrained constant-velocity motion model of the form*

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\boldsymbol{\Psi}} \\ \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{x}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{f} \\ \mathbf{E}\boldsymbol{\omega} \\ \boldsymbol{\alpha} \\ \mathbf{0} \end{bmatrix}, \tag{4.54}$$

where $\dot{\boldsymbol{\Psi}}$ *is the Euler representation of the orientation angles.*

Anchored world-centric map

Expressing the 3D constant velocity model in its error form using the same change

of variables as in the Inertial SLAM case, the model transforms to

$$\begin{bmatrix} \dot{\tilde{\mathbf{r}}} \\ \dot{\tilde{\mathbf{v}}} \\ \dot{\tilde{\Psi}} \\ \dot{\tilde{\boldsymbol{\omega}}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{v}} \\ \mathbf{0} \\ \hat{\mathbf{C}}^{\top}\tilde{\boldsymbol{\omega}} \\ \mathbf{0} \end{bmatrix}, \tag{4.55}$$

and its transition matrix is simply

$$\mathbf{F}_j = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{\mathbf{C}}^{\top} \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that the 6-DOF constant velocity case is very similar to the inertial SLAM example, with the difference that three more states are added for the the angular velocity. In the inertial SLAM case these states are typically measured with gyroscopes and as such are not estimated.

The measurement model is the same as in the Inertial SLAM case (4.29) and the measurement matrix is the same as (4.33). Adding to the measurements the observations to two or more anchors, the extension of the measurement Jacobian takes also the form in (4.36). For this model, the nullspace basis of the instantaneous observability matrix has dimension seven, with only two anchors. Or equivalently, we can say that with one camera, using a constant velocity model, and for one time segment, it is only possible to recover eight independent combinations of the state space.

The null space basis of the first segment observability matrix is,

$$\mathcal{N}(\boldsymbol{\mathcal{Q}}_1) = \left\{ \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{C_1^\top} \times \mathbf{r} \\ \mathbf{0}_{3\times3} \\ \mathbf{i} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{C_2^\top} \times \mathbf{r} \\ \mathbf{0}_{3\times3} \\ \mathbf{j} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{C_3^\top} \times \mathbf{r} \\ \mathbf{0}_{3\times3} \\ \mathbf{k} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} [\times\mathbf{r}]_\mathbf{1} \\ \mathbf{0}_{3\times3} \\ \mathbf{i} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} -[\times\mathbf{r}]_\mathbf{2} \\ \mathbf{0}_{3\times3} \\ \mathbf{j} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} -[\times\mathbf{r}]_\mathbf{3} \\ \mathbf{0}_{3\times3} \\ \mathbf{k} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ -\mathbf{r^s} \end{bmatrix} \right\}. \tag{4.56}$$

Notice that the size of $\frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s}$ depends on the number of anchors we are using, for this particular case using one unknown feature and three or more anchors, the size is $10\times n$. In the general case the size of H_j will be $2nm + 2na \times n$.

Adding one more piecewise segment to the analysis,

$$\boldsymbol{\mathcal{Q}}_{\texttt{SOM},2} = \begin{bmatrix} \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} \times \hat{\mathbf{r}}_1 & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_1^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{p}_1^s} \times \hat{\mathbf{C}}_1^\top \hat{\mathbf{r}}_1 \\ \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} \times \hat{\mathbf{r}}_2 & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{r}_2^s} & \mathbf{0} & \frac{\partial \mathbf{z}}{\partial \mathbf{p}_2^s} \times \hat{\mathbf{C}}_2^\top \hat{\mathbf{r}}_2 \end{bmatrix}, \tag{4.57}$$

which is full rank. Then the world-centric constant velocity model with for more than two anchors is observable in two steps.

This result is coincident to what we have shown for the constant velocity planar model.

4.2 Observability for Nonlinear Systems

For nonlinear systems Hermann and Krener [62] emphasise the dependence of the observability on the control inputs, contrary to the linear case. An analysis based on this dependence, plus the lack of error due to the linearisation, seems to give a better understanding of the nonlinear system.

Consider the system:

$$\Sigma : \begin{array}{c} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{z} = \mathbf{h}(\mathbf{x}) \end{array} \tag{4.58}$$

where $\mathbf{u} \in \mathbb{R}^l$, $\mathbf{x} \in \mathbb{R}^n$ is the state vector of a manifold M of dimension n and $\mathbf{z} \in \mathbb{R}^m$.

This system is observable if, and only if, the state can be expressed as a function of the observation $\mathbf{h}$, the input, and their derivatives with respect to time:

$$\mathbf{x} = \Upsilon\left(\mathbf{z}, \dot{\mathbf{z}}, \ldots, \mathbf{z}^{(b_1)}, \mathbf{u}, \dot{\mathbf{z}}, \ldots, \mathbf{z}^{(b_2)}\right) \tag{4.59}$$

For nonlinear systems, the observability can be checked by computing the dimension of the smallest codistribution that contains the output one-form[3], and is invariant with the control vector fields.

Let us consider a nonlinear system which is linear with respect to the control input, i.e.

$$\dot{\mathbf{x}} = \sum_p g_p(\mathbf{x})\mathbf{u}_p \,. \tag{4.60}$$

A sufficient condition allows one to conclude the observability of the system, by computing the dimension of the space spanned by the gradients of Ξ, where Ξ is the space that contains all the Lie derivatives [121] of the field $h(\mathbf{x})$ along the control vector fields of $\mathbf{g} = [g_1, g_2, \ldots, g_p]^\top$.

[3] Also called a covector, is a linear function which maps each vector in a vector space to a real number, such that the mapping is invariant with respect to coordinate transformations of the vector space.

Then, if the rank of

$$d\Xi = \left[\begin{array}{ccc|ccc} L_{\mathbf{g}}^0 dh_1 & \dots & L_{\mathbf{g}}^{n-1} dh_1 & \dots & L_{\mathbf{g}}^0 dh_m & \dots & L_{\mathbf{g}}^{n-1} dh_m \end{array} \right]^\top \tag{4.61}$$

is equal to the dimension n of the system, the *observability rank condition* according to [62] is satisfied and the system is *locally weakly observable*.

Specifically, the Lie derivative of a scalar $h(\mathbf{x})$ with respect to a vector field $\mathbf{g}(\mathbf{x})$ is the differential of a function defined by

$$L_{\mathbf{g}}^1 h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) \,. \tag{4.62}$$

Similarly the Lie derivative of $dh(\mathbf{x})$ with respect to $\mathbf{g}(\mathbf{x})$ is defined by

$$\begin{aligned} L_{\mathbf{g}}^1 dh(\mathbf{x}) &= dL_{\mathbf{g}}^1 h(\mathbf{x}) & (4.63) \\ &= \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \frac{\mathbf{g}(\mathbf{x})}{\mathbf{x}} + \left(\frac{\partial}{\partial \mathbf{x}} \left(\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right)^\top \mathbf{g}(\mathbf{x}) \right)^\top . & (4.64) \end{aligned}$$

The superscripts indicates recursive Lie derivatives defined as follows,

$$\begin{aligned} L_{\mathbf{g}}^0 dh(\mathbf{x}) &= dh(\mathbf{x}) & (4.65) \\ L_{\mathbf{g}}^q dh(\mathbf{x}) &= L_{\mathbf{g}}^{q-1} dh(\mathbf{x}) \frac{\mathbf{g}(\mathbf{x})}{\mathbf{x}} + \left(\frac{\partial}{\partial \mathbf{x}} \left(L_{\mathbf{g}}^{q-1} dh(\mathbf{x}) \right)^\top \mathbf{g}(\mathbf{x}) \right)^\top & (4.66) \end{aligned}$$

Making an analogy with the linear observability analysis of the Section 4.1, using (4.61) as the observability matrix, not only is possible to know the degree of observability of the system, but also to obtain its nullspace basis and to look at the nonobservable directions of the state as the following example shows.

Example 4.5

Let us consider the relative representation of the planar vehicle (4.11) in Example 4.1. For the nonlinear observability analysis it is convenient to introduce polar

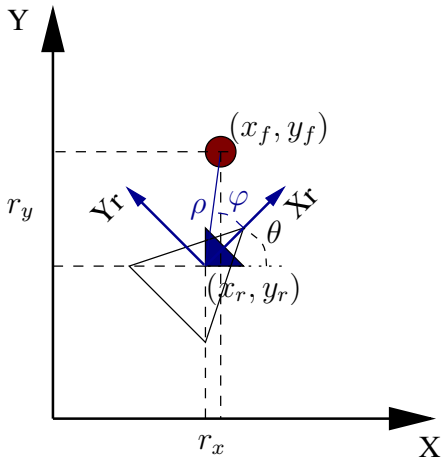


Figure 4.2: Camera on vehicle SLAM sensor-centric configuration. The Cartesian coordinates of the control point (x_r, y_r) are located on the base of the camera. The $(\rho, \varphi + \theta)$ are the polar coordinates of the base of the camera with respect to X_r and Y_r relative frame.

coordinates for the relative state $[r_x, r_y, \theta]$,

$$\rho = \sqrt{r_x^2 + r_y^2} \tag{4.67}$$

$$\varphi = atan2(r_y, r_x) - \theta\,, \tag{4.68}$$

where ρ *is the distance between the vehicle and a landmark and* φ *is the bearing of the landmark with respect to the vehicle considering vehicle orientation. Figure 4.2 shows the vehicle landmark configuration in the global and in the relative frames.*

Now the state will be $[\rho, \varphi, \theta]$*, and its dynamics from (4.11) given that* $r_x = \rho\cos(\theta + \varphi)$ *and* $r_y = \rho\sin(\theta + \varphi)$ *is*

$$\dot{\rho} = \upsilon\cos\varphi \tag{4.69}$$

$$\dot{\varphi} = \frac{\upsilon}{\rho}\sin\varphi - \omega \tag{4.70}$$

$$\dot{\theta} = \omega\,. \tag{4.71}$$

The system is already linear with respect to the input, then with $u_1 = \upsilon, u_2 = \omega$ *the*

vector field $\mathbf{g}$ *is*

$$g_1 = [\cos\varphi, \frac{1}{\rho}\sin\varphi, -1]^\top \tag{4.72}$$

$$g_2 = [0, 0, 1]^\top \tag{4.73}$$

For bearing-only measurements (4.12) $h = \varphi$*, we have*

$$L^0 h = \varphi\,, \tag{4.74}$$

and the first order Lie derivatives

$$L^1_{g_1} h = \tfrac{1}{\rho}\sin\varphi \quad L^1_{g_2} h = 0\,. \tag{4.75}$$

Having the vectors that spanned Ξ *and differentiating their Lie derivatives as require (4.61), we obtain*

$$d\Xi = \begin{bmatrix} 0 & -\frac{\sin\varphi}{\rho^2} & 0 \\ 1 & \frac{\cos\varphi}{\rho} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.76}$$

in a matrix form.

The rank condition is not achieve because the dimension of the space spanned by $d\Xi$ *is equal to* 2 *and* $n = 3$*. The nullspace basis is*

$$\mathcal{N}(d\Xi) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4.77}$$

Therefore, it is clear from this analysis that the global orientation of the vehicle is not observable, contrary of what the linear observability analysis shows.

4.3 Fisher Information Matrix and Observability

The motivation for finding conditions implying the non singularity of the limiting Fisher information matrix or simply information matrix is intrinsically linked with the asymptotic properties of the estimator in those models (consistency or asymptotic stability). The estimation problem is observable if the information matrix is nonsingular for a finite number of samples. Given observability, a solution for Riccati equation exists, and performance bounds on the accuracy of the state estimate can be determined. When a system is unobservable, the observability matrices describe which components of the state cannot be determined. Likewise, observability relates to the convergence of recursive the estimator used to solve the estimation problem.

An estimation problem may be unobservable for some time, but then become observable as the system evolves. In particular the estimation in bearing-only SLAM problem is dependent on the vehicle path through the nonlinear measurement equations. This dependence is related with the dynamic observability of the system. The observability of bearing-only SLAM requires sufficient vehicle motion. While the state estimation remains unobservable or only partially observable, the behaviour of the unobservable estimator states could be marginally stable or unstable in some cases, and in such cases the filter diverges. Only when sufficient motion and appropriate observations are achieved can the entire state estimate converge toward true values.

In linear systems the null space of the observability matrix shows the nonobservable part of the state. The Fisher information matrix contains information about the relative observability of the estimated state. The singular values of this matrix describe the relative observability of the corresponding part of the estimated state, showing which states can be estimated with good accuracy and which will have large uncertainties.

The information filter [97] maintains the inverse of the covariance matrix - the Fisher information matrix- associated with and extended Kalman filter implementation. The rank of the information matrix at any point in the estimation has a direct link to the rank of the observability matrix of the system. A rank deficient information matrix

indicates that the information along a given axis of the state space is zero and this state is unrecoverable due to partial observability in the system.

Example 4.6

In this example we show simulation results which demonstrate the way in which the observability affects the information in the estimation during SLAM. An extended information filter [97] is used to fuse encoder data (the proprioceptive sensor) of the ground planar vehicle of the Example 4.1 and bearing-only observations using the process and measurement models in (4.11) and (4.12) and using a world-centric state vector as shown in (4.13).

The information matrix allows us to represent complete initial uncertainty in the estimated states as zero information rather than infinite covariance, and provides a better estimator for representing the initial infinite uncertainty of the position of a feature along the line of sight of a single observation. The only remaining issue with the extended information filter for bearing-only SLAM is that the Jacobian of the observation model cannot be evaluated until a sufficiently accurate estimate of the range to the feature is derived from subsequent observations of a feature from different poses. To overcome this, the Jacobians of the process and measurement models are evaluated using the simulated truth data instead. This assumption is obviously not applicable in a real scenario. However, it does not affect the value of the inverse covariance matrix in the information filter. We can therefore examine the value of the inverse covariance matrix as an indication of the errors in the estimation process, under the assumption that linearisation errors in the extended information filter are ignored.

Figure 4.3 illustrates the trajectory taken by the vehicle. The vehicle starts off facing north and is stationary for 10 seconds. There are two observable features, one located directly north of the vehicle and one located to the north east. Both features are within the observation range of the on-board feature sensor for the entire duration of the simulation. After 10 seconds the vehicle moves north with a velocity of 1 *[m/s] for 4 seconds and then turns at* 0.25 *[rad/s] and continues to the east without stopping for*

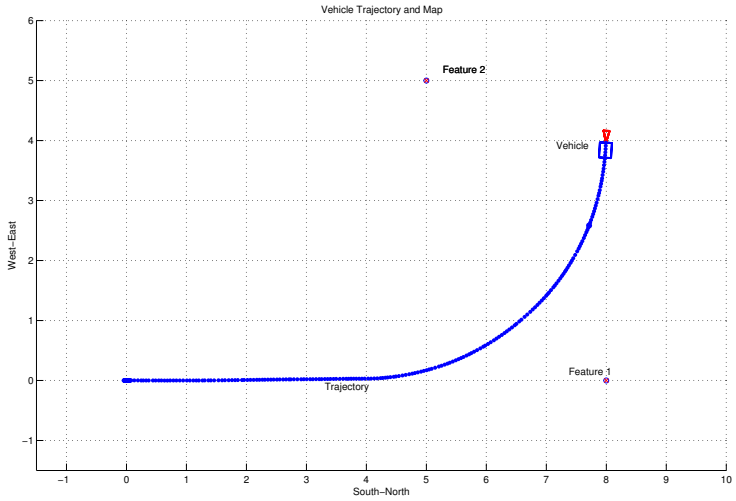


Figure 4.3: Vehicle trajectory and feature map for the simulation.The vehicle is stationary for 10 seconds then moves northerly for 4 seconds and turns and moves to the north-east for another five seconds. There are two features in the map, visible to the vehicle for the entire duration of the simulation.

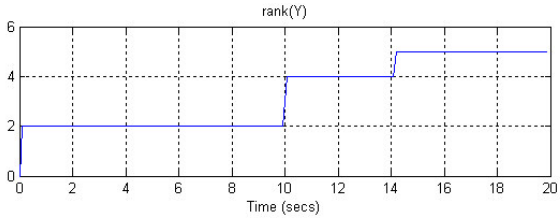


Figure 4.4: Rank of the information matrix. After 14 seconds the information matrix is rank 5 and has a rank deficiency of 2, representing the globally unobservable mode specified in (4.20).

another 6 seconds.

Figure 4.4 shows the rank of the information matrix during the simulation. Since there are two observed features, visible for the entire simulation, there are seven states in the state vector in total. The system model in this case corresponds to that in (4.13). While the vehicle is stationary, only two states are observable corresponding to a combination of the vehicle and feature position and vehicle orientation angle. As the vehicle moves north after 10 seconds, the vehicle's non-zero velocity and the multiple observations of feature 2 from different pose angles result in a rank increase of two in

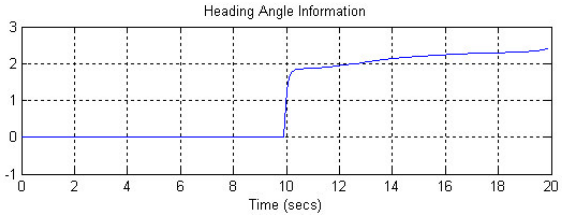


Figure 4.5: Marginalised orientation angle information. While the vehicle is stationary, the orientation angle is unobservable and thus has zero information. When the vehicle begins to move at the 10 second mark, the orientation angle becomes observable and thus the information increases.

the information matrix corresponding to an increase in the number of observable states. The orientation angle is observable and also the relative position of the second feature to the vehicle and the relative east position of the first feature to the second feature but not the relative north position of the first feature due to the vehicle moving along the line of sight to this feature. When the vehicle moves to the north-east after 14 seconds, the rank of information matrix rises to five corresponding to the relative north position of feature 1 w.r.t the vehicle becoming observable. The remaining rank deficiency of two corresponds to the unobservable modes in (4.20).

Figure 4.5 shows the marginalised vehicle orientation angle information[4] *during the simulation. The information about the orientation angle starts at zero and only begins to increase when the orientation angle becomes observable as the vehicle begins to move forward at the 10 second mark.*

Figure 4.6(a) shows the marginalised information from the information matrix corresponding to the modes $\tilde{x}_r + \sum_i \tilde{x}_f^{(i)}$ *and* $\tilde{y}_r + \sum_i \tilde{y}_f^{(i)}$. *These are the unobservable modes shown in (4.20) which cannot be made observable through any number of time segments or any amount of platform maneuvering. The information corresponding to these states thus remains at zero throughout the entire simulation.*

Figures 4.6(b) and 4.6(c) show the marginalised information corresponding to the

[4] Marginalisation of a single state from a multi-dimensional inverse covariance (information) matrix involves computing the Schur complement of the diagonal matrix element of the state of interest (see [47] for details).

state space modes $\tilde{x}_r - \tilde{x}_{1,f}$ and $\tilde{y}_r - \tilde{y}_{1,f}$ (i.e. the relative position of the vehicle w.r.t feature 1) and $\tilde{x}_{1,f} - \tilde{x}_{2,f}$ and $\tilde{y}_{1,f} - \tilde{y}_{2,f}$ (i.e. the relative position of feature 1 w.r.t feature 2). All information about the relative position of features w.r.t the vehicle or to other features is zero (they are unobservable) before the vehicle starts moving. After the 10 second mark when the vehicle starts moving north, information about the relative position of the first feature w.r.t the second feature in the easterly direction starts to increase and it is only after the vehicle starts to move east that the information about the relative position of feature 1 w.r.t both the vehicle and feature 2 on both axes starts to increase as all of these states become observable.

4.4 Conclusion

The purpose of this chapter was to achieve a better understanding of bearing-only SLAM systems from a control theory viewpoint. Our analysis was focused on determining the bases for the nullspace of the observability matrices for a variety of bearing-only SLAM systems. Linear observability analyses were performed by treating some systems as piecewise linear and under the consideration of bearing-only measurements. Moreover, the analysis was complemented with a nonlinear analysis of the most common bearing-only planar vehicle model.

Specifically, the observability analysis indicated the instantaneous unobservable modes in the system and the directions in the state space for which no information is being added over the set of observations. Our results can be used to determine the the set of motions that the vehicle must take in order to result in maximum observability and thus induce constrained error drift in the estimation task.

An example application in which maneuvers for path segments are chosen so as to enhance observability conditions for an inertial system is given in [19] for the case of range and bearing SLAM. An equivalent strategy could be developed using the

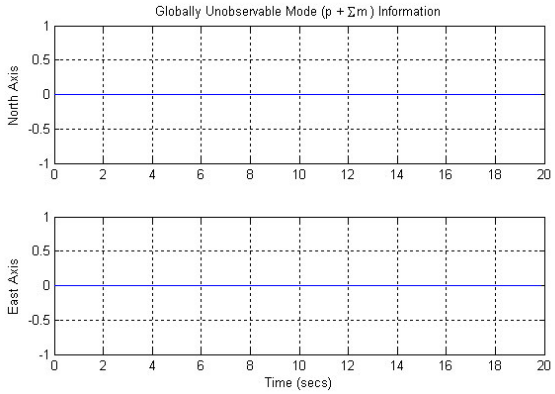
theoretical results in this chapter to the case of a planar vehicle with bearing only sensing.

For the planar vehicle, our analysis showed that the orientation is the only state that can be recovered in one step. In two steps, the sensor-centric model of the planar vehicle performing SLAM with bearing-only measurements is fully observable. We have formalised the notion that triangulation from different positions is needed to fully recover pose and have demonstrated the concept using tools from control theory. As a consequence to these findings, it is important to avoid the case of zero velocity for this case.

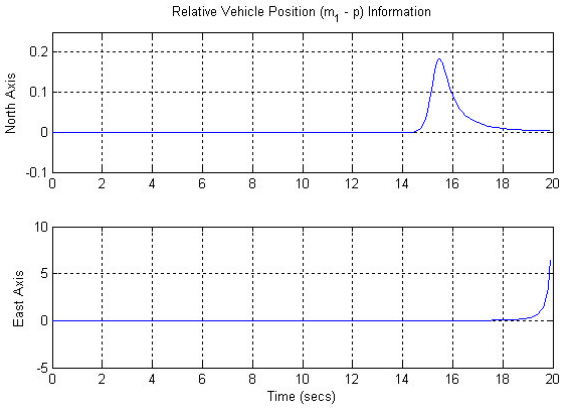
The 6-DOF inertial model becomes a constant velocity model when no acceleration is exerted on the system. When this happens, the system requires a third step to reach the rank condition for the TOM. In the case of zero velocity for the planar vehicle, the system simply becomes unobservable.

The bearing-only constant velocity models are more restrictive in terms of what can be observed within a limited number of segments. For the anchored systems, velocities can be recovered in one segment. The accelerations, inputs to the system, do not affect the observability, but it is clear that zero velocity produces a lack of observability.

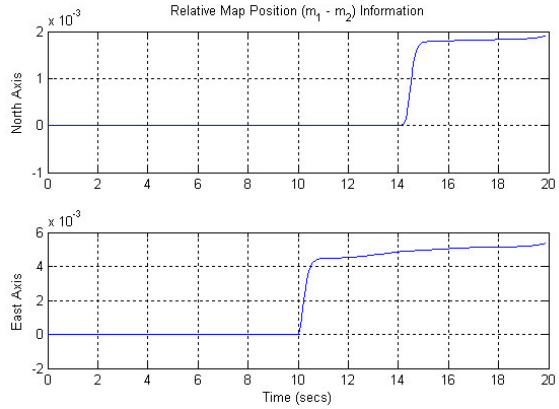
An interesting conclusion of the chapter is that the number of unobservable modes in the world-centric case reaches a lower limit. Adding more time segments to the analysis of any world-centric model is also of little use to estimating the global position of the vehicle or features, as these modes will always be unobservable. The use of anchors or using a sensor-centric approach, in the case of having proprioceptive sensors, allows the system to become fully observable when the appropriate vehicle motions are made.



(a) Modes $\tilde{x}_r + \sum_i \tilde{x}_f^{(i)}$ and $\tilde{y}_r + \sum_i \tilde{y}_f^{(i)}$.



(b) Modes $\tilde{p}_x - \tilde{x}_{1,f}$ and $\tilde{p}_y - \tilde{y}_{1,f}$.



(c) Modes $\tilde{x}_{1,f} - \tilde{x}_{2,f}$ and $\tilde{y}_{1,f} - \tilde{y}_{2,f}$.

Figure 4.6: Marginalised information of modes. a) These modes remain unobservable regardless of any maneuvering by the vehicle; b) these modes only become fully observable when the vehicle traverses laterally to each of the features; c) the easterly component (y-axis) becomes observable when the vehicle begins to move (after 10 seconds) but the northerly component (x-axis) only becomes fully observable when the vehicle traverses laterally to each of the features (after 14 seconds).

Chapter 5

Closed Loop Control

One of the objectives of this thesis is to close the control loop of the estimated system. A block diagram representation of such closed-loop system is shown in Figure 5.1.

In the case of a known map but noisy observations, it is possible to go from A to B with an appropriate control law, i.e. by following a predefined trajectory. However in the case of SLAM, the map is uncertain and the question is whether it is possible to follow a desired trajectory from A to B, while guaranteeing at the same time that the entire vehicle-map system remains stable.

In this chapter, our aim is to show how any such predefined trajectory can be followed as accurately as possible while at the same time a map is being built and the vehicle is kept localised. For any of such strategies however, convergence to the chosen intermediate navigation goals, e.g. predefined paths or locations, should be guaranteed by the generation of the appropriate low-level control commands.

The desired trajectory should come from a higher-level planning strategy. For example, vehicle maneuvers may be chosen to reduce localisation uncertainty [37, 118], or by guiding a vehicle during a SLAM session in order to explore the environment with the goal of minimising the entropy of the map [16, 49].

Once a high-level motion plan is generated, two control strategies are possible: open

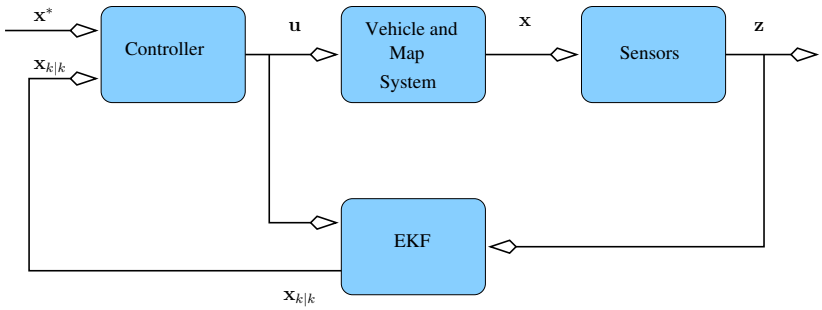


Figure 5.1: Block diagram of the closed loop system for estimation and control in SLAM process.

loop and closed loop. Open loop strategies seek to find a bounded sequence of control inputs, driving the vehicle from an initial position to an arbitrary final position, usually working in conjunction with a motion planner [83, 100]. Feedback control systems are generally more robust to uncertainty and disturbances compared to their open-loop counterparts. For this reason closed-loop control is essential in real mobile robotics because robots and sensors are subject to noise and uncertainty.

In this chapter we formalise the estimation-control approach to SLAM, concentrating only on the feedback closed-loop for the task of stabilising the system along the trajectory. In the literature, the most proximate approach to closing the control loop in SLAM is by incorporating visual servoing techniques as done in [23], or by implementing a PD controller over a trajectory produced by an A* algorithm [134].

Two control strategies are applied to the SLAM system. First in Section 5.1, a regulation control law that can be used to direct the vehicle to a desired location in an optimal manner. This strategy, known as optimal Linear Quadratic Gaussian (LQG) regulation, is obtained by minimising a quadratic performance index, and comes naturally from the Kalman filter derivation, being its dual.

In Section 5.2, we propose a second strategy, a nonlinear control technique called feedback linearisation to follow a predefined trajectory. Each section contains examples with simulations of a nonlinear vehicle model and a camera moving freely in Cartesian

space. The generalisation of the approach to the case of multiple vehicles is presented also in Section 5.2.

5.1 LQG Regulation

In this section, it is shown how to use the optimal vehicle state estimate provided by the Kalman filter to drive the vehicle to a desired location. In control theory, the problem is known as regulation, and, for a linear system with Gaussian noises and a quadratic cost function, an optimal solution exists in the form of the LQG [53].

The performance index is quadratic with respect to the state and the control inputs

$$\mathcal{J} = E\left[\sum_{i=0}^{k-1} \mathbf{x}_i^\top \mathbf{Q}_1 \mathbf{x}_i + \mathbf{u}_i^\top \mathbf{Q}_2 \mathbf{u}_i\right], \tag{5.1}$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are chosen *psd*. The minimisation of this performance index would drive the state $\mathbf{x}$ to zero, although the extension to a general desired nonzero state is straightforward [53].

The performance index in (5.1) can be decomposed into two terms, one for minimising the state estimate and the input , and the second one for minimising the estimation error. Given that the estimation error does not depend on the control input $\mathbf{u}$, it is chosen to minimise the first part of $\mathcal{J}$ only. The control law for such regulator is

$$\mathbf{u}_k = -\mathbf{L}_k \mathbf{x}_{k|k}\,, \tag{5.2}$$

$$\mathbf{L}_k = (\mathbf{Q}_2 + \mathbf{G}_k^\top \mathbf{T}_k \mathbf{G}_k)^{-1} \mathbf{G}_k^\top \mathbf{T}_k \mathbf{F}_k\,, \tag{5.3}$$

$$\mathbf{T}_k = \mathbf{Q}_1 + \mathbf{L}_k^\top \mathbf{Q}_2 \mathbf{L}_k + (\mathbf{F}_k - \mathbf{G}_k \mathbf{L}_k)^\top \mathbf{T}_k (\mathbf{F}_k - \mathbf{G}_k \mathbf{L}_k)\,. \tag{5.4}$$

Separation of the LQG into two parts, the optimal state estimation, and the optimal controller, gives a Kalman filter independent of the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$, which specify the optimal controller. Similarly, the optimal control gain $\mathbf{L}$ does not depend on the

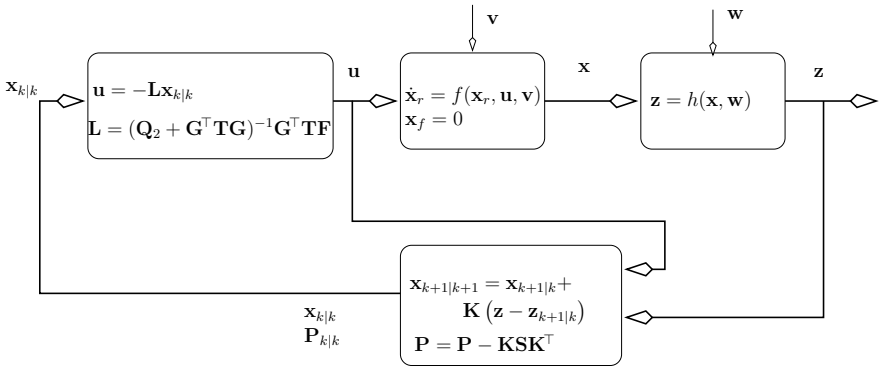


Figure 5.2: Structure of the closed-loop system. The LQG controller and the EKF.

statistics $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ of the random noises. This is referred to as the *Separation principle*.

Furthermore, the Separation principle allows to write the closed loop system dynamics as

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{e}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F} - \mathbf{G}\mathbf{L} & \mathbf{G}\mathbf{L} \\ \mathbf{0} & \mathbf{F} - \mathbf{K}\mathbf{H}\mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{e}_k \end{bmatrix} + \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{G} - \mathbf{K}\mathbf{H}\mathbf{G} & -\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{v}_k \\ \mathbf{w}_k \end{bmatrix}. \tag{5.5}$$

The structure of this stochastic control is shown in Figure 5.2.

The eigenvalues of the closed-loop system are given by those of the state-feedback regulator dynamic $\mathbf{F}-\mathbf{G}\mathbf{L}$ together with those of the state-estimator dynamics $\mathbf{F} - \mathbf{K}\mathbf{H}\mathbf{F}$. The closed loop system is stable when both of these sub-matrices are stable. For a fully observable monobot, it is straightforward to verify that $\mathbf{F} - \mathbf{K}\mathbf{H}\mathbf{F}$ is always stable, and the vehicle part of $\mathbf{F} - \mathbf{G}\mathbf{L}$ is stable for any positive definite $\mathbf{L}$.

Example 5.1

Continuing with the Example 3.4 that of the planar vehicle, we derive LQG regulator using the vehicle and measurement models (3.16) and (3.19). Note that the pose of the

vehicle, and hence, the control point is located off the vehicle axis of rotation in order to avoid kinematic constraints. The control point is chosen at the centre of projection of the sensor, in this particular case a laser range scanner, placed on the front of the vehicle, thus simplifying the measurement model.

In order to show the feasibility of using LQG regulation during SLAM, we simulate an environment with 16 landmarks. The vehicle reference frame is shown in Figure 3.9.

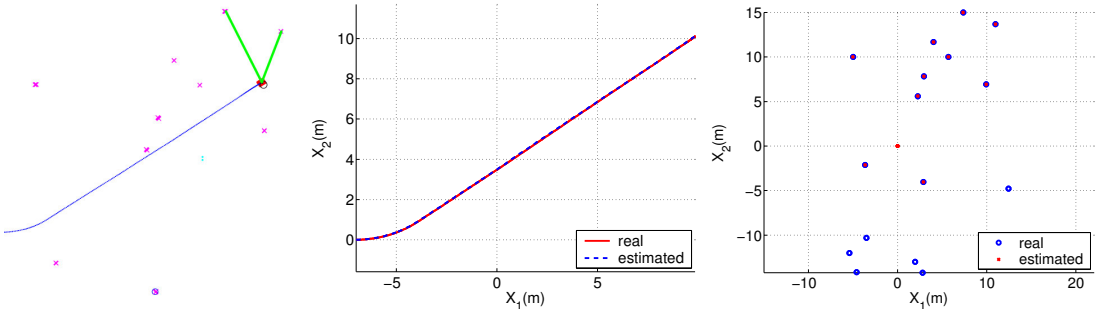
In the plots shown in Figure 5.3 the vehicle is driven with the LQG regulator from an initial location at (-7,0) to the point (10,10), while at the same time building a map of the environment, and using the revised Kalman estimates to recompute the stochastic control gain (for this nonlinear case, there is no guarantee that the control is optimal). Figure 5.4 shows plots of the vehicle state estimate, the estimation error, and the input command. The control law to get to the desired position $\mathbf{p}^* = [10; 10]$ *is*

$$\mathbf{u} = -\mathbf{L}(\mathbf{p}_{k|k} - \mathbf{p}^*) \tag{5.6}$$

where $\mathbf{L}$ *is the gain in (5.3),* $\mathbf{Q}_1$ *and* $\mathbf{Q}_2$ *are chosen psd equal to the identity.*

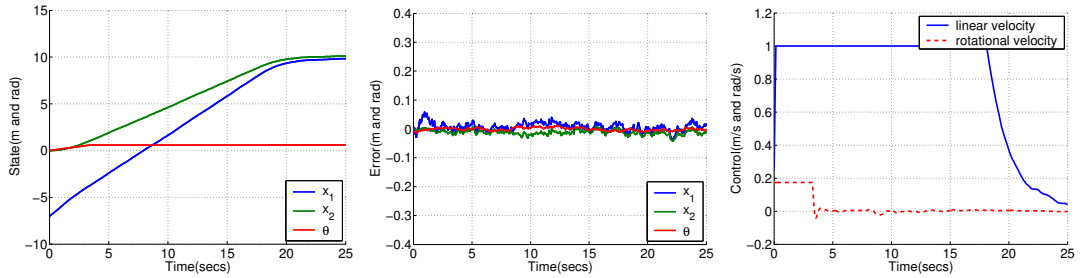
Figure 5.5 shows a runtime plot of the eigenvalues of $\mathbf{F} - \mathbf{GL}$ *evaluated with the plant Jacobians* $\mathbf{F}$ *and* $\mathbf{G}$*. The plot shows the control strategy to be not only optimal by duality from the EKF, but also stable, even when linearisations are used. Notice the three eigenvalues are less than one during the whole simulation time.*

LQG is stable for a controllable vehicle in SLAM. Moreover, having full observability, the filter in SLAM is also stable due to the separation principle. For the linear case we can be sure of this separation, but this is not the case for a nonlinear system.



(a) LQG and SLAM (b) Vehicle path (c) Final map

Figure 5.3: Linear quadratic Gaussian (LQG) regulation and SLAM for a 2D vehicle.



(a) State estimates (b) State estimate error (c) Input commands

Figure 5.4: State estimation and control using Linear Quadratic Regulation.

5.2 Feedback Linearisation

Linear control methods rely on the key assumption of small range operation for the linear model to be valid. When the required operation range is large, a linear controller is likely to perform poorly or can become unstable, because the nonlinearities in the system cannot be properly compensated. Nonlinear controllers, on the other hand, may handle the nonlinearities in large range operation directly.

Given a nonlinear plant to be controlled, we design a nonlinear control law using feedback linearisation in order to follow a desired predefined trajectory. The feedback linearisation approach is commonly used to control nonlinear systems by algebraically transforming the system dynamics into an equivalent model of a simpler form. It

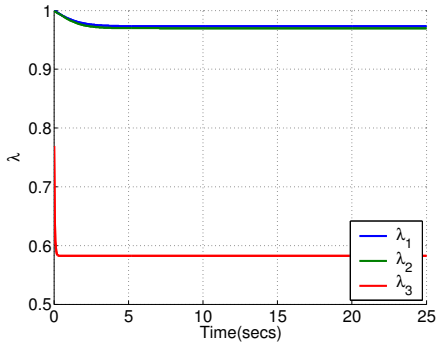


Figure 5.5: Eigenvalues of $\mathbf{F} - \mathbf{GL}$ for an LQG control strategy. Asymptotic stability is guaranteed when these eigenvalues are less than one.

differs from conventional (Jacobian) linearisation in that linearisation is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics. The controller presented here however, avoids nonholonomic kinematic constraints by changing the control point off the axle centre of the vehicle. For a recent treatment to the control of wheeled vehicles in the presence of nonholonomic constraints and estimation uncertainties, see [30].

To generate a direct relationship between the estimate of our state control point $\mathbf{y}_k = \mathbf{x}_{r,k|k}$, and the control input $\mathbf{u}_k$, the system model is rewritten as

$$\mathbf{y}_{k+1} = \mathbf{Q}_r(\mathbf{y}_k)\mathbf{u}_k. \tag{5.7}$$

We choose the control input to be in the form[1]

$$\mathbf{u}_k = \mathbf{Q}_r^{-1}(\mathbf{y}_k)\mathcal{U}\,, \tag{5.8}$$

where $\mathcal{U}$ is the new input to be determined. The nonlinearity in (5.7) is cancelled, and we obtain a simple linear single-integrator relationship between the estimation of the control point and the new input being $\mathcal{U} = \mathbf{y}_{k+1}$. The desired state trajectory or

[1] In the case when the matrix $\mathbf{Q}_r(\mathbf{y}_k)$ is not full rank the appropriate pseudoinverse matrix $(\mathbf{Q}_r^\top \mathbf{Q}_r)^{-1}\mathbf{Q}_r^\top$ can be used.

position is defined as $\mathbf{y}_k^*$.

Choosing the new input $\mathcal{U}$ as

$$\mathcal{U} = \mathbf{y}_{k+1}^* - \mathbf{Q}_2(\mathbf{y}_k - \mathbf{y}_k^*)\,. \tag{5.9}$$

The tracking error is defined as $\mathbf{e}_{c,k} = \mathbf{y}_k - \mathbf{y}_k^*$. Replacing the tracking error in (5.9) and the resulting expression in (5.8) and (5.7), the controlled closed loop system becomes

$$\mathbf{e}_{c,k+1} + \mathbf{Q}_2\mathbf{e}_{c,k} = \mathbf{0}\,, \tag{5.10}$$

being an asymptotically stable error dynamics if $\mathbf{Q}_2$ is chosen *psd*. This control law can be used to stabilise a nonlinear controllable system along a time parameterised trajectory or to a certain point.

Example 5.2

We will now show feedback linearisation control scheme applied to the same motion and measurement models as in Example 5.1. Now, instead of linearizing the system, we consider a linear model with respect to the input.

The closed loop equations of our estimated system are

$$\mathbf{x}_{k+1|k+1} = \mathbf{f}(\mathbf{x}_{k|k})\mathbf{u}_k + \mathbf{K}(\mathbf{z}_{k+1} - \mathbf{H}\mathbf{x}_{k+1|k}) \tag{5.11}$$

$$\mathbf{u}_k = -\mathbf{Q}_r^{-1}(\mathbf{y}_{k+1}^* - \mathbf{Q}_2(\mathbf{y}_k^* - \mathbf{y}_k)) \tag{5.12}$$

where

$$\mathbf{Q}_r = \begin{bmatrix} \cos\theta_k & -l\sin\theta_k \\ \sin\theta_k & l\cos\theta_k \end{bmatrix}, \tag{5.13}$$

and $\mathbf{Q}_2$ is an arbitrary psd constant matrix. Our new $\mathbf{L} = \mathbf{Q}_r^{-1}\mathbf{Q}_2$, by analogy with Example 5.1, depends on the state estimate $\mathbf{x}_{k|k}$, and guarantees convergence to zero for the tracking error $\mathbf{e}_{c,k}$ in finite time.

According to the Separation principle (5.5), asymptotic stability of the system is

guaranteed only for a control gain $\mathbf{L}$ *such that the eigenvalues of the vehicle part of the state in* $\mathbf{F} - \mathbf{GL}$ *are less than one.*

Figure 5.6 presents a simulation for the same planar vehicle following a circular path parameterised in time, while simultaneously performing SLAM. The path is

$$\mathbf{y}_k^* = \begin{bmatrix} r\cos(\omega k) \\ r\sin(\omega k) \end{bmatrix} \tag{5.14}$$

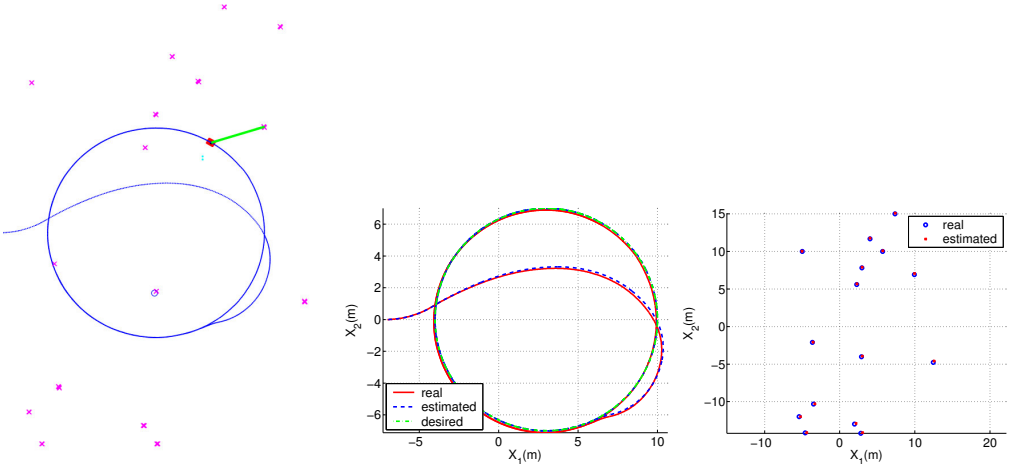
$$\mathbf{y}_{k+1}^* = \begin{bmatrix} -r\sin(\omega k) \\ r\cos(\omega k) \end{bmatrix}, \tag{5.15}$$

where $r = 7$ *[m] is the radius of the circle and* $\omega = \frac{5\pi}{180}$ *[rad* s^{-1}*] is angular velocity of the robot motion.*

The desired trajectory should come from a higher-level planning strategy. But since that is not the scope of this chapter, but to guarantee concurrent tracking and estimation stabilities, simple circular paths are chosen instead.

Figure 5.7 shows plots for the vehicle state estimate, the state estimation error, and the history of control commands. Note in the last plot, that when the motion is initiated, the control law chooses a saturated translational velocity to reach the circular path, settling around a steady-state value 0.6 *[m/s]. Ignoring the uncontrollability of the angular orientation produces a drastic fluctuation of the angular velocity signal during this initial transient interval, then stabilising to the desired angular velocity, set at* 5 *[deg/s].*

Control stability is determined by the eigenvalues of $\mathbf{F} - \mathbf{GL}$*, plotted in Figure 5.8. Note that, when the tracking error is large during the initial transient interval, the error dynamics is not stable. But, once the vehicle approaches the circular path, the controller stabilises. Presumably, by choosing a desired time parameterised trajectory with an initial pose coincident with the initial pose estimate, would suffice to solve this phenomenon.*

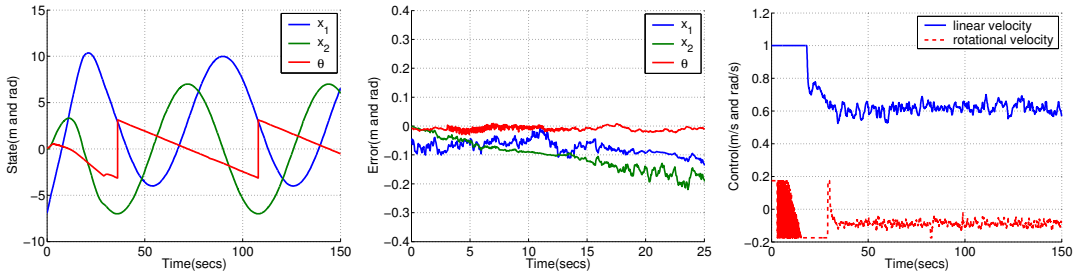


(a) Feedback linearisation control and SLAM (b) Vehicle path (c) Final map

Figure 5.6: Feedback linearisation strategy for control and SLAM.

Even with such a stable control law, given the kinematics constraint of the vehicle model used, the entire vehicle pose $[x_1, x_2, \theta]$ *cannot be stabilised (see [17]). For this reason, we have decided to let* $\mathbf{y}_k = [x_1, x_2]^\top$ *be the Cartesian coordinates of the vehicle location only, and not to try to optimally control the vehicle orientation. The control point is off the rotation axis. The matrix* $\mathbf{Q}_1$ *becomes full rank and the pseudoinverse in (5.8) is no longer necessary. The simulations show, that by controlling the vehicle position only, and letting the vehicle orientation be a free variable, after an initial transient interval, the predefined time parameterised trajectory can still be accurately followed with this type of vehicle.*

For a wheeled mobile robot in SLAM having fully observability, stability is guaranteed. Moreover, a stable tracking error dynamics is achieve by using a feedback linearisation control law.



(a) State estimates (b) State estimate error (c) Input commands

Figure 5.7: State estimation and control using feedback linearisation.

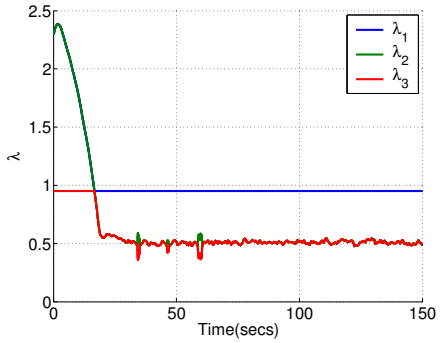


Figure 5.8: Eigenvalues of $\mathbf{F} - \mathbf{GL}$ for a feedback linearisation control strategy. Asymptotic stability is guaranteed when these eigenvalues are less than one.

5.2.1 Extension to MultiRobot Control

The extension to multirobot system is direct, the SLAM and control equations are derived not just for one, but for a group of vehicles. Following the same notation, the motion of the entire set of vehicles and the map measurements are governed by the discrete time stochastic state transition model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{5.16}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \,. \tag{5.17}$$

The state $\mathbf{x}_k = [\mathbf{x}_{r_1,k}^\top, \ldots, \mathbf{x}_{r_t,k}^\top, \mathbf{x}_{f(1)}^\top, \ldots, \mathbf{x}_{f(n)}^\top]^\top$ contains the pose of the vehicles $\mathbf{x}_{r_1}, \ldots, \mathbf{x}_{r_t}$ at time step k, and a vector of stationary map features $\mathbf{x}_{f(1)}, \ldots, \mathbf{x}_{f(n)}$.

The input vector $\mathbf{u}_k = [\mathbf{u}_{r_1,k}^\top, \ldots, \mathbf{u}_{r_t,k}^\top]^\top$ is a multi-vehicle control command, $\mathbf{v}_k = [\mathbf{v}_{r_1,k}^\top, \ldots, \mathbf{v}_{r_t,k}^\top]^\top$ is the plant noise, $\mathbf{w}_k = [\mathbf{w}_{f^{(1)},k}^\top, \ldots, \mathbf{w}_{f^{(n)},k}^\top]^\top$ is the sensor noise, and both are Gaussian random vectors with zero mean and block diagonal covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Similar to the case of a single vehicle, an optimal estimate of (5.16), in a least squares sense, is given by the expression

$$\mathbf{x}_{k+1|k+1} = \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0}) + \mathbf{K}(\mathbf{z}_{k+1} - \mathbf{h}(\mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0}))) \tag{5.18}$$

with covariance

$$\mathbf{P}_{k+1|k+1} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top - \mathbf{K}[\mathbf{H}[\mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top]\mathbf{H}^\top + \mathbf{R}]\mathbf{K}^\top . \tag{5.19}$$

The Jacobians $\mathbf{F}$ and $\mathbf{G}$ represent first order linearisations of the multi-vehicle model with respect to the state and the plant noise. Similarly, the Jacobian $\mathbf{H}$ contains first order linearisations of the measurement model with respect to the entire state.

Example 5.3

We apply now the same feedback linearisation approach to two robots in an environment equivalent to that in Example 5.2. The system dynamics (5.16) must be described in controllability canonical form. That is, linear with respect to the input $\mathbf{u}(k)$.

$$\mathbf{y}_{k+1} = \mathbf{Q}_r(\mathbf{y}_k)(\mathbf{u}_k + \mathbf{v}_k) \tag{5.20}$$

with $\mathbf{y}_k = [x_{r_1.k}, y_{r_1,k}, \ldots, x_{r_t,k}, y_{r_t,k}]^\top$ *only the multirobot location part of the state vector.*

Choosing the control law as

$$\mathbf{u}_k = \mathbf{Q}_r^{-1}\left(\mathbf{y}_{k+1}^* - Q_2(\mathbf{y}_k^* - \mathbf{y}_k)\right) , \tag{5.21}$$

the closed loop equations for the multirobot state and multirobot state estimate error

are,

$$\mathbf{y}_{k+1} = \mathbf{y}^*_{k+1} - \mathbf{Q}_2(\mathbf{y}_k - \mathbf{y}^*_k) + \mathbf{Q}_r\mathbf{v}_k \tag{5.22}$$

$$\mathbf{x}_{k+1|k+1} = (\mathbf{F} - \mathbf{KHF})\mathbf{x}_{k|k} + (\mathbf{G} - \mathbf{KHG})\mathbf{v}_k - \mathbf{K}\mathbf{w}_k\,. \tag{5.23}$$

As in the single-vehicle case, the estimation and the control problems separable fullfill separation principle. The eigenvalues of the closed-loop system are given by those of the feedback dynamics $\mathbf{Q}_r(\mathbf{y}_k)$*, together with those of the state estimator dynamics* $\mathbf{F} - \mathbf{KHF}$*. The closed loop system is stable only when both matrices are stable. We have designed* $\mathbf{Q}_2$ *for a stable closed-loop system, and for a fully observable estimation problem.*

Figure 5.9 shows simulation results for a pair of robots simultaneously following two time parameterised circular paths, while performing SLAM. The objective is to track the desired path as accurately as possible.

Figure 5.10 shows plots for the vehicle state estimates, the state estimation error, and the history of control commands.

Finally, Figure 5.11 shows the asymptotic landmark state estimate trace covariances. The plot is typical of a SLAM map: the landmark localisation uncertainties decrease until reaching an asymptotic minimum and specifically shows the decrease in all landmark localisation uncertainties as the algorithm proceeds, showing asymptotic convergence of the estimation part of the problem.

5.2.2 3D Single Camera Control

One of the main objectives of this thesis considers the problem of navigation using a single camera. From the point of view of SLAM, this problem is more difficult than the ones we have considered so far because the world is three-dimensional and observations are bearing only. The lack of observability inherent to this system shown in Chapter 4,

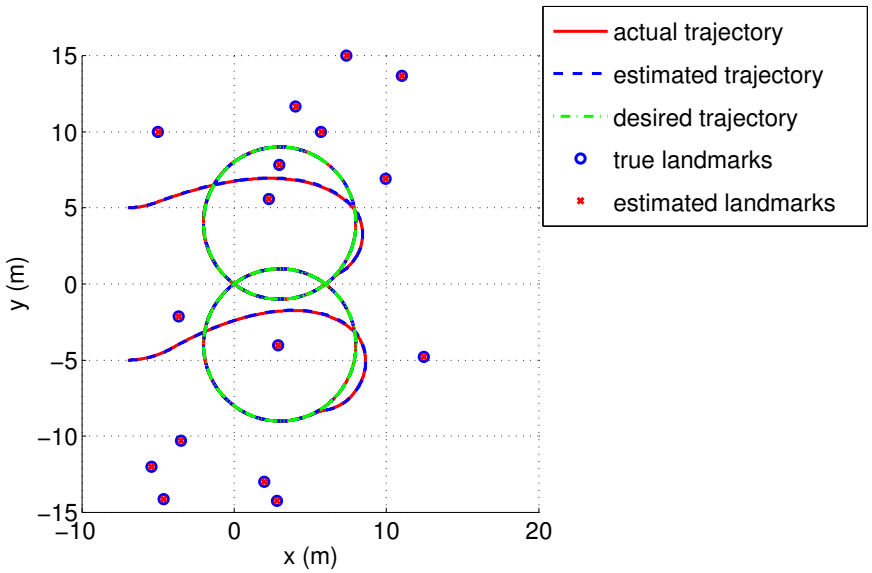


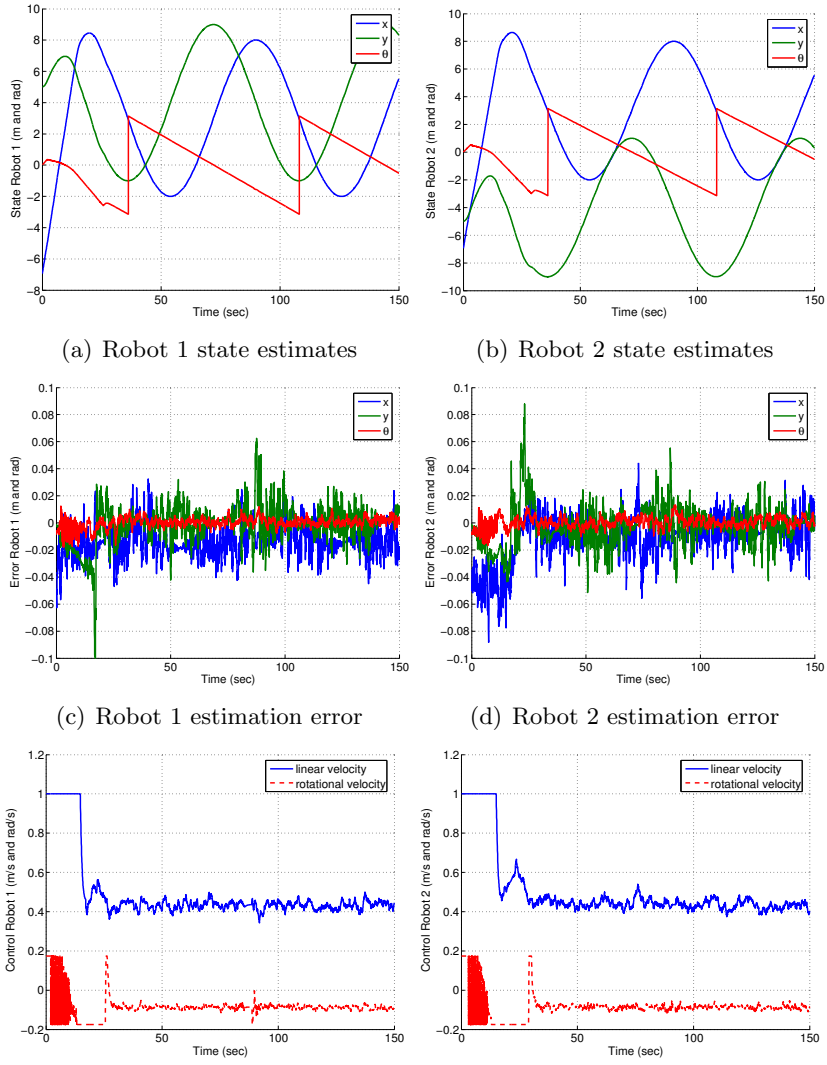
Figure 5.9: Simultaneous multirobot localisation, control, and mapping following circular paths.

makes inminent the use of an active control.

Sim in [118] analysed a family of parameterised in time trajectories for a 2D mobile robot with a bearing-only sensor to minimise the uncertainty meanwhile during exploration. Non constant motion allows this kind of systems to have good estimator behaviour. Moreover, bearing-only measurements produce uncertainty along the axis of the camera and perpendicular motion with respect to this axis produce larger reductions in uncertainty than motions along the sight of the camera. Another issue well known in SLAM is that some environment loops should to be closed from time to time. For such systems it is mandatory to do this early on so as to keep the variances bounded.

Example 5.4

Let us consider now a single camera moving in a a typical 3D human environment, mapping visual features with minimal prior information about motion dynamics, for instance using a constant velocity model. Our aim is to localise the sensor and build a



(a) Robot 1 state estimates

(b) Robot 2 state estimates

(c) Robot 1 estimation error

(d) Robot 2 estimation error

(e) Robot 1 control command

(f) Robot 2 control command

Figure 5.10: State estimation (vehicle pose and 2D map) and control using feedback linearisation to follow a circular path.

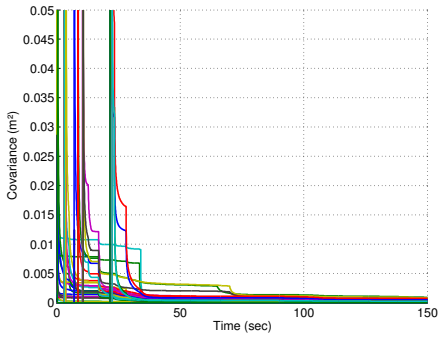


Figure 5.11: Landmark trace covariances of the circular path for one 2D vehicle.

feature map by computing the appropriate control input in order to follow any predefined path. This is another nonlinear system that can be controlled using separability during estimation and control.

The discrete constant velocity motion model is,

$$\mathbf{x}_{v,k+1} = \begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{q}_{k+1} \\ \boldsymbol{\upsilon}_{k+1} \\ \boldsymbol{\omega}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_k + (\boldsymbol{\upsilon}_k + \mathbf{f}_k \Delta t)\Delta t \\ \boldsymbol{\mathcal{P}} \mathbf{q}_k \\ \boldsymbol{\upsilon}_k + \mathbf{f}_k \Delta t \\ \boldsymbol{\omega}_k + \boldsymbol{\alpha}_k \Delta t \end{bmatrix}, \tag{5.24}$$

where $\mathbf{p}$, $\mathbf{q}$, $\boldsymbol{\upsilon}$, $\boldsymbol{\omega}$ *are the position, the orientation represented in quaternions, the linear velocity and the angular velocity respectively of the camera. The input to the system are the accelerations linear* $\mathbf{f}_k$ *and angular* $\boldsymbol{\alpha}_k$*. Notice the orientation is represented in quaternions, a similar motion model is used in Example 4.4, but with a different representation for the orientation (Euler angles).*

Suffice to say that $\mathbf{p} = [x, y, z]^\top$ *and* $\mathbf{q} = [q_0, q_1, q_2, q_3]^\top$ *denote the pose (three states for position and four for orientation using a unit norm quaternion representation), and* $\boldsymbol{\upsilon} = [\upsilon_x, \upsilon_y, \upsilon_z]^\top$ *and* $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ *denote the linear and angular velocities, respectively, corrupted by zero mean normally distributed linear and angular accelerations*

$\mathbf{f} = [f_x, f_y, f_z]^\top$, *and* $\boldsymbol{\alpha} = [\alpha_x, \alpha_y, \alpha_z]^\top$. *The quaternion transition matrix is*

$$\boldsymbol{\mathcal{P}} = \cos\left(\frac{\Delta t\|\boldsymbol{\Omega}\|}{2}\right)\mathbf{I} + \frac{2}{\|\boldsymbol{\Omega}\|}\sin\left(\frac{\Delta t\|\boldsymbol{\Omega}\|}{2}\right)\boldsymbol{\Omega}_\times\,, \tag{5.25}$$

with $\boldsymbol{\Omega} = [0, \omega_x, \omega_y, \omega_z]^\top$ *the angular velocity vector expressed in quaternion form, and* $\boldsymbol{\Omega}_\times$ *its skew-symmetric matrix representation.*

The redundancy in the quaternion representation is removed by a $||\mathbf{q}|| = 1$ *normalisation at each update, accompanied by the corresponding Jacobian modification.*

We assume a measurement model that makes bearing observations to three-dimensions point features. Image projection priors are estimated with a full perspective wide angle camera

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - u_c/\sqrt{d} \\ v_0 - v_c/\sqrt{d} \end{bmatrix} \tag{5.26}$$

where the position of a 3D map point is first transformed into the camera frame $\mathbf{x}_f^{c,i} = [x^c, y^c, z^c]^\top = \mathbf{C}(\mathbf{x}_f^i - \mathbf{p})$, *with* $\mathbf{C}$ *the rotation matrix equivalent of* $\mathbf{q}$, *and*

$$\begin{aligned} u_c &= f k_u x^c/z^c \\ v_c &= f k_v y^c/z^c\,. \end{aligned} \tag{5.27}$$

The radial distortion term is $d = 1 + K_d(u_c^2 + v_c^2)$, *and the intrinsic calibration of the camera is known; focal distance* f, *principal point* (u_0, v_0), *pixel densities* k_u *and* k_v, *and radial distortion parameter* K_d.

We propose to use feedback linearisation approach to control this camera moving in a 3D environment in order to stabilise along any desired path parameterised in time. For instance, the camera could be mounted on any given mobile platform with no motion constraints in the cartesian space and it should be able to track such path.

One of the most stable paths we found was a circle path which returns periodically

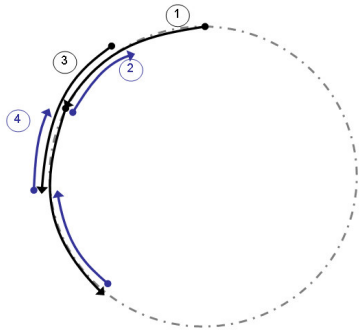


Figure 5.12: Desired trajectory for the vehicle. Back and forth circular path with $r = 1.5$ and $\omega = \frac{15\pi}{180}$.

as it is shown in Figure 5.12, the curve expressed in terms of the time is,

$$\mathbf{y}_k^* = \begin{pmatrix} r\sin\left(\omega_y^*\left(\sin\left(2k\right) - \frac{2k}{\pi}\right)\right) \\ 0 \\ -r\cos\left(\omega_y^*\left(\sin\left(2k\right) - \frac{2k}{\pi}\right)\right) \\ 0 \\ \omega_y^*\left(\sin\left(2k\right) - \frac{2k}{\pi}\right) \\ 0 \end{pmatrix}, \tag{5.28}$$

where r being the radio of the circle is constant and ω_y^* is a constant value that depends on the frequency. Note $\boldsymbol{\omega}^* = [0, \omega_y^*, 0]^\top$. Differentiating $\mathbf{y}_k^*$ with respect to the time gives the desired velocities $\mathbf{y}_{k+1}^*$ of the system.

We choose the control point as the camera position $\mathbf{y}_k = \mathbf{p}_{k|k}$. Note the system (5.24) is already linear with respect to the input in that case $\mathbf{Q}_r = \mathbf{I}$, therefore $\mathbf{u} = \mathcal{U}$.

The aim is to stabilise the system along the desired trajectory (5.28), consequently the proposed control is

$$\begin{bmatrix} \mathbf{f} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} -\mathbf{Q}_{2v}\left(\boldsymbol{v}_{k|k} - \mathbf{y}_k^*\right) - \mathbf{Q}_{2p}\left(\mathbf{p}_{k|k} - \mathbf{y}_k^*\right) \\ \boldsymbol{\alpha}^* - \mathbf{Q}_{2\omega}\left(\boldsymbol{\omega}_{k|k} - \boldsymbol{\omega}^*\right) \end{bmatrix} \tag{5.29}$$

where $\mathbf{Q}_{2v}$, $\mathbf{Q}_{2p}$ *and* $\mathbf{Q}_{2\omega}$ *are* psd *constant matrices for tuning the controller and for providing with it the appropriate units.*

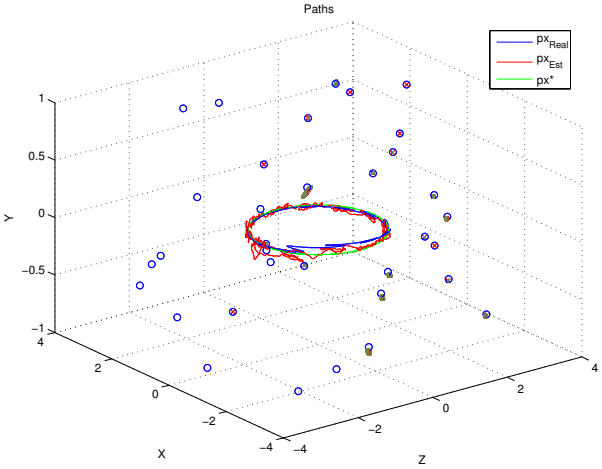
Our control law can regulate the system to a certain position or along a given trajectory. Moreover, this low-level control law could even be used to evaluate action outcomes given a desired position, as we will see in the following chapter.

To show how the proposed control law stabilises the estimated system along the desired trajectory we simulated a camera system moving freely in Cartesian space, using 3D points to localise itself while performing SLAM.

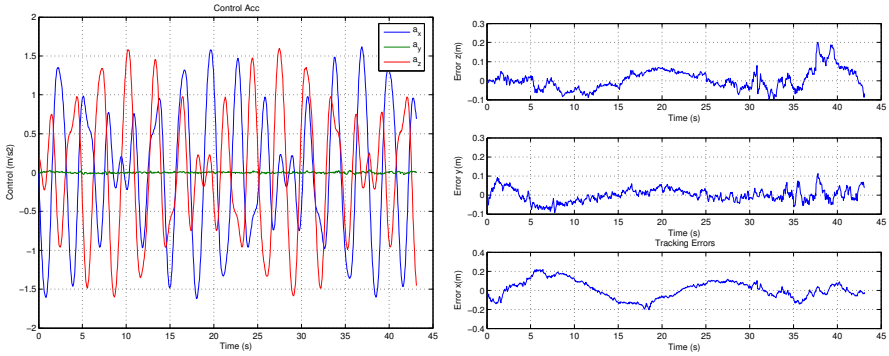
In order to get an idea how the real system would behave, we developed a simulated environment in Matlab/Simulink. This environment modeled the camera translating and rotating freely in 3D according to the model (5.24), 30 3D points are distributed uniformly around the whole space and they are projected into the 2D camera view using the full perspective wide-angle camera model in (5.26). The EKF is used to estimate the camera pose and the landmark positions. A separate module is in charge of computing the input to the system, the control commands. *This last module uses the estimated state as the input. In Appendix B is described more in detail the simulated setting.*

The proposed parameterisation in time trajectory is chosen taking into account the fact that intuitively a single camera SLAM system should close short loops in Cartesian space from time to time to remain localised and consistent, i.e. the estimation should be inside of the uncertainty represented by the covariance, and to must be able at the end, to reduce uncertainty.

Figure 5.13 shows a simulation of the monocular 6 degree of freedom (DOF) system in which the trajectory followed by the camera is a circle of radius 1.5 *[m] and the resulting feature map for a room of size* $6 \times 2 \times 8$ *[m]. Figure 5.13 a) shows the desired and estimated camera trajectory, Figure 5.13 b) shows the accelerations produced by the controller and 5.13 c) the tracking path errors, which indicate difference between desired and estimated 3D positions. For this simulation, the initial estimated pose is very close to the real value but with some uncertainty associated.*



(a) Back-and-Forth circling path



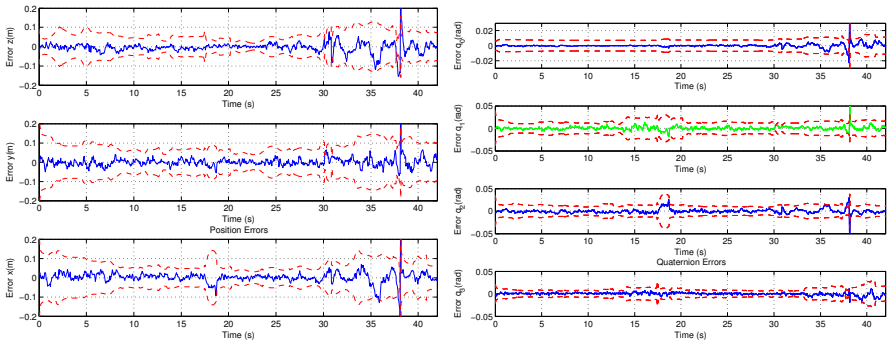
(b) Control Signals

(c) Position tracking error

Figure 5.13: State estimation and control for single camera SLAM. Estimated and camera desired trajectory, control signals and tracking position errors.

In Figure 5.14 the estimation errors and their covariances are shown. Notice that covariance is quite variant, and that closing short loops in Cartesian space is crucial to reduce it.

It should be stressed this SLAM system does not consider any information from odometry or any other sensor than a camera. This is the reason that enables such a system to be mounted in any mobile platform.



(a) Position estimation error (b) Orientation estimation error

Figure 5.14: State estimation for single camera SLAM. Estimation errors for camera position and orientation bounded for the 2σ variance.

5.3 Conclusion

Assuming separability between optimal state estimation and regulation, we have been able to present two vehicle control strategies that do not affect the estimation performance of a fully observable EKF based SLAM: an optimal Linear Quadratic Gaussian regulation technique dual of the Kalman filter, and a feedback linearisation control law. The first one law reaches a desired location in finite time while minimising a performance index related with the input and the estimated state. The second one guarantes asymptotical stability for closed-loop system tracking a time parameterised desired trajectory. An extra control strategy using feedaback linearisation is presented for the case of a single camera moving freely in Cartesian space.

The feasibility of using both approaches was validated with simulation results. In order to avoid the initial transient performance of the forward linearisation control strategy, the effects of the kinematics constraint of the chosen vehicle model should be further investigated.

It is important to stress out that the SLAM system must be fully observable in order to be able to use the estimates as input to any type of controller. Then, both estimation and control can be decoupled and standard techniques such as the ones

used here, Kalman filtering for estimation, and feedback linearisation for control, are plausible for closing the perception-action-loop even in multirobot SLAM. However, separability is only guaranteed for linear Gaussian systems, for nonlinear systems there exists some influence of the control into the estimation.

Another important remark we must mention is that because of the separability between estimation and control, a low level control law, as the one proposed in this chapter, is not able to improve the estimation by itself, i.e. gain observability. To do that, it is necessary to plan the trajectories avoiding unobservable direction as is shown in Chapter 4 or to use a high-level metric to gain information as it will be shown in the next chapter. Nevertheless, the control laws influence the estimates, and the estimates influence the control laws, making the coupled stability of these interacting systems a fundamental issue.

Chapter 6

Action Selection and Experiments in Visual Navigation

As was shown in the previous chapter a low level control law cannot improve the estimated system because of the separability principle. In general, for linear stochastic systems, the feedback control strategy has no probing effect in the sense of the mean-square estimation error. However, for any nonlinear system, as the planar vehicle, the choice of the feedback control can have some effect on estimation, but a higher-level strategy is still required to reduce uncertainty in the system. A block diagram representation of the closed-loop system is shown in Figure 6.1.

One of the objectives of this thesis consists of the guidance of an autonomous mobile robot with a single camera over unknown uneven environments. In the literature it has mostly been considered the case where the vehicle is moving on flat ground-plane. This is a reasonable assumption to make in many indoor scenarios. Nevertheless, most outdoor environments and more challenging indoor environments present non-flat terrains that are not usually considered by motion models during the control and estimation processes. Davison and Kita addressed this problem using a stereo vision system in [38], presenting a SLAM system for a wheeled robot navigating autonomously

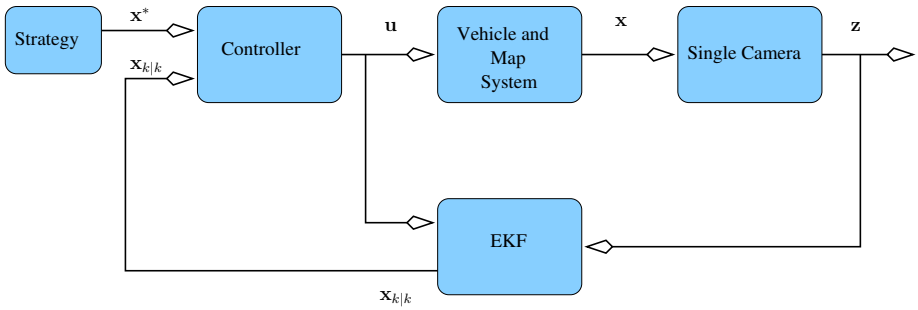


Figure 6.1: Block diagram of the closed loop system for estimation and control in Visual SLAM process.

in undulating terrain.

Another problem that makes the task of guiding a vehicle with a single camera quite difficult, is the limited knowledge about of the environment and the lack of information of depth from the type of sensor used. Estimating the position and velocities of the system while a map of landmarks is being constructed with the measurements provided a single camera requires the appropriate motion command. This command is necessary not only to explore the environment, but also to maintain the uncertainty in the estimation error bounded, i.e. reducing entropy. Given the probabilistic nature of the Bayesian approach to the solution of the SLAM problem, entropy reduction has recently gained popularity as a map building strategy for driving a robot during a SLAM session in order to minimise uncertainty [16, 49, 117].

In this chapter we are interested in the real-time estimation *and control* of a single camera's motion, moving in 3D in normal human environments and in non-flat mobile robot environments, mapping visual features without using a proprioceptive sensors. Localise the sensor and build a feature map by computing the appropriate control actions in order to improve overall system estimation is the goal.

Two main experiments are considered in this chapter. The first one is an on-line implementation for a single camera SLAM system that extends the work of Davison [34] by adding *control* to his otherwise *passive* monocular SLAM system. Given the real-

time characteristics of the visual SLAM system used, fast and efficient action evaluation is of utmost importance. Fortunately enough, the elements needed to validate the quality of actions with respect to entropy reduction are readily available from the SLAM priors [35], and, by making enough implementation adaptations, we are able to evaluate in real-time the value of a reasonably large number of actions.

In the second experiment a wide-angle camera is mounted on a vehicle navigating in uneven terrains. The aim is to compute the appropriate control command for mobile robot after choosing the best action according to the information gain. The expected information gain is evaluated propagating a particular action using the constrained motion model proposed in this chapter, with the advantage that this model considers not only the non-holonomic constraints of the vehicle, but the slope of the terrain as well.

Actions belong to a discrete set (e.g.. go forward, go left, go up, turn right, etc.), and the particular movement chosen is the one that maximises the mutual information gain between posterior states and measurements. Two metrics are presented to evaluate the set of actions that consider information gain. In Section 6.1 both mutual and Fisher information metrics are briefly explained. Section 6.2 presents our chosen control strategy addressing the two different experiments.

Section 6.3 presents the case of a camera that could be mounted in any given platform, because a constant velocity motion model is used. The problem consists of action-decision of a single, for instance hand-held, camera performing SLAM at video rate with generic 6DOF motion. Simulation results are shown comparing the three different approaches, constant angular velocity and the two metrics, to control the direction of gaze. The experimental setup and the results, where a GUI feeds-back motion commands to the user, are presented in a subsection.

In Section 6.4 we consider the experiment of the vehicle moving on uneven 3D terrain. We derive a different motion model for the vehicle considering the 3D position and the 3D orientation. The control strategy is illustrated through simulations. Lastly,

we present the results of real-time experiments, where velocity increments are applied to the vehicle navigating in non-flat terrains.

6.1 Metrics for Information Gain

Probabilities and log-likelihoods are defined on states or measurements. However, it is often valuable to also measure the amount of *information* contained in a given probability distribution. A formal mathematical description of information provides a measure of the compactness of a probability distribution on a state space; if a probability distribution is spread evenly across many states, or conversely if a probability distribution is highly peaked on a few states. Information is thus a function of the probability distribution, rather than the underlying state.

We believe that information measures can play an important role in designing the control strategy for navigation with only one camera. Two probabilistic measures are here evaluated for action selection while performing estimation; the Shannon information (or entropy) and the Fisher information. Both Shannon and Fisher information characterise the information contents of probability densities. These measures are related by the log-likelihood function. Entropy (Shannon information) is related to the volume of a typical set containing a specified probability mass and the Fisher information is related to the surface area of this typical set.

6.1.1 Mutual Information

Entropy is a measure of uncertainty; that is, it is a measure of how much randomness there is in the state estimate. Entropy, in the case of discrete variables, is defined as

$$H(X) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) \,, \tag{6.1}$$

which, for our case where $p(\mathbf{x})$ is a n-variable Gaussian distribution, reduces to

$$H(X) = \frac{1}{2}\log((2\pi)^n|\mathbf{P}|)\,. \tag{6.2}$$

Consider the following two random vectors: the state $\mathbf{x}$, and the i-th measurement $\mathbf{z}_i$. Their joint entropy is simply

$$H(X,Z) = -\sum_{\mathbf{x},\mathbf{z}_i} p(\mathbf{x},\mathbf{z}_i)\log p(\mathbf{x},\mathbf{z}_i) \tag{6.3}$$

and, their conditional entropy, the uncertainty about $\mathbf{x}$, given the knowledge in $\mathbf{z}_i$, is

$$H(X|Z) = \sum_{\mathbf{x},\mathbf{z}_i} p(\mathbf{x},\mathbf{z}_i)\log p(\mathbf{x}|\mathbf{z}_i)\,. \tag{6.4}$$

Doing some math,

$$H(X|Z) = H(X,Z) - H(Z)\,, \tag{6.5}$$

the difference between joint state and measurements entropy and the entropy of the same measurements.

The mutual information is defined as the difference between the entropy of a state and the entropy of the same state conditioned on the outcome of a second experiment, in this case the set of measurements,

$$I(X;Z) = H(X) - H(X|Z) \tag{6.6}$$

which, for our Gaussian multivariate case, evaluates to

$$\begin{aligned} I(X;Z) &= \frac{1}{2}\log\left(\frac{|\mathbf{P}_\mathbf{x}|}{|\mathbf{P}_\mathbf{x} - \mathbf{P}_\mathbf{xz}\mathbf{P}_\mathbf{z}^{-1}\mathbf{P}_\mathbf{xz}^\top|}\right) \\ &= \frac{1}{2}\log\left(\frac{|\mathbf{P}_{k+1|k}|}{|\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|k}\mathbf{H}_i^\top\mathbf{S}_i^{-1}\mathbf{H}_i\mathbf{P}_{k+1|k}^\top|}\right) \\ &= \frac{1}{2}\left(\log|\mathbf{P}_{k+1|k}| - \log|\mathbf{P}_{k+1|k+1}|\right)\,. \end{aligned}$$

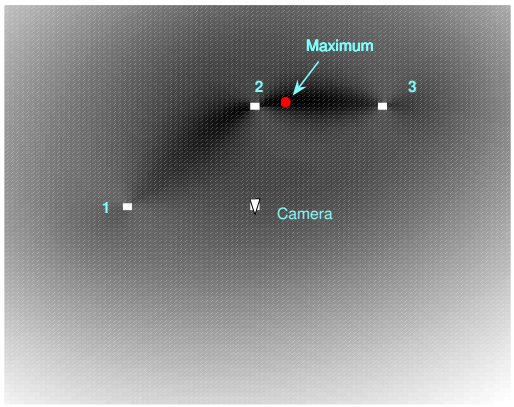


Figure 6.2: Maximisation of mutual information for the evaluation of motion commands. A simple 2D camera is located at the centre of the plot, and a decision where to move must be taken as a function of the pose and landmarks states, and the expected measurements. Three landmarks are located to its left, front, and right-front. Moving to the location in between landmarks 2 and 3 maximises the mutual information between the SLAM prior and the measurements for this particular example.

Mutual information is always positive; it is not possible to lose infomation through knowledge of the outcome of other experiments. This information metric is an *average* of what we would expect to gain before the actual value of $\mathbf{z}$ is known. Thus, in choosing a maximally mutually informative motion command, we are maximising the difference between prior and posterior entropies [92]. In other words, we are choosing the motion command that most reduces the uncertainty of $\mathbf{x}$ due to the knowledge of $\mathbf{z}$ as a result of a particular action. To exemplify this, Figure 6.2 shows the directions maximising the mutual information for a simple 2D camera and 3 landmarks.

Note that the use of mutual information only makes sense prior to reaching full correlation. In SLAM, $|\mathbf{P}_{k|k}|$ tends asymptotically to zero, at which point the map becomes fully correlated and there is nothing else the camera motions can do to improve the estimates of the landmark. From then on, entropy can still be used to decide what actions to take to reduce the camera's own uncertainty, and this can be done just by replacing $\mathbf{x}$ with $\mathbf{x}_r$ from the above discussion.

6.1.2 Fisher Information

A different measure of information commonly used in probabilistic modelling and estimation is Fisher information. Fisher information describes the information content about values of $\mathbf{x}$ contained in the distribution $p(\mathbf{z}_i|\mathbf{x})$. The Fisher information is defined as

$$\boldsymbol{\mathcal{F}}(\mathbf{x}) = E\left(\frac{\partial^2}{\partial \mathbf{x}^2} \log p(\mathbf{z}_i|\mathbf{x})\right) |_{\mathbf{x}=\mathbf{x}_0} \ . \tag{6.7}$$

Under the Gaussian assumption for sensor and platform noises, the minimisation of the least squares criteria (the KF) is equivalent to the maximisation of a likelihood function $p(\mathbf{z}_i|\mathbf{x})$ given the set of measurements Z^k, that is, the maximisation of the joint *pdf* of the entire history of observations, which for the case of SLAM evaluates to

$$\boldsymbol{\mathcal{F}} = \sum \mathbf{H}^\top \mathbf{S}^{-1} \mathbf{H} \, . \tag{6.8}$$

It should be noticed that, in the limit Fisher information is $\mathbf{P}^{-1}$.

The information for the reconstruction of the state contributed by the set of measurements at each iteration is contained in $\mathbf{H}^\top \mathbf{S}^{-1} \mathbf{H}$. The eigenvalues λ_j of this contribution to $\boldsymbol{\mathcal{F}}$ show which linear combinations of the states can be estimated with good accuracy and which will have large uncertainties from the coming measurements. It also shows which linear combinations of states are unobservable (see Chapter 4 for more details). When one dimension of $\boldsymbol{\mathcal{F}}$ has a very small eigenvalue, for example for information along the line of sight, the outcome is not a reliable measure of the elongation of the information hyperellipsoid, as it collapses the surface area to zero.

Under a Fisher information motion strategy, maximally informative actions move the robot as close as possible to the landmarks under observation. We do not want to move towards them, but only to orient towards them. Our idea of using the Fisher information is only to fixate our camera to those most uncertain landmarks, and use the change in entropy to select motion commands.

6.2 Action Selection

The objective is to move the camera in the direction that maximally reduces state estimation uncertainty by taking into account the mutual information between states and measurements maximising the information that can be acquired for any given action. Because of time restrictions actions cannot be evaluated at the same sample time, and it must be computed different priors for all actions. In such a case, computing the mutual information seems to be a better choice compared to the evaluation of entropy only.

The aim is to optimise both the localisation of the sensor and building of the feature map by computing the most appropriate control actions or movements. Getting the optimal action over all possible solutions would require an action selection algorithm based on multi-step look-ahead like the one presented by Huang *et al* in [64]. Because of the expensive complexity cost of multi-step look-ahead some approximation must to be done. The first approximation considers a small number of actions to be evaluated and the second only considers one action in t-steps, where t is number of steps missing to achieve the time in which all the actions are being compared, as we will see in the next section.

The control law to perform this task is evaluated using different cost functions provided by the different metrics presented above. The cost function for translation motion is,

$$\mathcal{J} = \max_{u_k} I(X; Z^k) \tag{6.9}$$

where the expected state is evaluated using the control input $\mathbf{u}_{k|k+1:k+\tau}(\mathbf{x}_p^*, \mathbf{z}^i_{k|k+1:k+\tau})$ computed to reach the desired poses $\mathbf{x}_p^*$ (giving a linear trajectory) and the expected measurements $\mathbf{z}^i_{k|k+1:k+\tau}$ after a time τ. A Dynamic Programming algorithm will be the solution for the optimal control problem, unfortunately in 60 [Hz] real world application the number of potential measurements grow exponentially in the dimension of the measurement space and in time.

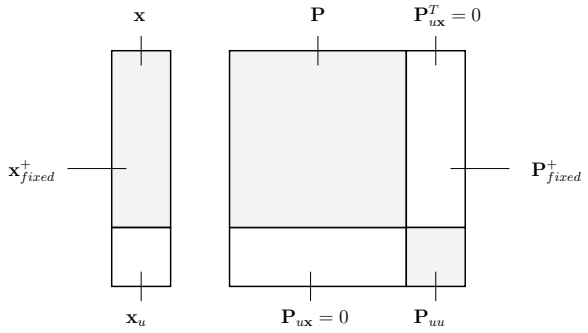


Figure 6.3: The EKF-SLAM state vector and covariance matrix representation for a fixed number of expected features with unknown mean and large covariance and with no cross covariance with the existing covariance matrix.

Using entropy for exploration only makes sense if we can be certain that uncertainty is reduced as landmarks (scene features) are being discovered. To that, one must have an idea first of the shape of the space to be mapped, and filling it with randomly placed landmarks with large uncertainty as in [18]. The error covariance matrix to evaluated is an extended matrix with a fixed number of non-correlated unvisited landmarks. In Figure 6.3 is shown a representation of the proposed vector state and covariance matrix.

With our chosen strategy overall entropy decay may happen at a lower pace, at the expense of actually choosing exploratory actions instead of homeostatic ones. Actions are compared at the same instant but evaluated at different time, that is the reason of using mutual information, together with the nonlinearity of the motion and measurement models.

6.3 Hand-Held Camera

As mentioned above the interest is focused on video-rate estimation *and control* of a single camera, moving rapidly with 6-DOF in 3D in normal human environments, mapping visual features with minimal prior information about motion dynamics. The aim is to localise the sensor and build a feature map by computing the appropriate control actions in order to improve overall system estimation. However, insisting on video-rate

performance using modest hardware imposes severe restrictions on the volume of computation that can take place in each 15 [ms] time step. Re-estimation must take place of course, but making strictly optimal camera movements would require, in addition, the computation of the derivatives of a well-chosen performance metric with respect to the inputs [1]. Such a computation remains unfeasible for 6-DOF nonlinear system and measurement models. Besides, human actions can only be approximate, and at low frequency. So, instead of computing the optimal motion command, we decide only upon a small set of choices.

The camera motion predictions are computed with the smooth unconstrained constant velocity motion model (5.24) presented in Example 5.4. Its translational and rotational velocities altered only by zero-mean, normally distributed accelerations and staying the same on average. The Gaussian acceleration assumption means that large impulsive changes of direction are unlikely.

This model is decoupled in terms of linear and angular velocities, therefore camera rotations do not affect the translation. The mutual information metric is used to chose the maximally informative translation action. In addition, since we want the camera to look at those landmarks with large uncertainty so as to reduce their covariance when observed from different locations. Then, three different strategies are evaluated for controlling camera rotation: a) constant angular velocity, b) maximising MI gain and c) maximising the trace of the Fisher information.

A full perspective wide angle camera model (5.26) is considered for estimation and action evaluation. This model was presented in Example 5.4.

6.3.1 Control Strategy

Combining the strategies of effectively controlling translation by maximising mutual information with controlling orientation by maximising the information available from the new position yields reliable active control of pose and velocity for a free moving camera, whilst building a map optimally, at the same time.

This way, by using the mutual information metric, maximally informative actions would prevent the camera from producing ill-posed measurements. Note that an omnidirectional sensor would not require a strategy to direct fixation. In this case, as opposed to the camera on a vehicle, translation and orientation changes are kinematically decoupled, for this reason, it makes sense to use different information measures in evaluating them.

Considering this unconstrained model, it is possible to use different metrics to evaluate the orientation changes, with the objective of looking at those landmarks with large uncertainty. The first option we propose is to use the same metric both for evaluating the effects of rotations and translations, MI. This measure will choose the direction that most reduces the uncertainty in general, for the camera pose and landmark positions. Our second choice is to maximise the trace of the Fisher information ($\mathbf{H}^\top\mathbf{S}^{-1}\mathbf{H}$). In this way we will be choosing the best direction to look at, in the sense that it is the one that is most informative with respect to the expected landmarks. A third choice is to use MI for translation and to control the angular velocity to remain constant ω_y.

The real-time requirements of the task preclude using an optimal control decision that takes into account all possible motion commands, which is impracticable to compute, leading to an exponential growth because of the "curse of history" of long term action evaluation. Instead we evaluate our information metrics for a small set of actions carried out over a fixed amount of time, and choose the best action from those.

The set of possible actions is divided in two groups. Mutual information is evaluated for the translational actions `go_forward`, `go_backwards`, `go_right`, `go_left`, `go_up`, `go_down`, and `stay`; and Fisher information is maximised from the set of orientation commands `turn_right`, `turn_left`, and `stay`.

In Appendix B we describe the simulated environment in which we test our control strategies as explained in Chapter 5. In our simulated setting, desired camera locations are predicted for the best action chosen, and the low-level control law presented in Section 5.2.2 is applied to ensure these locations are reached at the end of one second;

at which point the motion metric is again evaluated to determine the next desired action. Orientations however, are evaluated at frame rate, leaving the system to freely rotate, governed only by the information maximisation strategy.

The simulation considers a fixed number of expected landmarks to be found, and both the MI and Fisher information metrics are computed taking into account the corresponding full covariance matrices, including these unvisited landmarks, which have been initialised with large uncertainties. This is the only thing that prevents our control strategies from defaulting to homeostasis (see Figure 6.3).

Figure 6.4 contains simulation results from the mutual information strategy for the computation of motion commands, and compares various orientation computation schemes. The simulated environment represents a room $6\times6\times2.5$ $[\mathrm{m}]^3$ in size containing 33 randomly distributed point landmarks, out of which 6 are *anchors*, to be used as global references. Having these anchors the system is observable as is shown in Chapter 4 providing, in the same way, metric scale to the visual system.

The initial standard deviation in camera pose is 0.6 [m] in the z and x directions, 0.46 [m] in height y, and 45° in orientation, right after matching the fiduciary points, but before any motion takes place. Sensor standard deviation is set at 2 pixels, and data association is not known a priori. Instead, nearest neighbour χ-squared tests (see [9]) are computed to guarantee correct matching. New landmarks are initialised once their ratio of depth estimate to depth standard deviation falls below a threshold of 0.3. $\mathbf{P}_{uu} = 1.5$ $[\mathrm{m}^2]$ for each unknown feature.

The plots in Figures 6.4 and 6.5 show the results of actively moving a 6-DOF camera whilst building a map of 3D landmarks. In all cases, each of the seven motion actions will produce a displacement of 0.3 [m] in the corresponding direction. The mutual information metric is evaluated at each of these positions. The action that maximises the metric is chosen, and the camera is controlled to reach that position in one second with a PD control law. Orientation changes are computed every 50 [ms].

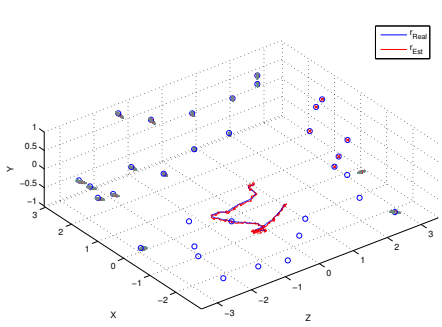
As mentioned before, three approaches were tested for the computation of gaze

commands with the following strategies: (i) constant rotational velocity of 0.2 [rad/s], frames (a,b); (ii) maximisation of mutual information both for the position and orientation of the moving camera, frames (b,c); and (iii) maximisation of mutual information for position and maximisation of Fisher information for gaze, frames (d,e). The experiment shown in the plots lasted 35 seconds.

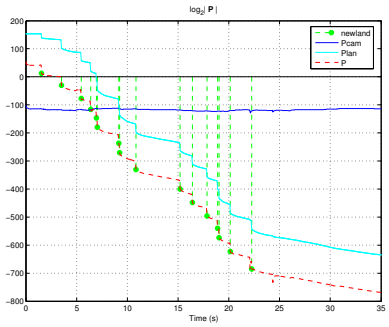
The constant rotational velocity and the mutual information strategies tend to insert landmarks into the map at a faster pace than the Fisher Information strategy. As can be seen in the error plots in Figure 6.5, this might not be always the best choice. It seems reasonable to let the system accurately locate the already seen landmarks before actively searching for new ones.

The third alternative, controlling camera orientation by maximising the Fisher Information entering into the filter, has the effect that it focuses on reducing the uncertainty of the already seen landmarks, instead of eagerly exploring the entire room for new landmarks. The reason is that landmarks that have been observed for a small period of time still have large depth uncertainty, and the Fisher Information metric is maximised when observations are directed towards them. The technique tends to close loops at a faster pace than the other two approaches, thus propagating correlations amongst landmarks and poses in a more efficient way. Additionally, reduction of the entropy occurs by positioning the camera where information is maximum, but there is no need to gaze the camera to reduce the pose and map entropies, instead the camera is gazed to the most uncertain features.

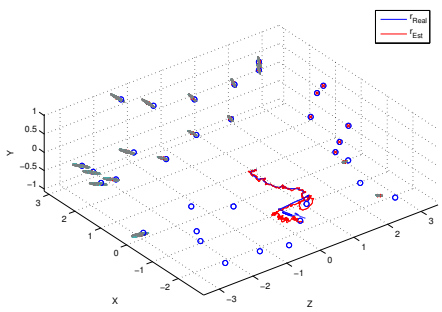
Strategy (iii) needs more time to reduce entropy and takes more time to insert the same number of landmarks in the map. But, at the point at which the same number of landmarks is available it has lower entropy than the other two strategies (see for example in Figure 6.4 for the simulation discussed, frames (d-f), that when the 14th landmark is added, the times are 19, 18, and 30 seconds, and the entropies are -530, -550, and -610).



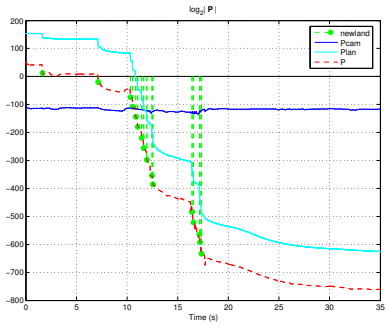
(a) Final Map by using MI for position and constant angular velocity.



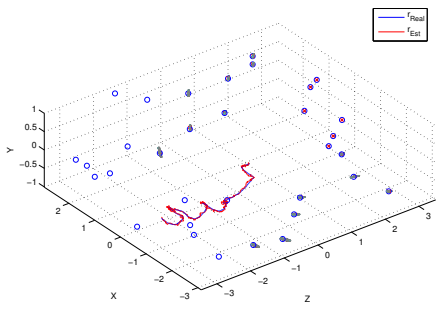
(b) Entropy for MI in position and constant angular velocity.



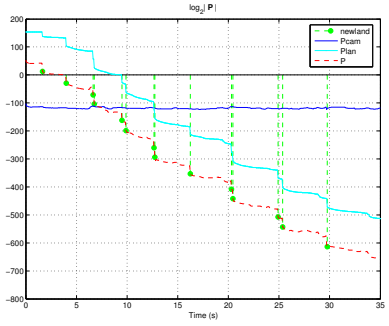
(c) Final Map by using MI for position and orientation.



(d) Entropy for MI in position and orientation.

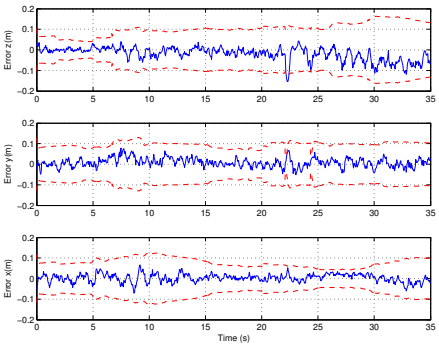


(e) Final Map by using MI for position and FI for orientation.

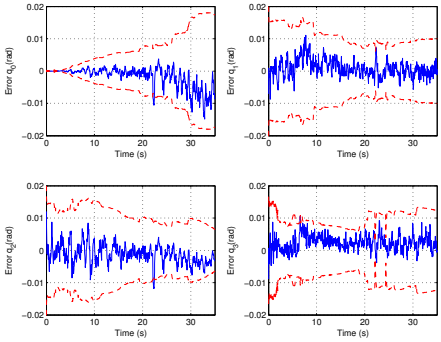


(f) Entropy for MI in position and FI in orientation.

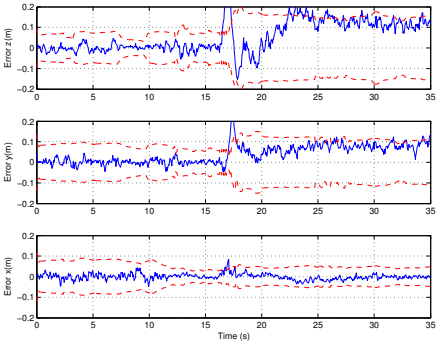
Figure 6.4: Trajectories with final maps and entropy. ($\mathtt{r}_{\mathtt{Real}}$ and $\mathtt{r}_{\mathtt{Est}}$ are the real and estimated camera trajectories, the label `newland` and the green dots and dotted vertical lines represent the value of entropy at the instant when new landmarks are initialised. `Pcam`, `Plan`, and `P` indicate the camera, map, and overall entropies.



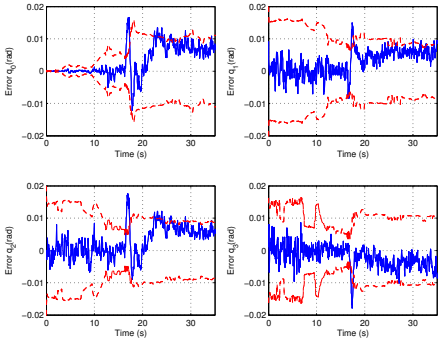
(a) Position error when using MI for position and constant angular velocity.



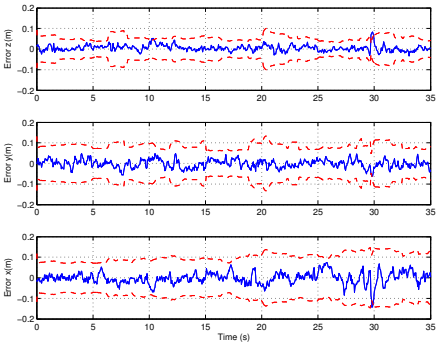
(b) Orientation error when using MI for position and constant angular velocity.



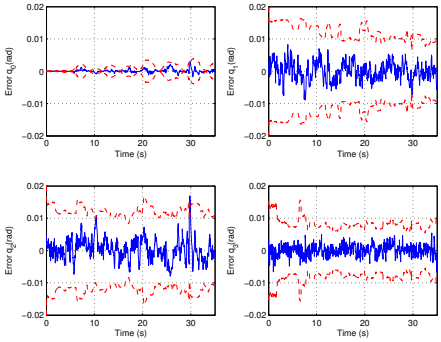
(c) Position error when using MI for position and orientation.



(d) Orientation error when using MI for position and orientation.



(e) Position error when using MI for position and FI for orientation.



(f) Orientation error when using MI for position and FI for orientation.

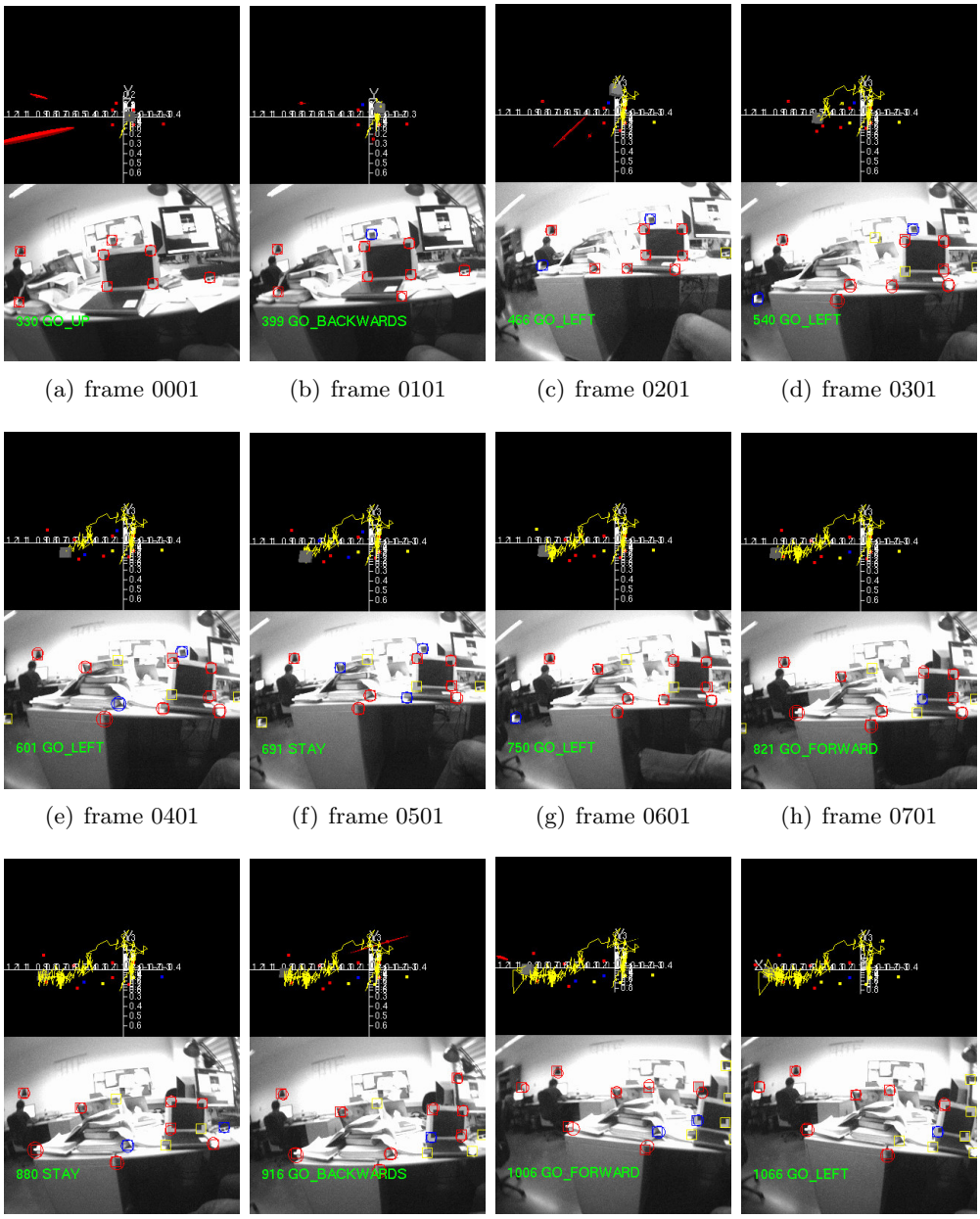
Figure 6.5: Estimation errors for camera position and orientation and their corresponding 2σ variance bounds. Position errors are plotted as x, y, and z distances to the real camera location in meters, and orientation errors are plotted as quaternions.

6.3.2 Experimental Results

We present in this section experimental results validating the mutual information maximisation strategy for the control of a hand-held camera in a challenging 15fps visual SLAM application. The experiments were implemented on top of the Single Camera SLAM application developed by Andrew Davison [36]. The characteristics of this application are explained in Appendix C. We developed an extention for this application that computes the desired actions using the approach presented in this chapter sending motion commands to the graphical user interface (GUI).

Within a room, the camera starts approximately at rest with some known object in view to act as a starting point and provide a metric scale to the proceedings (to make it fully observable at the beginning). The camera moves, translating and rotating freely in 3D, according to the instructions provided to a user through a graphical user interface, and executed by the user, within a room of a restricted volume, such that various parts of the unknown environment come into view. The aim is to estimate and control the full camera pose continuously during arbitrarily long periods of motion. This involves accurately mapping (estimating the locations of) a sparse set of features in the environment.

Given that the control loop is being closed by the human operator, only displacement commands are computed. So, for this particular case, gaze control is left to the user. Furthermore, the mutual information measure requires evaluating the determinant of the full covariance matrix at each iteration. Because of the complexity of this operation, single motion predictions are evaluated one frame at a time. It is only until the 15th frame in the sequence that all mutual information measures are compared, and a desired action is displayed on screen. That is, the user is presented with motion directions to obey every second. Note also, that in computing the mutual information measure, only the camera position and map parts of the covariance matrix are used, leaving out the gaze and velocity parts of the matrix. Finally, to keep it running in real-time, the resulting application must be designed for sparse mapping. That is, with

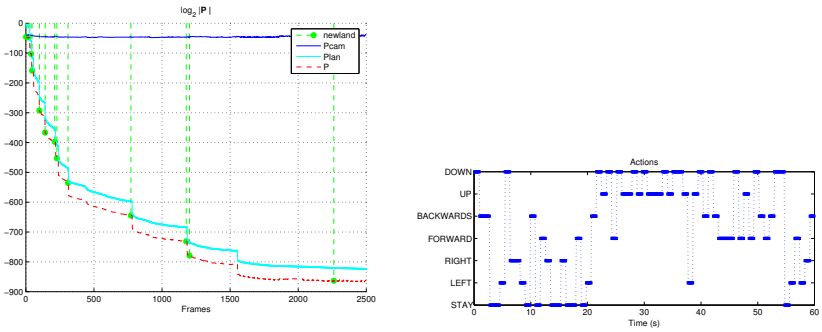


(a) frame 0001 (b) frame 0101 (c) frame 0201 (d) frame 0301

(e) frame 0401 (f) frame 0501 (g) frame 0601 (h) frame 0701

(i) frame 0801 (j) frame 0901 (k) frame 1001 (l) frame 1101

Figure 6.6: Snapshots of the Graphical User Interface (GUI) during active control monoSLAM.



(a) Camera, Map, and Total Entropies. (b) Actions for the first minute.

Figure 6.7: Real-time active visual SLAM with hand-held camera. a)Computed entropy during the experiment, the vertical lines represent the frame in which a feature was added to the state vector. b) Actions sent to the GUI.

the computing capabilities of an off-the-shelf system, our current application is limited to less than 50 landmarks.

In the library MonoSLAM (Appendix C) a class called MonoSLAMControl was implemented to compute the appropriate action, propagating states and covariances for each action and updating this priors with covariances associated to the features in the actual map inside of the expected field of view. After the whole EKF cycle, the same class computes the determinant of the resulting expected covariance matrix for each action and saves the maximum according the mutual information metric. The application MonoSLAMGlow was modified in order to print in the GUI the selected action.

Figure 6.6 shows the graphical user interface. The top part of the figure contains a 3D plot of the camera and the landmarks mapped, while the bottom part shows the information being displayed to the user superimposed on the camera view. Figure 6.7(a) contains a plot of the decrease in the various entropies for the map being built, and the list of actions chosen as shown to the user during the first minute (see Figure 6.7(b)).

Worth noticing is that in the real-time implementation, the system prompts the user for repeated up-down movements, as well as left-right commands. This can be

explained as if after initialising new features, the system repeatedly asks for motions perpendicular to the line of sight to best reduce their uncertainty. Also, closing loops has an interesting effect in the reduction of entropy, as can be seen around the 1500th frame on Figure 6.7(a).

6.4 Vehicle with a Single Camera

The typical wheeled mobile robot navigating in non-flat surfaces indoors or outdoors has the drawback of usually being modelled in a 2D world (see Example 3.4). In order to do the prediction step of the EKF, the predicted position using the 2D vehicle model or even the odometry information is quite different if we want the 3D information. For this reason, we propose a different model that not only considers the surface is not flat, but also considers the kinematic constraints of the differential skid vehicle and the camera position with respect to the rotation axis of the vehicle.

To model a wide-angle camera mounted on a 2D vehicle navigating in uneven terrains, we used the planar vehicle model (3.16) projected in a 3D cartesian space using the orientation given by the terrain. The resulting model is presented in subsection 6.4.1. The measurement model is again the one presented in Example 5.4.

In this case we have opted for the strategy that chooses those actions that maximise the mutual information between states and measurements to perform the navigation. The expected information gain is evaluated propagating a particular action using the constrained motion model proposed in this chapter, with the advantage that this model not only considers the nonholonomic constraints of the vehicle, but the slope of the terrain as well.

6.4.1 Constrained Camera Motion Model

It is assumed however, the camera is attached to a planar vehicle navigating in a 3D terrain. The vehicle is controlled by linear and angular velocities $\mathbf{u} = [v_r, \omega_r]^\top$ which

are tangent to the terrain surface. Robot motion taking into account surface contact at all times, we can substitute the previous motion prediction model with a constrained model for the continuous transition of the optic centre of the camera

$$\begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{\Psi}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{\Psi}_k \end{bmatrix} + \mathbf{\Gamma}\mathbf{u}_k \Delta t, \tag{6.10}$$

where

$$\mathbf{\Gamma} = \begin{bmatrix} -\sin\phi\sin\psi - \cos\phi\cos\psi\sin\theta & -l\cos\psi\cos\theta\cos\phi \\ \cos\phi\sin\psi - \sin\phi\cos\psi\sin\theta & -l\cos\psi\cos\theta\sin\phi \\ \cos\psi\cos\theta & -l\cos\psi\sin\theta \\ 0 & \sin\psi\tan\theta \\ 0 & \cos\psi \\ 0 & \frac{\sin\psi}{\cos\theta} \end{bmatrix},$$

$\mathbf{\Psi} = [\psi, \theta, \phi]^\top$ is a *yaw, pitch, roll* representation of $\mathbf{q}$, and l is the distance between the axle centre of the mobile robot and the camera optic centre.

6.4.2 Control Strategy

We present for this case the guidance strategy of a vehicle performing SLAM with a single wide-angle camera in even terrains. The control scheme is based on computing the instant vehicle accelerations that maximise mutual state and measurement information gain. Actions in the form of impulse accelerations guarantee smooth platform velocity change. The chosen command is then integrated to produce the input velocity that is sent to the vehicle. Given the real-time limitations of this system, only a limited number of actions can be evaluated at each step. These are the discrete set from Table 6.1.

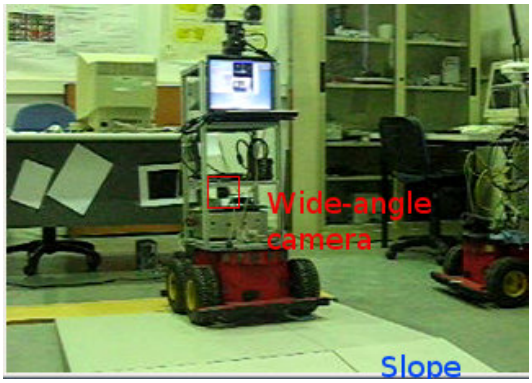


Figure 6.8: The mobile robot platform used in the experiments.

Table 6.1: Action Set

Action	Linear Acceleration	Angular Acceleration
0	0	0
1	0	$-\dot{\omega}_r$
2	0	$\dot{\omega}_r$
3	$-\dot{v}_r$	0
4	$\dot{v}_r$	0
5	$-\dot{v}_r$	$-\dot{\omega}_r$
6	$\dot{v}_r$	$\dot{\omega}_r$

To compare the actions, the motion model and the feedback control law from Example 5.4 are used to predict the prior mean $\mathbf{x}_{k+1|k}$ for each instant acceleration in the set, propagating the covariances by computing the corresponding Jacobians. Map features priors are also used to simulate the expected observations using the camera measurement model and the state prior. The posterior covariance is then computed taking into account only known landmarks inside the camera field of view.

At each time step we compute, in turn, the mutual information for one action in the set, using the prior and posterior covariance matrices. That is, for every linear and angular instant acceleration combination. Every 15th cycle, once all possible actions

have been evaluated for a lapse of at least 8 cycles, the action that maximises the mutual information is chosen, and a new velocity input is sent to the system.

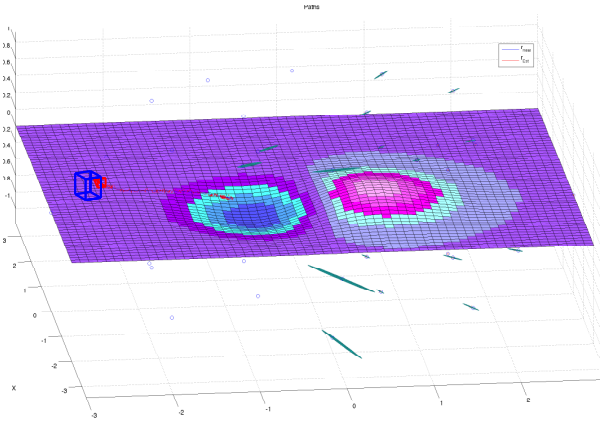
Assuming a fixed number of expected landmarks will be found within a 3D unexplored room as shown in Figure 6.3. During the action selection process, the unknown landmarks are taken into account in the covariance matrix initialised with large uncertainty as in the previous case.

For simulations the constrained motion model for the vehicle from (6.10), the full perspective wide angle camera model and a model of a 3D surface are used to simulate a mobile robot navigating in non-flat terrains. Not only in simulations, but also in real-time experiments the estimation does not use encoders information. In the other hand action decision process does use encoders velocity information and (6.10) to predict the expected position considering the set of available actions.

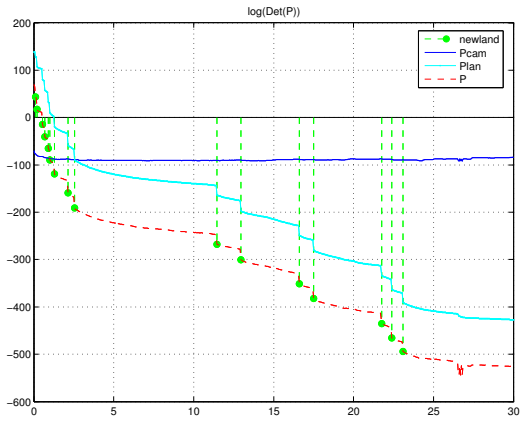
The aim is to choose impulsive acceleration commands for the vehicle in order to explore a small room while trying to reduce most the uncertainty. Accelerations are applied only every 15th step, and in between action decision, null acceleration is set, i.e. constant velocity behaviour is chosen until a new action is decided.

The control action is chosen from the discrete set of instant linear and angular accelerations shown in the Table 6.1. The values for $\dot{v}_r$ and $\dot{w}_r$ that produced the results shown in this section are 0.5 [m/s^2] and 0.3 [rad/s^2] respectively. The simulated environment shown contains 25 unknown landmarks and 6 known landmarks uniformly distributed in the room, as in the case of the unconstrained motion model these *anchors* give the metric and the observability condition to the SLAM system. The simulated wheeled mobile robot is navigating over a 3D sinusoidal surface.

Figure 6.9(a) shows the trajectory followed by the vehicle and the initialised landmarks with their uncertainty plotted as 2σ level hyperellipsoids. The expected covariance matrix is extended with the unknown feature uncertainties with diagonal values of 5 [m^2] each to avoid homeostasis. Entropy reduction is computed using the extended covariance. The instant at which new landmarks are added to the state are shown

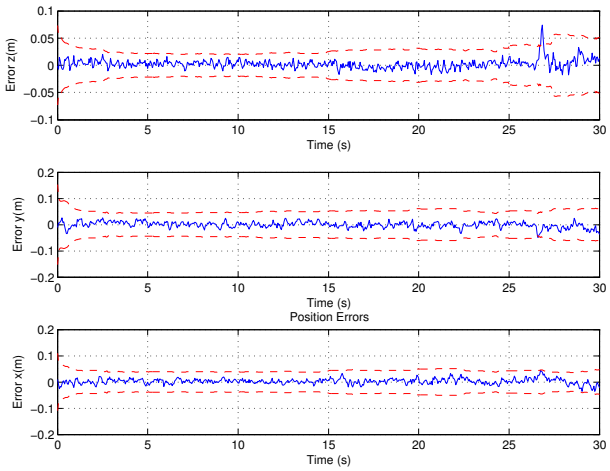


(a) Simulated robot trajectory

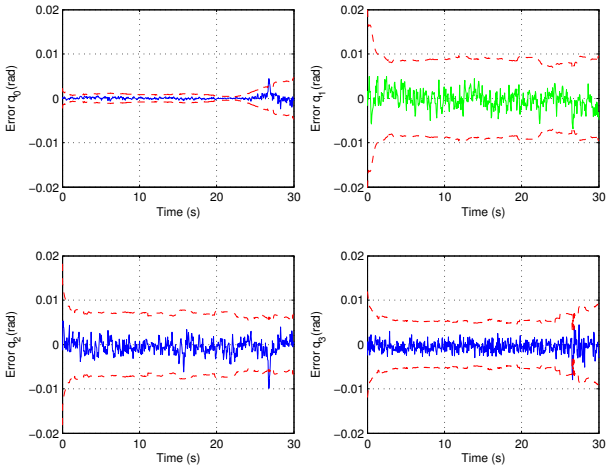


(b) Camera, map, and total entropies

Figure 6.9: Simulation of a mobile robot actively exploring a room. The mutual information maximisation strategy produces a nearly linear motion tangent to the surface. The vehicle starts at the shown terrain depression and proceeds backwards slightly rotating to increase map coverage. ($\mathtt{r}_{\mathtt{Real}}$ and $\mathtt{r}_{\mathtt{Est}}$ are the real and estimated vehicle trajectories, the label `newland` and the green dots and dotted vertical lines represent the value of entropy at the instant when new landmarks are initialised. `Pcam`, `Plan`, and `P` indicate the robot, map, and overall entropies.)



(a) Position error



(b) Orientation error

Figure 6.10: Estimation errors for camera position and orientation and their corresponding 2σ variance bounds. Position errors are plotted as x, y, and z distances to the real camera location in meters, and orientation errors are plotted as quaternions.

in Figure 6.9(b). Moreover, state estimation errors are shown in Figure 6.10 for the camera pose. Notice how when the terrain abruptly changes, the estimated velocities become underestimated in the direction the terrain changed. Thus in simulating vehicle motion, a more elaborate model taking into account surface discontinuities ought to be considered for very rough terrains.

The selected actions reduce the camera pose and velocity uncertainty first, tracking landmarks with low uncertainty. After that, the variance for unvisited landmarks with large uncertainty is reduced as new landmarks are added. Interestingly enough, the system autonomously explores by repeatedly choosing a negative linear acceleration. The effect is to augment the camera field of view with the consequent inclusion of new features in the model, but still maintaining known landmarks in sight, thus keeping the vehicle well localised at all times. In contrast to the previous case experiments, it is more difficult in this constrained motion setting to actively perform short loop closure orthogonal to the field of view. The reason being that the vehicle cannot achieve saccadic motions in the way a free-moving camera can.

At this point we can argue how the same tracking (unconstrained constant velocity 6-DOF motion model) and action selection strategies (maximising the mutual information between states and measurements) is capable of choosing different exploratory maneuvers depending on the characteristics of the platform: short loop closing for a 6-DOF free-moving camera, and backwards linear motion increasing the field of view for our vehicle.

6.4.3 Experimental Results

For the case where the camera is mounted on a vehicle we developed not only the module to compute the best of the set of action, but also the interface that communicates with it. In this case, the desired actions are sent both to the GUI and to the robot. A detailed explanation of this application and its libraries are presented in Appendix C.

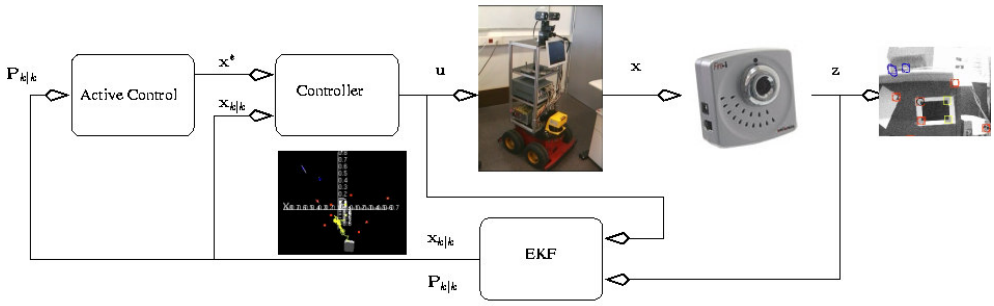


Figure 6.11: Structure of the visual navigation for a mobile robot in non-flat and partially unknown environments.

The robot is controlled using ARIA (Advanced Robotics Interface Application)[1]. ARIA communicates with the robot via a client/server relationship, using a serial connection. Two different process are running for this application, the MonoSLAM process and the RobotCONTROL, these processes communicate with each other using messages. The sort of information the processes share are the robot localisation, and the selected action.

Our main concern was to test the strategy during real-time vision-based SLAM execution. This section is devoted to a discussion on such results. The experiments were conducted on the mobile platform shown in Figure 6.8, with a wide-angle camera rigidly attached to the robot body, and for which an updated version of the single camera SLAM system reported in [36] was setup.

As in the previous case, within a room, the robot starts approximately at rest with some known object in view to act as a starting point and provide a metric scale to the proceedings. The robot moves, translating and rotating constrained by the 3D terrain, such that various parts of the unknown environment come into view.

The whole process is running at 15 [fps]. Figure 6.11 shows the proposed control scheme in a block diagram. Since our mutual information measure requires evaluating

[1] http://www.activrobots.com/SOFTWARE/aria.html

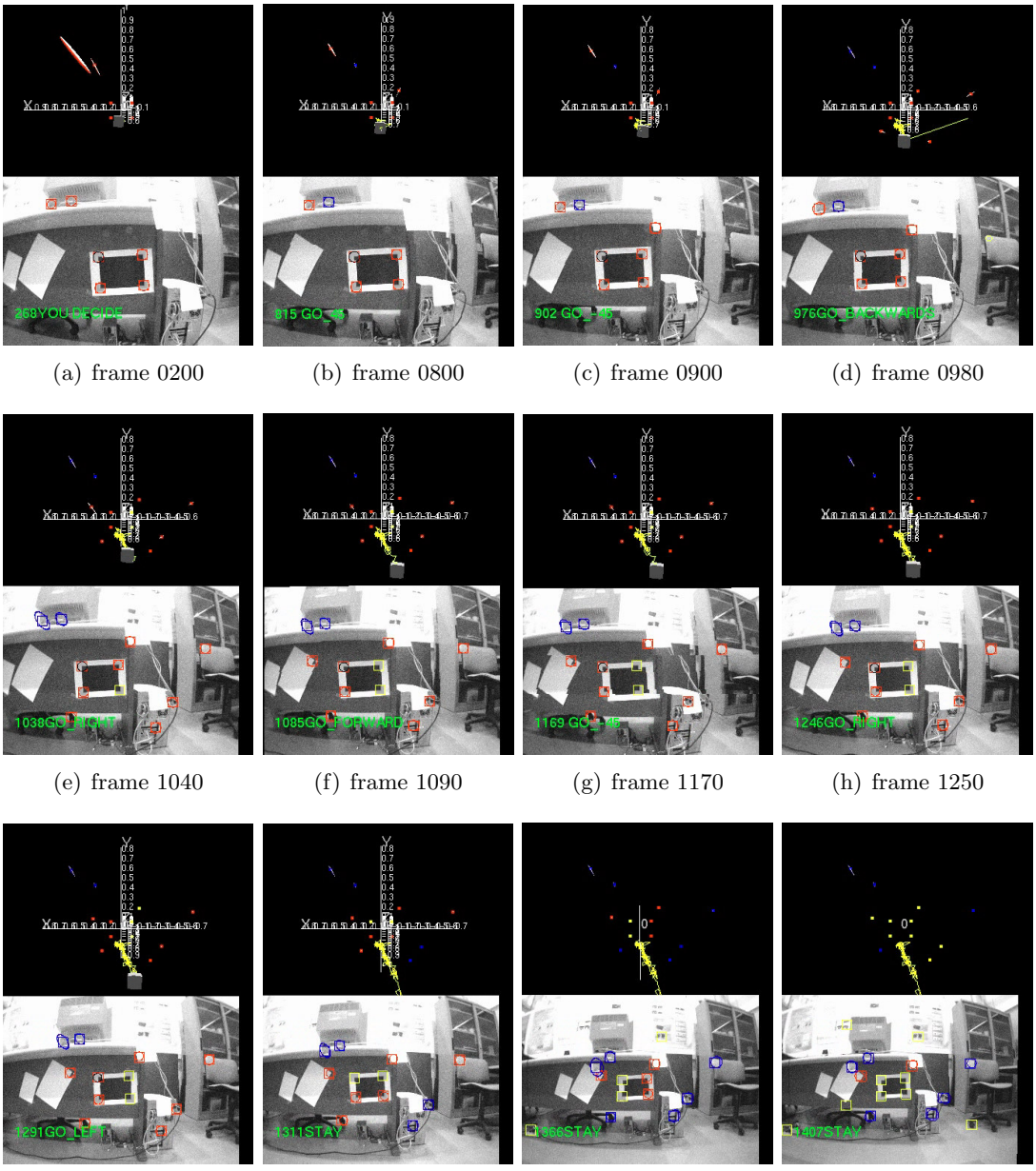
the determinant of the full covariance matrix (enlarged with the unvisited features) at each iteration, single motion predictions are evaluated one frame at a time. It is only every 15th frame in the sequence that all mutual information measures are compared, and the best action is sent to the mobile robot. For the experiments, the acceleration magnitudes were set to $\dot{v}_r = 0.1$ [m/s^2] and $\dot{\omega}_r = 5$ [deg/s^2]. When computing posteriors, these are all predicted for the duration that would take them to the end of the 15th frame, each action in turn being evaluated for a slightly shorter period of time. The motivation is that we want to be able to test actions in the basis of their effect at the very same point in time (at the end of the 15th frame). In order to evade any bias related to the time spent in evaluating the effect of actions, these are randomly ordered at each iteration.

As with the simulated setting, the robot navigates in uneven terrain as shown in Figures 6.8 and 6.13(a). In the plot, the estimated path (blue continuous line) is shown in 3D, as opposed to the vehicle odometry which is restricted to the $Z - X$ plane. The orientation angle from Figure 6.13(b) indicates the vehicle orientation with respect to the world axis Y (orthogonal to the white sheet of paper placed in front of the robot, which serves as global reference consistent to the world $Z - X$ plane).

As in the simulated case, our mutual information-based action selection strategy for this constrained motion case autonomously explores the room driving the vehicle back and forth, but mostly backwards, enlarging the field of view by pulling away from the initial view.

Figure 6.13(b) gives account of the actions sent to the robot, and shows as most frequent actions iterations between positive and negative linear acceleration. The feature map and camera pose are updated and displayed in real-time in the graphical user interface. Figure 6.12 shows a sequence of frames from the same experiment, that show the robot driving away from the initial known features.

Notice that in these experiments the robot can deal with obstacles, the obstacle avoidance task is part of a low-level behaviour in the control architecture.

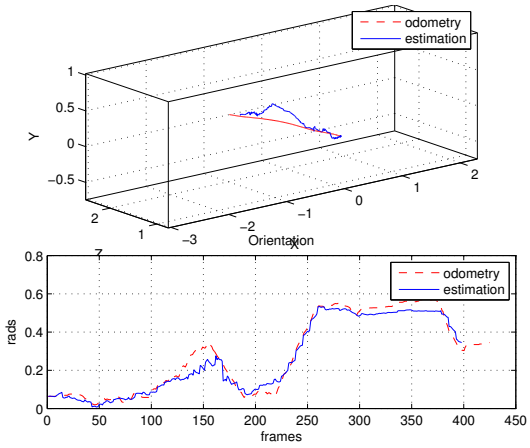


(a) frame 0200 (b) frame 0800 (c) frame 0900 (d) frame 0980

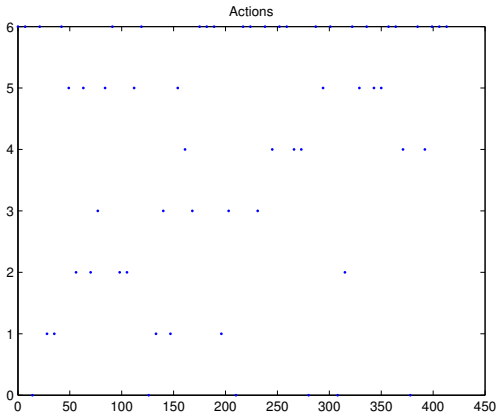
(e) frame 1040 (f) frame 1090 (g) frame 1170 (h) frame 1250

(i) frame 1300 (j) frame 1320 (k) frame 1370 (l) frame 1410

Figure 6.12: Snapshots of the Graphical User Interface (GUI) during autonoumous exploration.



(a) 3D Paths



(b) Actions followed

Figure 6.13: Real-time experiment of a single camera on a vehicle in uneven terrain. a) 3D trajectory of the vehicle and orientation w.r.t. y axis. b) Actions sent to the vehicle every second.

6.5 Conclusion

In conclusion, we have shown plausible motion strategies in a video-rate visual SLAM application. On the one hand, by choosing a maximal mutually informative motion command, we are maximising the difference between prior and posterior SLAM entropies, resulting in the motion command that mostly reduces the uncertainty of $\mathbf{x}$ due to the knowledge of $\mathbf{z}$. Alternatively, by controlling gaze maximising the information about the measurements, we get a system that prioritises in accurately locating the already seen landmarks before actively searching for new ones.

Our method is validated in a video-rate hand-held visual SLAM implementation. Given that our system is capable of producing motion commands for a real-time 6-DOF visual SLAM, it is sufficiently general to be incorporated into any type of mobile platform, without the need of other sensors.

A possible weakness of this information-based approach is that it estimates the utility of measurements assuming that our models are correct. Model discrepancies, and effects of linearisation in the computation of our estimation and control commands are thus not considered.

This chapter has presented as well an autonomous information-driven exploration strategy for a wheeled mobile robot equipped with a single wide angle camera and navigating in uneven terrains. The approach is based in choosing the action that maximises the information gain between state and measurement priors. Simulation and experimental results consistently show a behaviour in which the robot pulls back from an initial configuration, by having the camera search for more landmarks whilst reducing its own pose uncertainty.

We are able to localise a wheeled mobile robot in 6D (3D position and 3D orientation) using a single camera and driving it using an information theoretic approach.

The reported camera trajectories are simple because a) the robot is commanded by acceleration impulses that tend to drive the robot through smooth velocity changes,

and b) the real-time constraints of the implementation allow only for the evaluation of a very limited set of possible actions. The computational complexity in computing entropy does not permit large maps, in that case submapping will be a good solution.

It is worth noting that no high-level task-dependent path planning is being performed whatsoever. The exploratory actions are chosen purely in the context of entropy minimisation and a low-level control, in the case of the mobile robot navigating with a single camera.

Chapter 7

Conclusions

The main concern of this thesis has been to understand the navigation problem from the control theory perspective, so we divided the problem into two parts. The first part comprises the estimation (SLAM) in which we analysed stability, controllability, and most importantly observability conditions of the SLAM process. The second part involved the control and the estimation (SLAM with control) where we designed an active controller for a system considering a low level nonlinear feedback control and a higher level strategy to reduce uncertainty during estimation.

7.1 Estimation in SLAM

In this thesis we presented a stability analysis of the closed-loop estimation error dynamics (Chapter 3), showing that the partial observability inherent to SLAM process makes the estimate does not converge to the true values even for the linear case. This happens because the filter in which the vehicle and landmark states are stacked in the same state vector has marginally stable error dynamics. However, marginal stability does guarantee convergence of the corresponding Riccati equation to at least one positive semi-definite (*psd*) solution. Due the lack of asymptotic stability, it is not possible to obtain a zero mean state error estimate, unless partial observability is corrected,

either anchoring the state to one or more landmarks or by making using a relative maps.

In a controllability analysis we showed that the Kalman gain of the map components tends quickly to zero, the reason is landmarks in the map are assumed to be constant, so there is no process noise entering to these states producing the controllability condition is not satisfied. Adding artificial process noise can fix up this situation and equivalently it can reduce the computational complexity of the EKF algorithm under certain circumstances.

Considering the particular case of bearing-only SLAM, we modelled some nonlinear systems, such as the typical planar vehicle performing SLAM, as piecewise linear. A linear observability analysis for time-variant system was performed in Chapter 4. This analysis allowed us to characterise the non-observable states using the nullspace basis of the stripped observability matrix. Our results can be used to determine the set of motions that the vehicle must avoid in order to maximise observability.

Our dynamic observability analysis showed that the world-centric mapping with bearing-only observations is partially observable, recovering only the orientation component of the state if there exists translation of the vehicle. Some other configurations like world-centric with anchors and relative maps are fully observable, as the analysisi using stripped observability matrix shows in two steps. However, there are some directions that the system cannot recover in one step, but applying the appropriate vehicle motions the system becomes fully observable.

In addition, systems without proprioceptive sensors are more subtle, because only anchors can give the enough information to recover the scale considering observations of a single camera. We showed that linear velocity is easily recovered in one step and in two steps the system is verified as fully observable.

Bearing-only SLAM was also examined using non-linear observability analysis, in particular using Lie algebra, and the differences between the non-linear and TOM/SOM linearised analysis were compared.

7.2 Control in SLAM

We studied the stochastic problem of controlling a SLAM system. The appropriate solution for a controller in linear systems is the Linear Quadratic Gaussian regulator and we proposed to use the linearised version of this LQG controller. The LQG regulator allow us to stabilise the observable SLAM system about a desired point.

Using a different scheme, given the separability principle and given an observable SLAM system, we proposed a feedback linearisation control law to stabilise the system along a trajectory. We showed in simulations that not only the nonlinear vehicle but also the single camera are able to track a nonlinear path performing SLAM at the same.

It is important to stress out that the SLAM system should be fully observable in order to be able to use the estimates as input to any type of controller. Then, both estimation and control can be decoupled and standard techniques such as the ones used here, Kalman filtering for estimation, and feedback linearisation for control, are plausible for closing the perception-action-loop even in SLAM.

Bearing-only SLAM systems are quite sensitive in terms of estimation, due to the presence of long uncertainties along camera view axis. In order to the control aids the estimation process, a higher level strategy is needed. For this purpose, we designed an active control strategy that reduces uncertainty using mutual information gain as a metric to evaluate a set of discrete actions. Such strategy is presented in Chapter 5. One way to solve this optimal motion planning problem is applying a Dynamic Programming algorithm. The major disadvantage of such dynamic programming approaches is their extensive computational. Since we reason about a high-dimensional state estimation problem and because our real-time constraints (at least 15 fps), we had to approximate solutions for the motion command that rely on some assumptions. In essence, our approach can be regarded as one step look-ahead over the action-decision tree, but propagated in more than one step to evaluate every action at the same time. A future work must consider the evaluation of the optimal solution of this problem in real-time.

Our method was validated in a video-rate hand-held visual SLAM implementation. Given that our system is capable of producing motion commands for a real-time 6-DOF visual SLAM, it is sufficiently general to be incorporated into any type of mobile platform, without the need of other sensors.

Finally, in our last experiment presented in Chapter 6, we wanted to consolidate all the work presented in this thesis. We validated the active control strategy in real-time for a mobile robot navigating in a non-flat unknown terrain. The mobile robot was controlled using information from only one camera. It autonomously builds a visual feature map while at the same time optimises its localisation within this map using the most appropriate commands maximising the expected information gain between prior states and measurement. We developed the constrained 3D motion model to infer the position of the vehicle in order to evaluate the mutual information for all possible actions at the same time. Actions that hinder observability were avoided and the feedback linearisation control law was used to stabilise the vehicle along the chosen action until a new action was sent to the robot.

7.3 Future Work

There are several open issues in SLAM problem but for our particular case we consider as future work sensor fusion to improve mapping, as well a long-term plans to achieve large-scale exploration tasks or other kind of missions in the sense of navigation.

Strategies as local mapping must be considered in a further work, in order to perform exploration in larger environments not only indoors but also outdoors. Moreover some other descriptors (features) could be consider as well.

In this work we did not consider long-term navigation like mission or task planning as we defined in Section 2.1, that part is considered as a future work. Planning under uncertainty while mapping requires moving ahead from our approach involving local action selection, to longer term planning including task description. For instance, one

approach to start seeking in the problem is by planning in partially observable continues domains via value iteration over POMDPs [108].

Appendix A

Kalman Filtering

In this Appendix we will derive the Kalman filter equations from the state-space representation, instead of using the Bayesian point of view, as it is done in the original Kalman filter work [73]. We present the derivation for the linear case -the Kalman Filter- and its extention for non-linear systems -the Extendend Kalman Filter-.

A.1 KF-SLAM

Considering the problem of SLAM as the stochastic estimation of a discrete-time linear system, and expressed as a vector difference equation with additive white Gaussian noise that models unpredictable disturbances; the dynamic plant equation is simply

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{v}_k \tag{A.1.1}$$

where $\mathbf{x}_k$ is the augmented state vector formed by appending the vehicle state estimate and the landmark location estimates, $\mathbf{u}_k$ is a known input vector, and $\mathbf{v}_k$, is the k-th term of a sequence of zero-mean Gaussian process noise with covariance $\mathbf{Q}_k = E\left[\mathbf{v}_k\mathbf{v}_k^\top\right]$.

The measurement equation is

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \tag{A.1.2}$$

with $\mathbf{w}_k$ the k-th term of a sequence of zero-mean white Gaussian measurement noise with covariance $\mathbf{R}_k = E\left[\mathbf{w}_k \mathbf{w}_k^\top\right]$.

Using the aforementioned linear system, the algorithm for the optimal state estimator (the Kalman filter) is as follows:

First, compute the a priori state prediction

$$\mathbf{x}_{k+1|k} = \mathbf{F}\mathbf{x}_{k|k} + \mathbf{G}\mathbf{u}_k \tag{A.1.3}$$

followed by an a priori measurement prediction

$$\mathbf{z}_{k+1|k} = \mathbf{H}\mathbf{x}_{k+1|k} \tag{A.1.4}$$

Next, compute the a posteriori state estimate, known also as the update of the state estimate

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}\left(\mathbf{z}_{k+1} - \mathbf{z}_{k+1|k}\right) \tag{A.1.5}$$

where $\mathbf{K}$ is the Kalman gain for optimal estimation in the mean square error sense.

Finally, replacing the system from (A.1.3) and (A.1.4) in (A.1.5), the closed-loop system becomes

$$\begin{aligned} \mathbf{x}_{k+1|k+1} \quad = \quad & \left(\mathbf{F} - \mathbf{KHF}\right)\mathbf{x}_{k|k} + \\ & \left(\mathbf{G} - \mathbf{KHG}\right)\mathbf{u}_k + \mathbf{K}\mathbf{z}_{k+1} \end{aligned} \tag{A.1.6}$$

In order to compute the optimal filter gain for the linear system one needs:

The state prediction covariance

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^{\top} + \mathbf{Q}_k, \tag{A.1.7}$$

the innovation covariance

$$\mathbf{S}_{k+1} = \mathbf{R}_{k+1} + \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^{\top} \tag{A.1.8}$$

and finally, the filter gain

$$\mathbf{K} = \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^{\top}\mathbf{S}_{k+1}^{-1} \tag{A.1.9}$$

The update of the state covariance is computed with

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{S}_{k+1}\mathbf{K}^{\top} \tag{A.1.10}$$

For constant plant and sensor covariances, the steady state value for the covariance matrix is given by the solution of the Riccati equation

$$\mathbf{P} = \mathbf{F}(\mathbf{P} - \mathbf{P}\mathbf{H}^{\top}(\mathbf{H}\mathbf{P}\mathbf{H}^{\top} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P})\mathbf{F}^{\top} + \mathbf{Q} \tag{A.1.11}$$

A.2 EKF-SLAM

Let us consider now a nonlinear system where both process and measurement models are nonlinear.

Provided the set of observations $Z^k = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ was available for the computation of the current map estimate $\mathbf{x}_{k|k}$, the expression

$$\mathbf{x}_{k+1|k} = \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0}) \tag{A.2.1}$$

gives an a priori noise-free estimate of the new locations of the vehicle and map features

after the vehicle control command $\mathbf{u}_k$ is input to the system. Similarly,

$$\mathbf{z}_{k+1|k} = \mathbf{h}(\mathbf{x}_{k+1|k}, \mathbf{0}) \tag{A.2.2}$$

constitutes a noise-free a priori estimate of sensor measurements.

Given that the landmarks are considered stationary, their a priori estimate is simply $\mathbf{x}_{f,k+1|k} = \mathbf{x}_{f,k|k}$; and the a priori estimate of the map state error covariance showing the increase in robot localization uncertainty is

$$\begin{aligned} \mathbf{P}_{k+1|k} &= E[\tilde{\mathbf{x}}_{k+1|k}\, \tilde{\mathbf{x}}_{k+1|k}^{\top}] && \text{(A.2.3)} \\ &= \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^{\top} + \mathbf{G}\mathbf{V}\mathbf{G}^{\top} && \text{(A.2.4)} \end{aligned}$$

The Jacobian matrices $\mathbf{F}$ and $\mathbf{G}$ contain the partial derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$ and $\mathbf{v}$, evaluated at $(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{0})$ respectively.

Assuming that a new set of landmark observations $\mathbf{z}_{k+1}$ coming from sensor data has been correctly matched to their map counterparts, one can compute the error between the measurements and the estimates with $\tilde{\mathbf{z}}_{k+1|k} = \mathbf{z}_{k+1} - \mathbf{z}_{k+1|k}$. This error aids in revising both the map and robot locations. The a posteriori state estimate is

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}\tilde{\mathbf{z}}_{k+1|k} \tag{A.2.5}$$

and the Kalman gain is computed with (A.1.9) where $\mathbf{S}$ is termed the measurement innovation matrix,

$$\mathbf{S} = \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^{\top} + \mathbf{W} \tag{A.2.6}$$

and $\mathbf{H}$ contains the partial derivatives of $\mathbf{h}$ with respect to $\mathbf{x}$ evaluated at $(\mathbf{x}_{k+1|k}, \mathbf{0})$.

Finally, the a posteriori estimate of the map state error covariance must also be revised once a measurement has taken place. It is revised with the Joseph form to

guarantee positive semi-definiteness

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{KH})\,\mathbf{P}_{k+1|k}\,(\mathbf{I} - \mathbf{KH})^{\top} + \mathbf{KWK}^{\top}\,. \tag{A.2.7}$$

Appendix B

Simulated Environment

In this Appendix we are going to describe a toolbox developed for Simulink using Matlab. The code combines Simulink blocks and Matlab functions to create a closed-loop control and SLAM system. It is composed of

- Motion Models: This block contains a continuous motion model, an integrator (usually ODE45) and a normalisation function. We developed different motion models that can be interchange such as 1) a planar differential steer mobile robot, 2) a 3D differential steer mobile robot, and 3) a 6-DOF constant velocity model. The input of this block is the control signal and the output the simulated state, Figure B.2(a).

- Measurement Model: This block contains the obsevation models: a camera projection model or a laser-range scanner, i.e. range and bearing measurements. The input of this block is the simulated state and the output the measurements perturbed by Gaussian zero-mean noises.

- Filter (EKF): This block contains the Extended Kalman Filter. The input to this block is the control signal, the simulated measurements and the motion and measurement models. The output is the estimated state, Figure B.2(b).

- Controller: Depending on the navigation task, following a pre-designed path or uncertainty reduction, this block receives the estimated state and compute the appropriated control signal.

The blocks can be interchanged to allow the system to use different motion or measurement models, a different filter and different controllers.

This environment has different outputs, one is a Graphical Interface that shows online the simulated system, the estimated system B.3(a) and the camera view B.3(b) (in case such sensors were being used). A different output is the Matlab's workspace, all the variables to analyse the system behaviour are sent to the there as structures with time. Control signals and some other variables could be plotted on algorithm-time during the simulation execution.

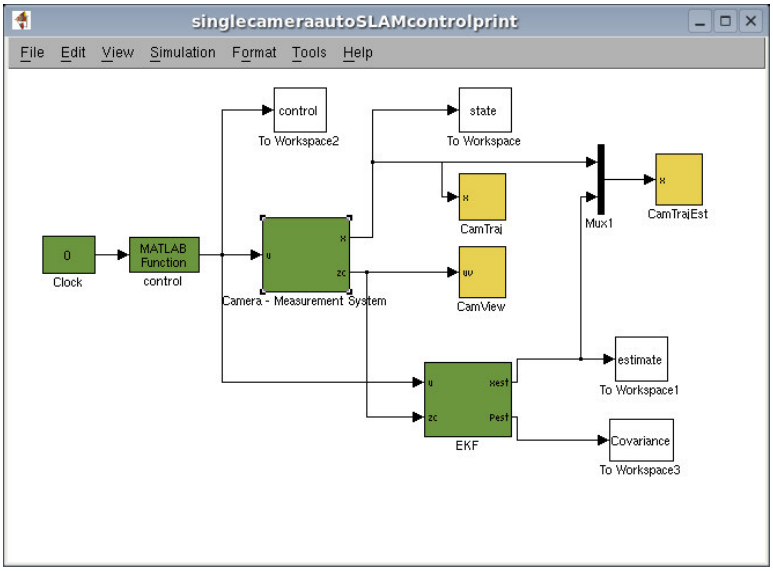
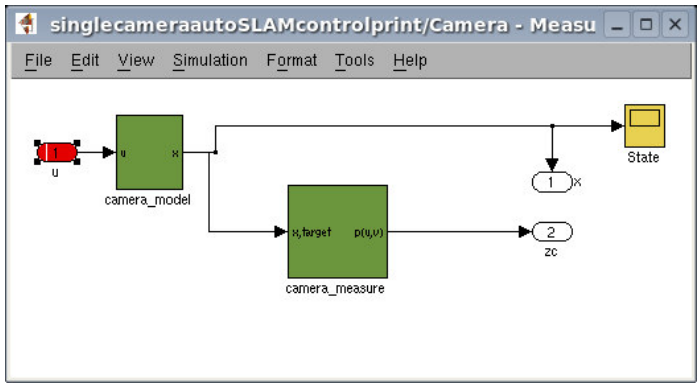
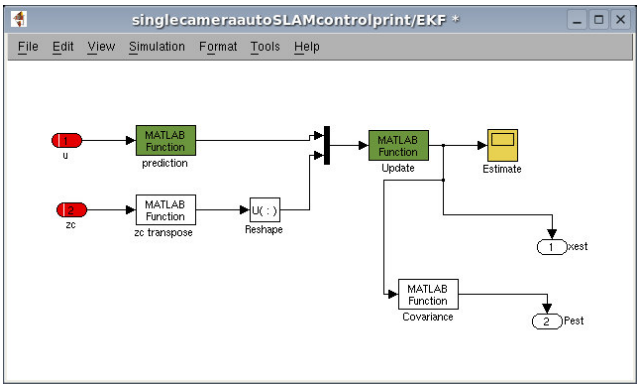


Figure B.0.1: Simulink main model of the controlled SLAM single camera simulator.

The toolbox can be used separately for dynamics system simulation, localisation, SLAM, and SLAM with control.

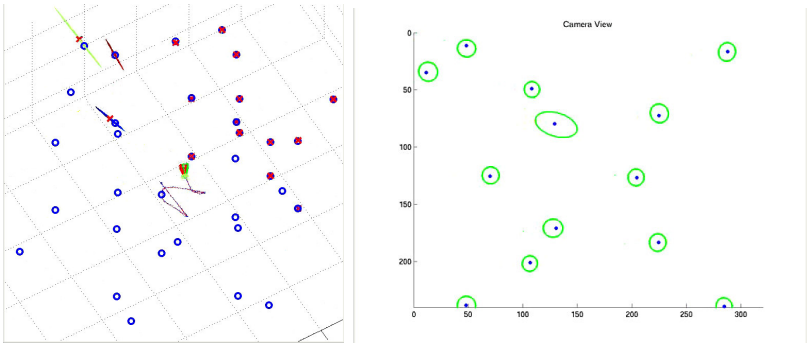


(a) Motion and measurement models



(b) Extended Kalman Filter

Figure B.0.2: Submodels of the controlled SLAM single camera simulator.



(a) Graphical output for simulated and estimated state. Camera pose and features position with their uncertainties plotted as 3D ellipses.

(b) Graphical output for camera view. Full perpective projection of the Simulated 3D points and the 2D uncertainty ellipses of each point.

Figure B.0.3: Graphical interface in simulate environment (Simulink/Matlab).

Appendix C

Single Camera SLAM Implementation

C.1 MonoSLAM

SceneLib is an open-source C++ library for SLAM designed and implemented by Andrew Davison (Imperial College, UK). SceneLib depends in part on the open source VW34 library for computer vision developed at Oxford's Active Vision Lab under the leadership of David Murray and Ian Reid, who have also had a large role in guiding the development of SceneLib. SceneLib is released with full source code under the GNU Lesser Public License (LGPL).

SceneLib is a generic SLAM library in principle, with a modular approach to specification of the details of robot and sensor types. However it also has specialised components to permit real-time vision-based SLAM with a single camera (MonoSLAM) and the design is optimised towards this type of application.

Available alongside the SceneLib library (which does not depend on any particular GUI) is source code which compiles to the executable program MonoSLAMGlow. This is an example application of MonoSLAM which can take images in real-time from

an IEEE1394 camera or off-line from a disk sequence and perform sparse monocular SLAM. The program has an interactive GUI using the GLOW toolkit and OpenGL.

MonoSLAM is written in standard C++ and it contains three libraries and an application.

• SceneLib: a generic SLAM library. Base classes for motion models, features, measurements and the Kalman Filter.

• SceneImproc: Image processing for MonoSLAM (i.e. feature detection and correlation).

• MonoSLAM: Specific motion and feature measurement models for single-camera SLAM, and a control class.

• MonoSLAMGlow: Application based on GLOW/GLUT library which uses the previous libraries.

An Extended Kalman Filter is used to estimate camera position, orientation, linear and angular velocities and 3D feature position using sequential innovation. A full covariance matrix $\mathbf{P}$ is maintained, complete with off-diagonal elements: typically clusters of close features will have position estimates which are uncertain with respect to the world reference frame but their highly correlated relative positions are well known. Holding correlation information means that measurements of any one of this cluster correctly affects the estimate of the others, and is the key to being able to re-visit and recognise known areas after periods of neglect.

Feature measurements are the locations of salient image patches. These patches are detected once to serve as long-term visual landmarks, giving us a sparse set of landmarks gradually accumulated and stored indefinitely. An active search within elliptical search regions defined by the feature innovation covariance and a template matching via exhaustive correlation search are implemented in order to match features between different sample times.

The real-time requirement only allows the system to add new features if number of visible features drops below a threshold (e.g. 12). Then it chooses salient image

patches from a search box in an underpopulated part of the image. Then, to compute an estimate of depth the line is sampled with 100 particles, spaced uniformly between 0.5m and 5m from the camera. When depth covariance is small, a new feature is added to the state vector. Moreover, features are deleted if more than half of attempted measurements fail.

A fundamental characteristic of full-covariance EKF SLAM is that the computational complexity of the filter update is of order N^2, where N is the number of features in the map. The Kalman Filter update time begins to grow rapidly when the number of features approaches 100, and going past this would require a different implementation, in order to meet the real-time requirement. Moreover if an action computation is required, as the one in Chapter 6, the amount of features that can be maintained is reduced by half.

C.2 Feature Extraction

The approach followed is similar to the one in [37], who showed that relatively large (9×9 to 15×15 pixels) image patches are able to serve as long-term landmark features with a surprising degree of viewpoint-independence. Each interest region is detected once with the Shi-Tomasi saliency operator, and match correspondences are detected in subsequent frames using normalised sum-of-squared differences [39]. Although more robust detectors such as the SIFT have become widely popular for their ability to find and match features with higher degree of uniqueness, they come at the expense of heavier computational load.

Image projection is modelled using the full perpective projection model with distortion. When an image feature is detected, its measurement must either be associated with an existing feature or be added as a new feature in the map. The location of the camera, along with the locations of the already mapped features, are used to predict the feature position (u, v) using (5.26), and these estimates checked against the mea-

surements using a nearest neighbour test. Feature search is constrained to 3σ elliptical regions around the image estimates as defined by the innovation covariance matrix $\mathbf{S}_i = \mathbf{H}_i \mathbf{P}_{k+1|k} \mathbf{H}_i^\top + \mathbf{R}$, with $\mathbf{H}_i$ the Jacobian of the sensor model with respect to the state, $\mathbf{P}_{k+1|k}$ the prior state covariance, and measurements $\mathbf{z}_i$ assumed corrupted by zero mean Gaussian noise with covariance $\mathbf{R}$.

Knowledge of $\mathbf{S}_i$ is what permits a fully active approach to image search; $\mathbf{S}_i$ represents the shape of a 2D Gaussian pdf over image coordinates and choosing a number of standard deviations (gating, normally at 3σ) defines an elliptical search window within which the feature should lie with high probability. In this system, correlation searches always occur within gated search regions, maximising efficiency and minimising the chance of mismatches. $\mathbf{S}_i$ has a further role in active search; it is a measure of the information content expected from a measurement. Feature searches with high $\mathbf{S}_i$ (where the result is difficult to predict) will provide more information about estimates of camera and feature positions.

C.3 Initialisation

Inserting a new feature to the map cannot be done immediately because the measurement model is non-invertible. Though bearing is recoverable from one measurement, 3D depth is not.

Several schemes have been reported [7, 39, 123], and it is adopted the first of these. The initial measurement results in a semi-infinite line with Gaussian uncertainty in its parameters, starting at the estimated camera position and heading to infinity along the feature viewing direction. A 1D particle distribution represents the likelihood of the 3D feature's position along this line. The line is projected as an epipolar line into subsequent images, but specifically it is the projection of the point particles and their uncertainly ellipses that provide the regions to be searched for a match, in turn producing likelihoods for Bayesian re-weighting of the depth distribution. A small

number of steps is required to reduce to below a threshold the ratio of the standard deviation in depth to the depth estimate itself. At that time, the depth distribution is re-approximated as Gaussian and the feature is initialised as a 3D point $\mathbf{x}_f^i$ into the map.

The projective nature of camera measurements means that while a measurement tells us the value of an image measurement given the position of the camera and a feature, it cannot be directly inverted to give the position of a feature given image measurement and camera position since the feature depth is unknown. Initialising features in single camera SLAM will therefore be a difficult task: 3D depths for features cannot be estimated from one measurement.

An obvious way to initialise features would be to track them in 2D in the image over a number of frames and then perform a mini-batch update when enough evidence had been gathered about their depth. However, this would violate our top-down methodology and waste available information: such 2D tracking is actually very difficult when the camera is potentially moving fast. Additionally, we will commonly need to initialise features very quickly because a camera with a narrow field of view may soon pass them by.

It is important to realise that a statement like not invertible does not have real meaning in a Bayesian framework, in which everything is uncertain and we must talk about probability distributions rather than in binary statements. Even after seeing a feature only once, we can talk about a PDF for its 3D position assuming that we had some prior belief about its depth. However, to use the feature in our SLAM map we require that its 3D position PDF can reasonably be modelled as a multi-variate Gaussian and this is why we cannot initialise it fully after just one measurement.

The approach we take therefore after one measurement is to initialise a 3D line into the map along which the feature must lie. This is a semi-infinite line, starting at the estimated camera position and heading to infinity along the feature viewing direction, and like other map members has Gaussian uncertainty in its paremeters. Its

representation in the SLAM map is: where is the position of its one end and is a unit vector describing its direction.

Along this line, a set of discrete depth hypotheses are made, analogous to a 1D particle distribution: currently, the prior probability used is uniform with 100 particles in the range 0.5m to 5.0m, reflecting indoor operation (quite a different type of prior may be required in larger environments where features may be very distant or even effectively at infinity). At subsequent time steps, these hypotheses are all tested by projecting them into the image. As Figure shows, each particle translates into an elliptical search region.

Feature matching within each ellipse (via an efficient implementation for the case of search multiple overlapping ellipses for the same image patch) produces a likelihood for each, and their probabilities are reweighted. Figure shows the evolution of the distribution over time, from uniform prior to sharp peak. When the ratio of the standard deviation of depth to depth estimate drops below a threshold, the distribution is safely approximated as Gaussian and the feature initialised as a point into the map. The important factor of this initialisation is the shape of the search regions

C.4 Map Management

With the ability to add features to the map comes the need for criteria to decide when this should be necessary, and potentially when features should be deleted. The map maintenance criterion aims to keep the number of reliable features visible from any camera location close to a predetermined value determined by the specifics of the measurement process, the required localisation accuracy and the computing power available. Feature visibility (more accurately predicted measurability) is calculated based on the relative position of the camera and feature, and the saved position of the camera from which the feature was initialised: the feature must be predicted to lie within the image, but further the camera must not have moved or rotated too far

from its initialisation viewpoint of the feature or we would expect correlation to fail. Features are added to the map if the number visible in the area the camera is passing through is less than this threshold. This criterion was imposed with efficiency in mind it is undesirable to increase the number of features and add to the computational complexity of filtering without good reason. Features are detected by running the image interest operator of Shi and Tomasi to locate the best candidate within a box of limited size (around 100×50 pixels) placed within the image this is for reasons of efficiency in a real-time implementation.

The position of the search box is currently chosen randomly, with the constraints only that it should not overlap with any existing features and that based on the current estimates of camera velocity and angular velocity any detected features are not expected to disappear from the field of view immediately. No effort is currently made to detect features in useful positions in terms of improving localisation information although this would be an interesting avenue for research more important is to find the features of strong image salience which exist in the image and to have them widely distributed across the image.

A feature is deleted from the map if, after a predetermined number of detection and matching attempts when the feature should be visible, more than a fixed proportion (in our work 50 percent) are failures. This criterion prunes bad features which are not true 3D points or are often occluded. A degree of clutter in the scene can be dealt with even if it sometimes occludes landmarks. As long as clutter does not too closely resemble a particular landmark, and does not occlude it too often from viewing positions within the landmark's region of expected visibility, attempted measurements while the landmark is occluded will simply fail and not lead to a filter update. Problems only arise if mismatches occur due to asimilarity in appearance between clutter and landmarks, and this can potentially lead to catastrophic failure. Correct operation of the system relies on the fact that in most scenes very similar objects do not commonly appear close enough to lie within a single image search region (and special steps would need to be taken to enable the system to work in scenes with repeated texture).

Bibliography

[1] A. Adam, E. Rivlin, and I. Shimshoni. Computing the sensory uncertainty field of a vision-based localization sensor. *IEEE Transactions on Robotics and Automation*, 17(3):258–267, June 2001.

[2] J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 397–402, New Orleans, April 2004.

[3] J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability when building fully correlated maps. *IEEE Transactions on Robotics*, 21(4):771–777, August 2005.

[4] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Multirobot C-SLAM: Simultaneous localization, control, and mapping. In *Proceedings of the IEEE ICRA Workshop on Network Robot Systems*, Barcelona, April 2005.

[5] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Stochastic state estimation for simultaneous localization and map building in mobile robotics. In V. Kordic, A. Lazinica, and M. Merdan, editors, *Cutting Edge Robotics*, chapter 3, pages 223–242. pro literatur Verlag, 2005.

[6] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Unscented transformation of vehicle states in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 324–329, Barcelona, April 2005.

[7] T. Bailey. Constrained initialisation for bearing-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1966–1971, Taipei, September 2003.

[8] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, Beijing, October 2006.

[9] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, 2001.

[10] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipole. *International Journal of Computer Vision*, 33(2):22–39, 1999.

[11] A. Bicchi, D. Pratichizzo, A. Marigo, and A. Balestrino. On the observability of mobile vehicles localization. In *Proceedings of the 6th IEEE Mediterranenan Conference on Control and Automation*, Sardinia, June 1998.

[12] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[13] J. Borenstein and Y. Koren. The vector field histogram fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, June 2001.

[14] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An *atlas* framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1899–1906, Taipei, September 2003.

[15] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the *atlas* framework. *International Journal of Robotics Research*, 23(12):1113–1139, December 2004.

[16] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptative robotic exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 540–545, Lausanne, October 2002.

[17] R. Brockett. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, pages 181–191. Birkhauser, 1983.

[18] M. Bryson and S. Sukkarieh. An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3770–3775, Edmonton, August 2005.

[19] M. Bryson and S. Sukkarieh. Active airborne localisation and exploration in unknown environments using inertial SLAM. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, March 2006.

[20] M. Buchberger, K. Jörg, and E. Puttkamer. Laser radar and sonar based world modeling and motion control for fast obstacle avoidance of the autonomous mobile robot 240 references mobot-iv. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 534–540, Atlanta, May 1993.

[21] D. Burschka and G. Hager. Vision-based control of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1707–1713, Seoul, May 2001.

[22] D. Burschka and G. Hager. V-gps - image-based control for 3d guidance systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1789–1795, Las Vegas, October 2003.

[23] D. Burschka and G. Hager. Principles and practice of real-time visual tracking for navigation and mapping. In *Proceedings of the International Workshop on Robot Sensing*, pages 1–8, Graz, May 2004.

[24] G. Campion, G. Bastin, and B. Andrea-Novel. Structural properies and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1):47–62, February 1996.

[25] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós. The SPMap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, October 1999.

[26] J.A. Castellanos, J. Neira, and J.D. Tardós. Limits to the consistency of EKF-based SLAM. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[27] K. S. Chong and L. Kleeman. Mobile robot map building for an advanced sonar array and accurate odometry. *International Journal of Robotics Research*, 18(1):20–36, January 1999.

[28] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.

[29] J. Clarke and A. Zisserman. Detection and tracking of independent motion. *Image and Vision Computing*, 14:565–572, Aug. 1996.

[30] P. Coelho and U. Nunes. Path-following control of mobile robots in the presence of uncertainties. *IEEE Transactions on Robotics*, (2):252–261, April 2005.

[31] F. Conticelli, D. Prattichizzo, F. Guidi, and A. Bicchi. Vision-based dynamic estimation and set-point stabilization of nonholonomic vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2771–2776, San Francisco, April 2000.

[32] J. Courtney and A. Jain. Mobile robot localization via classification of multisensor maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1672–1678, San Diego, May 1994.

[33] A. K. Das, R. Fierro, V. Kumar, B. Southhall, J. Spletzer, and C. J. Taylor. Real-time vision-based control of a nonholonomic mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1714–1719, Seoul, May 2001.

[34] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1403–1410, Nice, October 2003.

[35] A. Davison. Active search for real-time vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–73, Beijing, October 2005.

[36] A. Davison, Y. Gonzalez-Cid, and N. Kita. Real-time 3D SLAM with wide - angle vision. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[37] A. J. Davison and D. W. Murray. Simultaneous localisation and map-building uisng active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002.

[38] A.J. Davison and N. Kita. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Proceedings of the 15th IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 384–391, Kauai, December 2001.

[39] A.J. Davison, W.W. Mayol, and D.W. Murray. Real-time localisation and mapping with wearable active vision. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 315–316, Tokyo, October 2003.

[40] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[41] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal. Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. In *Proceedings of the American Control Conference*, pages 1493–1497, Chicago, June 2000.

[42] T. Duckett and U. Nehmzow. Mobile robot self-localization and measurement of performance in middle-scale environments. *Robotics and Autonomous Systems*, 24(1-2):57–69, August 1998.

[43] H. Durrant-Whyte. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. Technical Report SAND2001-3685, Sandia National Laboratories, November 1991.

[44] H. Durrant-Whyte, S. Majumder, M. de Battista, and S. Scheding. A Bayesian algorithm for simultaneous localisation and map building. In *Proceedings of the 10th International Symposium on Robotics Research*, Lorne, November 2001.

[45] C. Estrada, J. Neira, and J.D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, August 2005.

[46] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, December 2006.

[47] R.M. Eustice, M. Walter, and J. J. Leonard. Sparse extended information filters: Insights and sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, August 2005.

[48] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press, Cambridge, 1993.

[49] H.J.S. Feder, J.J. Leonard, and C.M. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18:650–668, 1999.

[50] J. B. Folkesson and H. I.Christensen. Robust SLAM. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[51] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 30:391–427, November 1999.

[52] U. Frese. Treemap: An $o(logn)$ algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103–122, September 2006.

[53] K. Furuta and A. Sano. *State Variable Methods in Automatic Control*. John Wiley & Sons, Chichester, 1988.

[54] P. W. Gibbens, G. M. W. M. Dissanayake, and H. F. Durrant-Whyte. A closed form solution to the single degree of freedom simultaneous localisation and map building (SLAM) problem. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 408–415, Sydney, December 2000.

[55] D. Goshen-Meskin and I.Y. Bar-Itzhack. Observability analysis of piece-wise constant systems - Part I: Theory. *IEEE Transactions on Aerospace and Electronic Systems*, 28(4):1056–1067, October 1992.

[56] J. Guivant and M. Nebot. Navigation in large outdoor unstructured environments. In *Proceedings of the 10th International Symposium on Robotics Research*, Lorne, November 2001.

[57] J. E. Guivant and E. M. Nebot. Optimization of simultaneous localization and map-builidng algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

[58] J. E. Guivant and E. M. Nebot. Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Transactions on Robotics and Automation*, 19(4):749–755, August 2003.

[59] G.D. Hager, D. J. Kriegman, A. S. Georghiades, and O. BenShahar. Toward domain-independent navigation: Dynamic vision and control. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 3257–3262, Tampa, December 1998.

[60] C. G. Harris and M. Stephens. A combined corner edge detector. In *Proceedings of the Alvey Vision Conference*, pages 189–192, Manchester, August 1988.

[61] K. Hashimoto. Visual servoing: Real time control of robot manipulators based on visual sensory feedback. *Robotics and Autonomous Systems*, 7:401–422, 1993.

[62] R. Hermann and A.J. Krener. Nonlinear observability and controllability. *IEEE Transactions on Automatic Control*, 22(5), 1977.

[63] S. Huang and G. Dissanayake. Convergence analysis for extended kalman filter based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 412–417, Orlando, May 2006.

[64] S. Huang, N.M. Kwok, G. Dissanayake, Q.P. Ha, and G. Fang. Multi-Step look-ahead trajectory planning in SLAM: Possibility and necessity. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, April 2005.

[65] T. S. Huang and O. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, 1989.

[66] S. Hutchinson, G. D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(1):649–774, 1996.

[67] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new method for the non-linear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, March 2000.

[68] S. J. Julier. The spherical simplex unscented transformation. In *Proceedings of the American Control Conference*, Denver, June 2003.

[69] S. J. Julier. The stability of covariance inflation methods for SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2749–2754, Las Vegas, October 2003.

[70] S. J. Julier and J. K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4238–4243, Seoul, May 2001.

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

[72] K. W. Jörg. World modeling for an autonomous mobile robot using heterogenoussensor information. *Robotics and Autonomous Systems*, 14:159–170, 1995.

[73] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering - Transactions of the ASME*, pages 35–45, March 1960.

[74] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel. Localisation et suivi robustes d'un robot mobile grâce à l'analyse par intervalles. *Traitement du Signal*, 17(3):207–219, 2001.

[75] B.H. Kim, D.K Roh, J.M. Lee, M.H. Lee, K. Son, M.C. Lee, J.W. Choi, and S.H. Han. Localization of a mobile robot using images of a moving target. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 253–258, Seoul, May 2001.

[76] J. Kim and S. Sukkarieh. Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045, July 2004.

[77] J. Kim and S. Sukkarieh. Improving the real-time efficiency of inertial SLAM and understanding its observability. In *Proceedings of the IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pages 21–26, Sendei, September 2004.

[78] S. J. Kim. *Efficient Simultaneous Localization and Mapping Algorithms using Submap Networks*. PhD thesis, Massachusetts Institute of Technology, Cambridge, June 2004.

[79] J. Košecká, R. Bajcsy, and M. Mintz. Control of visually guided behaviors. In C. M. Brown and D. Terzopoulos, editors, *Real-Time Computer Vision*. Cambridge Univ. Press, Cambridge, 1995.

[80] D. Kragic and H. Christensen. Robust visual servoing. *International Journal of Robotics Research*, 22(10), 2003.

[81] E. Krotkov and R. Hoffman. Terrain mapping for walking planetary rover. *IEEE Transactions on Robotics and Automation*, 10(6):728–739, 1994.

[82] D. Lambrinos, R. Moller, T. Labhart, R. Pfiefer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.

[83] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic, London, 1991.

[84] J. P. Laumon. *Robot motion planning and control*. Springer Verlag, Berlin, 1998.

[85] K.W. Lee, W.S Wijesoma, and J. Ibanez-Guzman. On the observability and observability analysis of SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3569–3574, Beijing, October 2006.

[86] J. Leonard and P. Newman. Consistent, convergent, and constant time SLAM. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1143–1150, Acapulco, August 2003.

[87] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–292, 1992.

[88] J. J. Leonard, R. R. Rikoski, P. M. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research*, 21(10-11):943–976, 2002.

[89] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach and D. Koditschek, editors, *Proceedings of the 9th International Symposium on Robotics Research*, Salt Lake City, November 1999.

[90] Y. Liu, T.S. Huang, and O. D. Faugeras. Determination of camera location from 2-d to 3-d line a point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:28–37, 1990.

[91] Y. Ma, J. Kosecka, and S. Sastry. Vision guided navigation for nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3):521–536, June 1999.

[92] D. J. C. MacKay. Information based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1992.

[93] R. Mahony and T. Hamel. Visual servoing using linear features for under-actuated rigid body dynamics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1153–1158, Maui, November 2001.

[94] E. Malis. Survey of vision-based robot control. Technical report, Institut National de Recherche en Informatique et en Automatique.

[95] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, April 1999.

[96] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, August 2005.

[97] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, New York, 1979.

[98] H. P. Moravec. The stanford cart and the cmu rover. *Proceedings of the IEEE*, 71(7):872–884, July 1983.

[99] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 116–121, St. Louis, March 1985.

[100] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, May 1993.

[101] E. Nebot, J. Guivant, and J. Nieto. ACFR, experimental outdoor dataset, 2002.

[102] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, December 2001.

[103] P. M. Newman. A short note on feature initialization and relocation. Technical report, Massachusetts Institute of Technology.

[104] P. M. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, The University of Sydney, Sydney, March 1999.

[105] J. Nieto, J. Guivant, and E. M. Nebot. DenseSLAM: The unidirectional information flow (uif). In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[106] G. Garcia P. Bonnifait. Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles. *IEEE Transactions on Robotics and Automation*, 14(4), August 1998.

[107] R. Pissard-Gibollet and P. Rives. Applying visual servoing tecniques to control a mobile hand-eye system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, May 1995.

[108] J.M. Porta, M.T.J Spaan, and N. Vlassis. Robot planning in partially observable continuous domains. In *Robotics: Science and Systems I*, Cambridge, June 2005.

[109] R. Rocha, J. Dias, and A. Carvalho. Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems*, 53:282–311, 2005.

[110] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1136–1141, Sacramento, April 1991.

[111] A. H. Sayed, B. Hassibi, and T. Kailath. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, 2000.

[112] L. Sciavicco and B. Siciliano. *Modeling and Control of Robot Manipulators*. Springer-Verlag, London, 2000.

[113] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, August 2002.

[114] S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, June 2005.

[115] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994.

[116] R. Siegwart and I.R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Intelligent Robotics and Autonomous Agents. The MIT Press, London, 2004.

[117] R. Sim. Stable exploration for bearings-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2422–2427, Barcelona, April 2005.

[118] R. Sim, G. Dudek, and N. Roy. Online control policy optimization for minimizing map uncertainty during exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1758–1763, New Orleans, April 2004.

[119] R. Sim and N. Roy. Active exploration planning for SLAM using extended information filters. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004.

[120] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 673–678, Barcelona, April 2005.

[121] J-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, 1991.

[122] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[123] J. Sola, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, August 2005.

[124] K. T. Song and J. H. Huang. Fast optical flow estimation and its application to real-time obstacle avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2891–2896, Seoul, May 2001.

[125] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems I*, pages 65–72, Cambridge, June 2005.

[126] R. Talluri and J. K. Aggarwal. Mobile robot self-location using model-image feature correspondence. *IEEE Transactions on Robotics and Automation*, 12(1):63–77, February 1996.

[127] C. Taylor. Building representations for the environment of a mobile robot from image data. In *Proceedings of the SPIE Conference on Mobile Robots*, pages 331–339, Boston, November 1995.

[128] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, October 1998.

[129] S. Thrun. Probabilistic algorithms in robotics. *Artificial Intelligence Magazine*, 21(4):93–109, 2000.

[130] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*, World Scientific Series in Robotics and Intelligent Systems. Morgan Kaufmann, 2002.

[131] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, September 2003.

[132] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hï¿½nel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

[133] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, July 2004.

[134] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hannel, M. Montemerlo, A. Morris, Z. Omohundro, and C. Reverte. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, 11(4):79–91, December 2004.

[135] D. P. Tsakirirs, P. Rives, and C. Samson. Applying visual servoing techniques to control nonholonomic mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, 1997.

[136] K. Usher, P. Corke, and P. Ridley. Home alone: Mobile robot visual servoing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, November 2001.

[137] T. Vidal-Calleja, J. Andrade-Cetto, and A. Sanfeliu. Conditions for suboptimal filter stability in SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 27–32, Sendei, September 2004.

[138] T. Vidal-Calleja, J. Andrade-Cetto, and A. Sanfeliu. Estimator stability analysis in SLAM. In *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.

[139] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, A. Sanfeliu, and J. Andrade-Cetto. On the observability of bearing only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4114–4118, Rome, April 2007.

[140] T. Vidal-Calleja, A.J. Davison, J. Andrade-Cetto, and D.W. Murray. Active control for single camera SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1930–1936, Orlando, May 2006.

[141] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto. Autonomous single camera exploration. In *Proceedings of the 2nd. Jornada de Recerca en Automàtica, Visió i Robòtica*, pages 121–127, Barcelona, July 2006.

[142] H. Wang and M. Brady. A structure-from-motion algorithm for robot vehicle guidance. In *Proceedings of the Intelligent Vehicles Symposium*, pages 30–35, Detroit, July 1992.

[143] H. Y. Wang, S. Itani, T. Fukao, and N. Adachi. Image-based visual asaptive tracking control of nonholonomic mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6, Maui, November 2001.

[144] P. Whaite and F.P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, March 1997.

[145] H. Zhang and J. Ostrowski. Visual motion planning for mobile robots. *IEEE Transactions on Robotics and Automation*, 18(2):199–208, April 2002.

[146] H. Zhang and J.P. Ostrowski. Visual servoing with dynamics: Control of unmanned blimp. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 618–623, Detroit, May 1999.