TÉCNICO LISBOA

Design and Validation of a Localisation and Control System for a Nonholonomic vehicle

João Vítor Viana Barbosa

Dissertação para a obtenção de Grau de Mestre em

Engenharia Aeroespacial

Júri

Presidente:	Prof. João Miranda Lemos
Orientador:	Prof. Paulo Oliveira
Co-orientador:	Prof. Carlos Cardeira
Vogais:	

Outubro de 2013

To my mother and sister.

Acknowledgments

A Masters thesis represents, for us Aerospace Engineering students, not only the end of a five year journey, but also the opportunity to embrace a complex and appealing subject and dedicate a longer time to it. For introducing me to this challenge, giving more significance to the end of this cycle, and being always there to guide, support and motivate me, I show my gratitude to my advisor Professor Paulo Oliveira and co-advisor Professor Carlos Cardeira. I would also like to thank Eng. Camilo and Mr. Raposeiro for their invaluable assistance and availability to build the prototype used in this work and without which it would not be possible. My thanks goes to Professor Miguel Tavares, for letting me visit the Biomechanics Laboratory of Lisbon and assisting me by collecting the experimental data that greatly contributed to the validation of this work.

I want to thank my friends Nuno Carvalho and Claúdia Dias, my usual project partners, with whom I learned a lot and made this journey smoother. To Joel Reis, for his unconditional help and friendship throughout these years, my sincere thanks. To all the friends I met at the university, whose names go without saying, who intentionally or not, contributed to the completion of this work, for their support, my thanks. I would also want to thank the guys at the laboratory for the witty jokes, good advise and excellent working environment.

Last, but not least, to my mother and sister, who always helped keep my mind in place, for their appreciation and unconditional support during both good and bad moments, without whom I would not have become who I am today, my deepest gratitude.

This work was partially supported by : i) the project PRODUTECH-PTI (P. 3904) under the program COMPETE/QREN/FEDER; ii) FCT, through IDMEC, under LAETA; and iii) the project PTDC/EEA-CRO/102102/2008, IDMEC.

Resumo

Esta dissertação propõe e valida experimentalmente um sistema de acostagem, implementado num robô de locomoção diferencial, composto por um sistema de localização e por uma lei de controlo não-linear por retroacção. A observação do estado é conseguida através de sistema de localização baseado num único ponto de referência visual e composto por dois filtros, um para a estimação de posição e outro para a estimação de orientação. A estimação é feita no referencial do corpo permitindo assim a utilização de um modelo cinemático LPV (Linear Parameter Varying). Os estimadores globalmente estáveis resultantes são parametrizados por dados de odometria e corrigidos por medições de posição e atitude relativas de uma referência visual fornecidas por um sensor RGB-D (red, green, blue and depth) instalado a bordo do robô. Foram feitas experiências utilizando um robot móvel e um sistema *Qualysis Motion Tracking* para validação com trajectória real. A posição e orientação assim como o escorregamento linear e angular são estimados, ambos com prova de observabilidade, resultando num sistema de localização em tempo real eficaz sem que seja necessário que a referência visual esteja sempre visivel. É feito um estudo estatístico dos erros de estimação para verificação das suposições de ruído Gaussiano inerentes à teoria do Filtro de Kalman. A convergência do erro é alcançada independentemente da estimativa inicial de posição e orientação fornecida, validando portanto a estabilidade global do sistema de localização. O problema de acostagem é resolvido encontrando uma lei de controlo suave, invariante no tempo, global e assimptoticamente estável recorrendo a técnicas de Lyapunov exprimindo a cinemática do robô em coordenadas polares, permitindo uma condução, semelhante à humana, em anel fechado que guia o robô até um determinado objectivo com uma dada orientação e uma curvatura ajustável. São feitas simulações do problema de acostagem para ilustrar a peformance do sistema, sendo este também validado com experiências realizados com o robô de locomoção diferencial supramencionado.

Palavras-chave: Lyapunov, Acostagem, Localização, Kalman Filter, Robótica Móvel

Abstract

This thesis proposes and experimentally validates a docking system for a differential drive robot, composed by a localisation system and full state feedback control law. Observation of the state consists of a single landmark-based absolute mobile robot localization system, composed by two filters, one for attitude estimation and the other for position estimation. The estimation is carried out in the body-frame allowing for the model kinematics to be LPV (Linear Parameter Varying). The resultant globally stable estimators are parametrized by odometry data and updated by landmark position and attitude measurements provided by an on-board RGB-D (red, green, blue and depth) sensor. Experiments were carried out, making use of a wheeled mobile robot and a Qualysis Motion Tracking System for ground truth validation. Attitude and position as well as linear and angular slippages, both proven observable, are estimated, resulting in an effective real-time localization system without requiring the landmark to be always visible. A statistical study of the estimation error was carried out to verify the Gaussian noise assumptions inherent to the use of Kalman filtering. Error convergence is achieved regardless of the initial estimate of both position and attitude, validating the system global stability. The docking problem is solved by finding a smooth, time-invariant, globally asymptotically stable feedback control law using Lyapunov techniques by expressing the kinematics of the robot in polar coordinates, allowing for a very human-like closed-loop steering that drives the robot to a certain goal with a desired attitude and a tunable curvature. Simulations of the docking problem are presented to illustrate the performance of the system and it is also validated by performing tests on the aforementioned real robot.

Keywords: Lyapunov, Docking, Mobile Robots, Kalman Filter, Localisation

Contents

List of Figures

Nomenclature

Symbol conventions

$\mathbf{A}$ Matrix belonging to $\mathbb{R}^{m\times n}$.

$\mathbf{a}$ Vector belonging to $\mathbb{R}^{m}$.

a Scalar variable.

A Scalar constant.

Landmark and Vehicle notation

$\mathbf{p}$ Position belonging to either $\mathbb{R}^2$ or $\mathbb{R}^3$.

ψ Attitude.

v, ω Respectively, Linear and Angular Velocity.

s, b Respectively, Angular and Linear SLippages.

$\mathbf{n}$ Normal vector.

$\mathbf{e}$ 2D Position of Landmark $\{B\}$.

Subscripts

i, j, k Computational indexes.

x, y, z Cartesian components.

t Indication of time dependency.

J Relative to reference frame $\{J\}$.

Superscripts

* Equilibrium point.

T Transpose.

J Expressed in reference frame $\{J\}$.

Chapter 1

Introduction

The automation of industry is a major motivation and driver of research in the area of robotics, resulting in facilities often operated by autonomous units, be it robotic arms, complete production units or autonomous ground vehicles used for transportation of cargo and raw or processed materials. The area of mobile robotics has been especially explored due to its vast potential and broad set of applications, but also due to the complexity of certain tasks and the vast amount of fields needed to be covered to make them possible. Mobile robots, unlike their stationary counterparts, have an element of versatility, being able to apply their functionalities where they are most needed, and many levels of automatism can be considered. The most basic automatic control can be found in some *teleoperated* robots, where the only action that is not controlled by a human operator is the sometimes complex act of locomotion, as is the case for legged robots such as RHex [2] or BigDog [3] which are usually meant to operate in hazardous or rough terrain. A mobile robot can also be operated by a set of way-points, where it autonomously decides which trajectory to take from one place to another. The automatism reaches its peak when no human input is needed and the robot sets objectives for himself, as is the case of robots with continuous tasks such as a cleaner robot.

When it comes to mobile robots, two major disciplines concerning autonomy arise, being those *localisation* and *navigation*, not overlooking, of course, all the sensory and actuator engineering necessary for the former two to be possible. They are both connected by the simple fact that the robot needs to be aware of its state, i.e., have a feedback of its position and attitude in order to compute its next action or what control commands to send to the actuators. Localisation of mobile robots can rely on hardware independent from the robot, as is the case of transponders, reflectors, global positioning systems or external cameras, or they can localise themselves by only using on-board sensors. The latter approach is of paramount importance as is reduces hardware costs and also allows for a greater versatility. On-board localisation systems usually take benefit of a sort of range detection system such as sonar, laser-range finders or even stereo-cameras. When used with hardware able to sense the kinematics or dynamics of the movement, such as optical wheel encoders and IMU boards, open doors for the use of several localisation algorithms.

The type of application of a particular mobile robot and its working environment are the main drivers in the choice of the kind of localization needed, ranging from a topological kind of localization [4, 5], often aided by a structured map or other representation of the environment, to a scenario where the robot may have to build its own map of the surroundings while simultaneously localizing itself in it, solution that is widely known as SLAM [6, 7]. These algorithms are usually based in particle filters [8, 9, 10, 11] or Extended Kalman Filters [12, 13], often making use of visual landmarks as in [14] or previously learned features present in the environment [15]. Recently, using RGB-D cameras such as Microsoft Kinect ™for SLAM algorithms has become quite popular [16], mostly due to the versatility and low cost associated with these sensors.

Having an industrial application in mind, this work envisions a localisation system that exploits structured visual landmarks for guidance of a differential drive robot to a certain objective.

1.1 Motivation

Mobile robots' tasks in industrial facilities most often include the transportation of cargo or raw/processed materials from one point to another, being then essential for the robot to be able to identify packages, localise them in its referential and approach them for transportation. Although localisation algorithms for unstructured environments often show very good results, they may not be enough for tasks of precision such as the docking of a mobile robot in a cargo container or a charging station.

This work proposes an innovative method of localisation of a landmark in the body frame of a robot based on Linear Kalman Filters by using feature recognition with an RGB-D camera and odometry readings from optical wheel encoders. Furthermore, the filtering solution also estimates both linear and angular slippages by making use of the ability to recognise both position and attitude of a given landmark in the body frame of the robot. The second objective of this work is to design a control feedback law that drives the robot from an initial position to a desired location with a given angle of arrival, task commonly known as docking or parking problem in the scientific community.

Having the robot be guided by a vision system allows for more versatility in an industrial facility regarding docking stations, positioning of cargo pallets, consequently simplifying the overall task map building and task planning, dropping the need of extreme precision regarding these actions. The developed work does not mean to completely compose a fully autonomous system but rather be a modular algorithm used by any task planner implemented.

1.2 State-of-the-art

Ever since mobile robots began to be used, the docking problem has been studied in a number of different approaches, given its important role in the automatism of a robots' task, since it is desirable that the robot is able to sustain long term activity by recharging itself [17]. The solutions found in the literature to solve this problem vary both in algorithm and sensor payload. One approach, defined as visual servoing, of which an early contribution was [18], is to represent a given task directly by an error relative to a goal image to be captured by the vision system. This approach saw a greater development from 1990 onward with works such as [19, 20], with a great contribute of the task function approach [21]. Visual servoing benefits from contributes with out-of-body cameras, i.e., Camera-Space Manipulation (CSM) [22], Mobile Camera-Space Manipulation (MCSM) which extends the latter with body embedded cameras and more recently [23] which computes the goal configuration using visual landmarks.

Other approaches to the docking problem include the computation of feedback control laws by using Lyapunov and backstepping techniques that lean on Ultra-Short Baseline (USBL) acoustic positioning [24] applied on the underwater counterpart of this work, the use of electromagnetic homing systems [25], optical guidance approaches such as [26] and computing the deceleration needed by a robot, resorting to an estimation of a *time-to-contact* (τ) through optical flow field divergence measurements of an image stream as in [27] and references therein. In [28] a method based on the direction of arrival (DOA) of signals transmitted by RFID transponders is proposed, showing that a robot can dock in a station transmitting through an RFID by using two antennae installed on-board of the vehicle. A localisation method using homography between several consecutive images is proposed in [29]. A method proposing the estimation of the position and orientation of a visual landmark is proposed in [30] for later docking and automatic recharging, thus being similar to the work here presented.

1.3 Contribution

The work presented in this thesis is composed by two major parts: i) Vision Based Localisation system; and ii) Nonlinear Control for the Parking Problem. The main contribution of the first part is that the localisation system leans on Linear Kalman filter theory, benefiting from all the celebrated stability characteristics inherent to it, allowing for the localisation to rely on exact linear and angular kinematic motion laws and excluding the need of transposing measurements to the estimate frame, which is beneficial seeing as this reference changes often have a degree of uncertainty associated. Also the localisation system, relying on Kalman Filtering, has fully characterized uncertainties. Moreover, the localisation strategy was implemented on a real-time system and validated with ground truth data. Regarding the parking problem, the main contribution consisted on the implementation of a smooth, time-invariant and globally asymptotically stable nonlinear feedback control law on a real-time system and validation of the results, having used the implemented localisation estimate as the output of the system, providing this work with a multidisciplinary character. Furthermore, this work envisions to attempt implementation on an industrial setting composed by a manifold Control, Localisation and Decision System that will be managing tasks of a crew of AGVs responsible for the transportation of raw and processed materials in several industrial factories. A vision based feedback control law allows for a more versatile industrial setting, relaxing the need of pallets and other sort of cargo having a precise positioning in a given map, thus facilitating, for instance, operations in an unloading area.

1.4 Thesis Outline

This thesis is organised in seven chapters, including the introductory chapter. Chapter 2 introduces the problem by presenting the derived kinematics of a differential drive robot and, after introducing Kalman Filtering theory, presenting the filtering solution. Chapter 3 presents the necessary tools and definitions for nonlinear control stability analysis and a proposition for a smooth nonlinear feedback control law that stabilises the system around a limiting point. Next, details of the implementation of each system, including the vision algorithm, and also the depiction and solving of some difficulties that arise when trying to implement a real-time system are described in Chapter 4. The two following chapters present the validation of the proposed methods by simulation (Chapter 5) and by real-time experimental results (Chapter 6). Finally Chapter 7 presents the conclusions of this dissertation.

Chapter 2

Sensor based localisation relative to a landmark

In this chapter, the problem of designing a localisation system, based on measurements relative to a landmark, will be addressed. To solve this problem, a Kalman filter will be used, where no linearisation will be needed. The fact that the localisation system is meant to localise a certain landmark in the sensor based framework alleviates the problem from the often needed representation of the sensor measurements into a filtering framework. This avoids, not only, nonlinearities associated with such change of coordinates, but also the possible augmentation of sensor noises due to the transformation having itself to be estimated. This work assumes that the attitude and position of the landmark can be measured by a sensor such as a RGB-D camera, that the mobile platform consists of a differential drive vehicle and that the velocity of each wheel can be measured.

This chapter contains a description of the problem along with a definition of the system kinematics in Section 2.1, as well a short observability analysis. Afterwards, a derivation of the Kalman filter discrete and continuous case (the latter being denominated by Kalman-Bucy filter) is described in Section 2.3.

2.1 Problem Description

The environment under study in this work is depicted in Fig. 2.1, where the frame $\{I\}$ is fixed to Earth, which is considered to be stationary, thus making $\{I\}$ an inertial frame. Our vehicle is defined by frame $\{B\}$ and is hence called the sensor-based frame or body-fixed frame. Both frames $\{I\}$ and $\{B\}$ are defined by the orthonormal basis $\{{}^I\mathbf{i}_I, {}^I\mathbf{j}_I\} \in \mathbb{R}^2$ and $\{{}^I\mathbf{i}_B, {}^I\mathbf{j}_B\} \in \mathbb{R}^2$, respectively.

In order to transform a position written in the body-fixed frame $\{B\}$ into a position written in the inertial frame $\{I\}$, a translation and a rotation need to be executed. The translation is defined by the body-fixed frame position in the inertial frame $\mathbf{p}_B^I$, and so the landmark position in both frames follows (2.1).

$$\mathbf{p}_B^I(t) + {}^I\mathbf{p}_l^B(t) = \mathbf{p}_l^I(t) \tag{2.1}$$

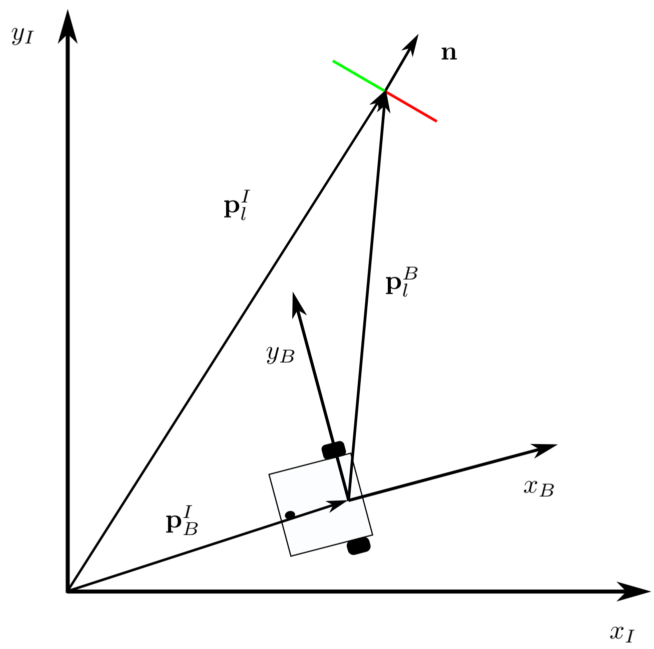


Figure 2.1: Reference frames.

where ${}^I\mathbf{p}_l^B(t) \in \mathbb{R}^2$ is the landmark position in the body-fixed frame, expressed in the inertial frame and $\mathbf{p}_l^I(t)$ is the landmark position in the inertial frame, expressed in the inertial frame. The landmark position in the body fixed frame $\mathbf{p}_l^B(t) \in \mathbb{R}^2$ is the objective and will henceforth, for a matter of simplicity, be denoted as $\mathbf{e}(t)$.

The rotation matrix from $\{B\}$ to $\{I\}$ that completes (2.1) is denoted by ${}^I\mathbf{R}_B(t) \in SO(2)$. This belongs to the special orthogonal group, i.e the group of matrices that preserves the inner product of two transformed vectors, thus preserving their length and relative orientation. This rotation matrix respects the following properties.

- It has a unitary determinant: $\det\left({}^I\mathbf{R}_B(t)\right) = 1$,
- It creates a correspondence between representations in $\{B\}$ and $\{I\}$,${}^I\mathbf{e}(t) = {}^I\mathbf{R}_B(t){}^B\mathbf{e}(t)$,
- The inverse of this rotation is its transpose, ${}^I\mathbf{R}_B(t)^{-1} = {}^I\mathbf{R}_B(t)^T$ and, by definition, it transforms vectors from $\{I\}$ to $\{B\}$:$\mathbf{e}(t) = {}^I\mathbf{R}_B(t)^T{}^I\mathbf{e}(t)$
- The rotation kinematics is given by

$$
{}^I\dot{\mathbf{R}}_B(t) = \mathbf{S}(\omega){}^I\mathbf{R}_B(t), \tag{2.2}
$$

where

$$
\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix},
$$

and $\omega \in \mathbb{R}$ is the angular velocity of the body-fixed frame. The rotation ${}^I\mathbf{R}_B(t)$ will henceforth be denoted as $\mathbf{R}(t)$ for a matter of simplicity.

It is straightforward to show that the inverse rotation follows a similar expression to (2.3) by taking the derivative on bot sides of $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ and making the necessary substitutions.

$$\dot{\mathbf{R}}^T(t) = -\mathbf{S}(\omega)\mathbf{R}^T(t) \tag{2.3}$$

After completely defining the rotation matrix, the position kinematics of the robot in the reference of inertia is possible to be expressed by

$$\dot{\mathbf{p}}(t) = {}^I\mathbf{R}_B(t)\mathbf{u}(t) \tag{2.4}$$

where $\mathbf{u}(t) = [v(t) \quad 0]^T$ and $v(t) \in \mathbb{R}$ is the robot velocity in the body-fixed frame.

2.2 Linear Time-Varying Kinematics

In this section the kinematics of the robot will be presented, and in this particular case two separate systems are considered for the position and attitude, and by doing this, linear kinematics on the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \end{cases} \tag{2.5}$$

are found, allowing to take advantage of the stability and optimality properties of the celebrated Kalman Filter. Each of the kinematic systems will have its estimator and those will be explained in the following sections.

2.2.1 Position Kinematics

The goal of this work is to have a localization system working in the sensor-based frame, as stated above, and so we need to be able to express $\dot{\mathbf{e}}(t)$ for the position of the landmark kinematic derivation. By expressing the product of (2.1) by $\mathbf{R}^T$ we get the position $\mathbf{e}$ expressed in order of $\mathbf{p}_l$ and $\mathbf{p}$ each corresponding to the landmark position and $\{B\}$ position in $\{I\}$ (respectively $\mathbf{p}_l^I$ and $\mathbf{p}_B^I$)

$$\mathbf{e}(t) = \mathbf{R}(t)^T(\mathbf{p}_l(t) - \mathbf{p}(t)). \tag{2.6}$$

Taking the time derivative $\dot{\mathbf{e}}(t)$

$$\dot{\mathbf{e}} = \dot{\mathbf{R}}^T(t)(\mathbf{p_l}(t) - \mathbf{p}(t)) + \mathbf{R}^T(t)(\dot{\mathbf{p}}_l(t) - \dot{\mathbf{p}}(t)), \tag{2.7}$$

and considering that the landmark will be static in the inertial reference system, the term $\dot{\mathbf{p}}_l(t)$ will be dropped, and by using (2.6) and (2.4) in (2.7) we get

$$\dot{\mathbf{e}} = -\mathbf{S}(\omega)\mathbf{R}^T(t)(\mathbf{p}_l(t) - \mathbf{p}(t)) - \mathbf{R}^T(t)\dot{\mathbf{p}}(t). \tag{2.8}$$

By further using the substitutions of (2.6) and (2.4) we will get the simplified equation

$$\dot{\mathbf{e}}(t) = -\mathbf{S}(\omega)\mathbf{e}(t) - \mathbf{u}(t), \tag{2.9}$$

where $\mathbf{u}(t) = [v(t)\ \ 0]$. It can be further assumed that the common mode velocity $v(t)$ can suffer from a biased measurement. The velocity could then be expressed as $v(t) = \bar{v}(t) + b$ where b is the constant or slow varying slippage and $\bar{v}(t)$ is the measured linear velocity, while $v(t)$ is the true linear velocity. If the state vector is $\mathbf{x}(t) = [\mathbf{e}^T(t)\ b(t)]^T$ then the matrix expression for the kinematics of the position will be given by

$$\dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} 0 & \omega & -1 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}^{\mathbf{x}}(\omega(t))} \mathbf{x}(t) + \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}^{\mathbf{x}}} \bar{v}(t) + \mathbf{v}(t), \tag{2.10}$$

where $\mathbf{v}(t) \in \mathbb{R}^3$ is the white plant noise caused by the uncertainty in our model and follows the following properties:

$$E[\mathbf{v}(t)] = 0, \quad \forall t \in \mathbb{R}, \tag{2.11}$$

$$E[\mathbf{v}(t)\mathbf{v}^T(\tau)] = \mathbf{Q}\delta(t-\tau). \tag{2.12}$$

Seeing as the position of the landmark in $\{B\}$ is available through measurements taken with an RGB-D camera, the output equation takes the form

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\mathbf{C}^{\mathbf{x}}} \mathbf{x} + \mathbf{w}(t), \tag{2.13}$$

where $\mathbf{w}(t)$ represents the noise generated by the camera sensor as well as the detection algorithm and has similar properties of the plant noise

$$E[\mathbf{w}(t)] = 0, \quad \forall t \in \mathbb{R}, \tag{2.14}$$

$$E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{R}\delta(t-\tau), \quad \forall t, \tau \in \mathbb{R}. \tag{2.15}$$

It is also assumed that the plant and sensor noise are uncorrelated, i.e.

$$E[\mathbf{w}(\eta)\mathbf{v}(\tau)] = 0, \quad \forall \eta, \tau \in \mathbb{R}. \tag{2.16}$$

The continuous system is defined and an expression for $\mathbf{x}(t)$ can be found resorting to the variation of constants method

$$\mathbf{x}(t) = \mathbf{\Phi}^{\mathbf{x}}(t, t_0)\mathbf{x}(0) + \int_{t_0}^{t} \mathbf{\Phi}^{\mathbf{x}}(\tau, t_0)\left(\mathbf{B}^{\mathbf{x}} v_\tau + \mathbf{v}(t)\right) d\tau, \tag{2.17}$$

where $\mathbf{\Phi}^{\mathbf{x}}(t, t_0)$ is the transition matrix for system 2.10.

Definition 2.1 (Transition Matrix). *A transition matrix of a system such as 2.5, denoted as $\mathbf{\Phi}(t, t_0)$, maps the state $\mathbf{x}$ from time t_0 to a time t through the generic solution presented in 2.17, where $\mathbf{\Phi}(t, t_0)$ must satisfy*

$$\frac{\partial \mathbf{\Phi}(t, t_0)}{\partial t} = \mathbf{A}(t)\mathbf{\Phi}(t, t_0). \tag{2.18}$$

The notions transmitted by Definition 2.1 and equation (2.17) allow for the derivation the discrete counterpart of system (2.10). Making use of 2.18, the transition matrix takes the form

$$\begin{aligned} \mathbf{\Phi}^{\mathbf{x}}(t, t_0) &= e^{\int_{t_0}^{t} \mathbf{A}(\tau)d\tau} \\ &= \begin{bmatrix} \cos(\theta - \theta_0) & \sin(\theta - \theta_0)) & -\frac{t-t_0}{\theta-\theta_0}\sin(\theta - \theta_0) \\ -\sin(\theta - \theta_0) & \cos(\theta - \theta_0) & \frac{t-t_0}{\theta-\theta_0}(1 - \cos(\theta - \theta_0)) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.19}$$

Considering now the non-homogeneous solution, and assuming that the velocity vector $\mathbf{u}_t$ will be constant between each sampling time we can find the expression for $\mathbf{x}_k = \mathbf{x}(t_k)$

$$\mathbf{x}_k = \mathbf{\Phi}^{\mathbf{x}}_k(\omega_k)\mathbf{x}_{k-1} + \mathbf{G}^{\mathbf{x}}_k v_{k-1} + \mathbf{v}_k \tag{2.20}$$

where $\mathbf{v}_k$ is the discrete white noise, $\mathbf{G}_k$ is the discrete input matrix

$$\mathbf{G}^{\mathbf{x}}_k = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}^{\mathbf{x}}(\tau, t_{k-1})\mathbf{B}^{\mathbf{x}} d\tau = \begin{bmatrix} -\frac{\sin \omega_{k-1} T_k}{\omega_{k-1}} \\ \frac{1-\cos \omega_{k-1} T_k}{\omega_{k-1}} \\ 0 \end{bmatrix} \tag{2.21}$$

. where $T_k = t_k - t_{k-1}$, being $k = 0, 1, 2...\infty$. The relation between the plant noise in the continuous system $\mathbf{Q}$ and the discrete equivalent $\mathbf{Q}_k$ one can be found through (2.23), where T_k is considered small and therefore $\mathbf{\Phi}(t_0, t_0 + T_k) \to \mathbf{I}$.

$$\mathbf{Q}_k = E\left[\mathbf{v}_k \mathbf{v}_k^T\right] \underset{\Delta t \to 0}{\approx} \int\int_{\Delta t} E\left[\mathbf{v}(\tau)\mathbf{v}(\eta)^T\right] d\tau d\eta. \tag{2.22}$$

By taking (2.12) and substituting it on (2.23), the approximate form of $\mathbf{Q}_k$ is found

$$\mathbf{Q}_k \underset{\Delta t \to 0}{\approx} \mathbf{Q} T_k \tag{2.23}$$

Similarly there is also a relationship between $\mathbf{R}_k$ and $\mathbf{R}$ coming from the correspondence

$$\mathbf{y}_k = \frac{1}{T_k}\int_{t_k}^{t_{k-1}} \mathbf{y}(\tau)d\tau. \tag{2.24}$$

By then taking (2.13) and approximating $\mathbf{x}(t)$ during small time-steps

$$\mathbf{y}_k \underset{\Delta t \to 0}{\approx} \mathbf{C}\mathbf{x}_k + \int\limits_{\Delta t} \mathbf{w}(\tau) d\tau, \tag{2.25}$$

leading to the discrete sensor noise covariance matrix

$$E\left[\mathbf{w}_k \mathbf{w}_k^T\right] = \mathbf{R}_k = \frac{1}{\Delta t} \int \int\limits_{\Delta t} E\left[\mathbf{w}(\tau)\mathbf{w}(\eta)^T\right] d\tau d\eta. \tag{2.26}$$

Using (2.15) and integrating yields

$$\mathbf{R}_k = \frac{\mathbf{R}}{\Delta t}. \tag{2.27}$$

We can then see that the covariance goes towards ∞ as the sampling frequency rises. In order for the entire state vector $\mathbf{x}$ to be able to be estimated, the system needs to be observable, since the observation matrix $\mathbf{C}$ does not give information about all the states. For the case of LTV systems, the concept of observability Gramian must be introduced.

Definition 2.2. ***Observability Gramian.*** *A system is considered to be observable in the time interval $t \in [t_0, \quad t_f]$ if and only if the observability gramian defined as*

$$\mathbf{W_O}(t_1, t_0) = \int_{t_0}^{t_f} \mathbf{\Phi}^T(\tau, t_0)\mathbf{C}^T(\tau)\mathbf{C}(\tau)\mathbf{\Phi}(\tau, t_0) d\tau, \tag{2.28}$$

is non-singular.

Using 2.28 we get that for the pair $\mathbf{A}^\mathbf{x}$ and $\mathbf{C}^\mathbf{x}$ the observability gramian takes the form

$$\mathbf{W_O}(t_1, t_0) = \int_{t_0}^{t_1} \begin{bmatrix} 1 & 0 & -\frac{\tau - t_0}{\theta - \theta_0}\sin(\theta - \theta_0) \\ 0 & 1 & \frac{\tau - t_0}{\theta - \theta_0}(\cos(\theta - \theta_0) - 1) \\ -\frac{\tau - t_0}{\theta - \theta_0}\sin(\theta - \theta_0) & \frac{\tau - t_0}{\theta - \theta_0}(\cos(\theta - \theta_0) - 1) & 2(\frac{\tau - t_0}{\theta - \theta_0})^2(1 - \cos(\theta - \theta_0)) \end{bmatrix} d\tau, \tag{2.29}$$

which can be particularized for the linear movement case ($\omega = 0$), resulting in

$$\mathbf{W_O}(t_1, t_0)|_{\omega=0} = \begin{bmatrix} \Delta t & 0 & -\Delta t \\ 0 & \Delta t & 0 \\ -\Delta t & 0 & -\Delta t^2 \end{bmatrix}.$$

The integral does not change the matrix rank and so both matrices show to have $rank(\mathbf{W_O}) = 3$ which is the same as the number states present in the state vector, thus rendering this system observable.

2.2.2 Attitude Kinematics

This section will focus on deriving the attitude estimator. Firstly the kinematic model will be described, giving then place for a brief observability analysis. The proposed kinematic system estimates explicitly the unavoidable angular slippage that may occur due to the lack of knowledge of the contact points with the floor as well as the lack of precision in the measurement of each wheel radius or asymmetries in mechanical construction. Here the angular slippage $s(t)$ is considered to be slow time-varying or even constant ($\dot{s} = 0$). The model that describes the attitude system is given by the kinematics and the output equations

$$\dot{\theta}(t) = \mathbf{A}^\theta \theta(t) + \mathbf{B}^\theta \omega(t) + \nu(t), \tag{2.30}$$

and

$$y(t) = \mathbf{C}^\theta \theta(t) + \eta(t), \tag{2.31}$$

respectively, where

$$\theta(t) = \begin{bmatrix} \psi(t) \\ s(t) \end{bmatrix}, \mathbf{A}^\theta = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \mathbf{B}^\theta = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \mathbf{C}^\theta = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

and $\nu(t)$ and $\eta(t)$ are the plant noise and output noise, respectively, both assumed to respect an unbounded normal distribution, i.e,

$$\nu(t) \sim N(\mathbf{0}, \mathbf{Q}_\theta)$$
$$\eta(t) \sim N(\mathbf{0}, \mathbf{R}_\theta).$$

This model addresses the landmark as if it is moving in the body reference system and so $\psi(t)$ represents the landmark attitude in it, which as stated before, is under the assumption that it is possible to define a unique reference system in the said landmark, and that the camera is able to detect its orientation. Assuming a constant angular velocity between two sampling instants, the state transition equation for this linear time invariant system is

$$\theta_{k+1} = \mathbf{\Phi}^\theta(T_k)\theta_k + \mathbf{G}_k^\theta \omega_k + \nu_k, \tag{2.32}$$

in which ω_k is the measured angular velocity obtained using the command sent to the dual-motor driver, $\mathbf{\Phi}^\theta(T) = \exp(A^\theta T_k)$, $\mathbf{G}_k^\theta = \omega_k \int_0^{T_k} \mathbf{\Phi}^\theta(T_k - \tau) \left(\mathbf{B}^\theta\right)\right) d\tau$ and T_k is the time between samples k and $k+1$, a measured quantity, rather than a constant sampling period.

The state vector needs to be estimated by using the output equation, which in turn does not, apparently, give information about all the states, and so a simple observability analysis is carried out next.

Since the continuous system is LTI, (2.33) is sufficient to assess the observability of the system

$$\mathcal{O}_a = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.33}$$

The observability matrix verifies $rank(\mathcal{O}_a) = 2$, which renders the attitude state vector observable.

2.3 Kalman Filter

In order for a system to be controlled, an estimate of the state must be present at a stable time interval. Only taking the measurements of landmarks wouldn't provide it, since it often is not even available at the camera field of view, therefore some kind of filtering that fuses the odometry and these landmark observations is needed. This section of this work will focus on the derivation of the state vector estimator used which is an optimal linear estimator in the sense that it provides the minimum variance estimator by optimizing the computation of the Kalman gain matrix in either the discrete or continuous case, minimizes the MSE (mean square error) of the state vector and maintains the error mean value null. This estimation method is called *filtering* because it is able, in some extent, to smooth the measurements $\mathbf{z}_k$ or $\mathbf{z}(t)$ by rejecting some of the sensor noise.

$$\mathbf{z}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{w}(t) \tag{2.34}$$

The derivation of this filter is under the assumption that we are in the presence of a linear dynamic system, as is the case depicted in 2.2 and also that the measurements have a linear behaviour in respect to the state vector $\mathbf{x}(t)$ such as (2.34), but make no assumption about the nature of the probability distrubution for the process and measurement noise, respectively $\xi(t)$ and $\mathbf{w}(t)$ besides that their mean value is zero and that both noises are uncorrelated.

$$E\left[\mathbf{w}(\tau)\right] = 0, \quad \forall \tau \tag{2.35}$$

$$E\left[\mathbf{v}(\tau)\right] = 0, \quad \forall \tau \tag{2.36}$$

$$E\left[\mathbf{v}(\tau)\mathbf{w}(\eta)\right] = 0, \quad \forall \tau, \eta \tag{2.37}$$

2.3.1 Continuous Time Filter

The continuous time case of the Kalman Filter (referred to as Kalman-Bucy Filter) represents the optimal linear estimator for linear systems. Consider a continuous system such as

$$\dot{\mathbf{x}}_t = \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t + \mathbf{L}_t\xi_t. \tag{2.38}$$

The state vector estimate equation is assumed to be linear with the measurements

$$\dot{\hat{\mathbf{x}}}_t = \mathbf{F}_t\hat{\mathbf{x}}_t + \mathbf{G}_t\mathbf{u}_t + \mathbf{K}_t\mathbf{y}_t + \mathbf{v}_t. \tag{2.39}$$

The goal is to have a unbiased estimate of the state vector, which implies that the expected value of the estimation error must be always zero and consequently its derivative is also zero

$$E\left[\tilde{\mathbf{x}}_t\right] = 0 \tag{2.40}$$

$$\Rightarrow \frac{\partial}{\partial t} E\left[\tilde{\mathbf{x}}_t\right] = 0 \tag{2.41}$$

$$\Rightarrow E\left[\dot{\tilde{\mathbf{x}}}_t\right] = 0. \tag{2.42}$$

Having this in mind, we can write the equation $E\left[\dot{\tilde{\mathbf{x}}}_t\right]$ using (2.39) and a usual dynamic system equation such as the one depicted by (2.10) resulting in

$$E\left[\dot{\tilde{\mathbf{x}}}_t\right] = \mathbf{F}_t\tilde{\mathbf{x}}_t + \left[\mathbf{A}_t - \mathbf{F}_t - \mathbf{K}_t\mathbf{C}_t\right]\mathbf{x}_t \tag{2.43}$$

$$+ \left[\mathbf{B}_t - \mathbf{G}_t\right]\mathbf{u}_t - \mathbf{v}_t + \mathbf{L}_t\xi_t - \mathbf{K}_t\mathbf{y}_t. \tag{2.44}$$

The goal is that the expected value of this equation is null and so by applying the expected value operator $E(.)$ on each side we get

$$0 = \left[\mathbf{A}_t - \mathbf{F}_t - \mathbf{K}_t\mathbf{C}_t\right] E\left[\mathbf{x}_t\right],$$
$$+ \left[\mathbf{B}_t - \mathbf{G}_t\right]\mathbf{u}_t - \mathbf{v}_t,$$

which, taking into account that generally $E\left[\mathbf{x}_t\right] \neq 0$ and $\mathbf{u}_t \neq 0$, leads to the following attributions

$$\mathbf{F}_t = \mathbf{A}_t - \mathbf{K}_t\mathbf{C}_t \tag{2.45a}$$

$$\mathbf{G}_t = \mathbf{B}_t \tag{2.45b}$$

$$\mathbf{v}_t = 0 \tag{2.45c}$$

$$\tilde{\mathbf{x}}_{t_0} = 0 \Rightarrow \hat{\mathbf{x}}_{t_0} = \bar{\mathbf{x}}_0 \tag{2.45d}$$

With (2.45) the Kalman-Bucy state estimate equation can be written in (2.46). The dynamics of the Kalman-Bucy Filter estimated state vector mimic those of the real system, apart from the fact that it is not affected by plant noise, but is instead affected by an update term that transfers the difference between the measurements $\mathbf{y}(t)$ and the estimate of what those measurements would be $\hat{\mathbf{y}} = \mathbf{C}(t)\hat{\mathbf{x}}(t)$ weighted by the Kalman gain matrix $\mathbf{K}_t$ resulting in

$$\dot{\hat{\mathbf{x}}}_t = \mathbf{A}_t\hat{\mathbf{x}}_t + B\mathbf{u}_t + \mathbf{K}_t(\mathbf{y}_t - \mathbf{C}_t\hat{\mathbf{x}}_t). \tag{2.46}$$

The equation that will rule the dynamics of the error covariance matrix $\mathbf{P}_t$ still needs to be found, as well as the gain matrix $\mathbf{K}_t$ that will minimize its value. It is known that $\mathbf{P}_t = E\left[\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T\right]$ and by using (2.43) is straightforward that

$$\dot{\mathbf{P}}_t = \left[\mathbf{A}_t - \mathbf{K}_t\mathbf{C}_t\right]\mathbf{P}_t + \mathbf{P}_t^T\left[\mathbf{A}_t - \mathbf{K}_t\mathbf{C}_t\right]^T \tag{2.47}$$

$$+\mathbf{L}_t\mathbf{Q}_t\mathbf{L}_t^T + \mathbf{K}_t\mathbf{R}_t\mathbf{K}_t^T, \quad \mathbf{P}_{t_0} = \mathbf{P}_0. \tag{2.48}$$

The aim is to minimize $tr\left(\mathbf{P}_t\right)$ and with a similar train of thoughts and using similar laws we can set the derivative of the latter to zero and solving it for $\mathbf{K}_t$, thus finding $\mathbf{K}_t^*$, the optimal Kalman gain matrix

$$\mathbf{K}_t^* = \mathbf{P}_t\mathbf{C}_t^T\mathbf{R}_t^{-1} \tag{2.49}$$

Replacing this gain in (2.47) we get the widely known Ricatti differential equation, which dictates the behaviour of the error covariance matrix $\mathbf{P}_t^*$

$$\dot{\mathbf{P}}_t^* = \mathbf{A}_t\mathbf{P}_t^* + \mathbf{P}_t^*\mathbf{A}_t^T + \mathbf{L}_t\mathbf{Q}_t\mathbf{L}_t^T - \mathbf{P}_t^*\mathbf{C}_t^T\mathbf{R}^{-1}\mathbf{C}_t\mathbf{P}_t^*. \tag{2.50}$$

The unavailability of landmarks will make $\mathbf{C}_t$ a matrix of zeros, thus making the last term of both equations 2.46 and 2.50 drop, making the derivative's trace $tr\left(\dot{\mathbf{P}}_t^*\right)$ positive.

2.3.2 Discrete Time Filter

The discrete Kalman filter represents an iterative process that is able to estimate a dynamic state vector at each given time step according to its dynamic model and corrects it with observations by the sensor package. This iterative process is divided into two different steps, being those the prediction step, where the dynamics of the system are used to predict the state vector given the last one available, and the update step, where the current state is corrected by a quantity called the **innovation process** which consists of the difference between the actual measurements and the expected measurements weighted by the Kalman Gain. This section will show the derivation of this gain as well as the determination of the probability density function of the state vector $\mathbf{x}_k = \mathbf{x}(t_k)$. Assuming that we have an unbiased estimate $\hat{\mathbf{x}}_k$ at any given iteration (meaning that $E\left[\tilde{\mathbf{x}}_k\right] = E\left[\mathbf{x}_k - \hat{\mathbf{x}}_k\right] = 0$), and that the sensor package provided a measurement $\mathbf{y}_{k+1}$, we wish to find $\mathbf{x}_{k+1}$ such that

$$\hat{\mathbf{x}}_{k+1} = \mathbf{K}'_{k+1}\hat{\mathbf{x}}_k + \mathbf{K}_{k+1}\mathbf{y}_{k+1}. \tag{2.51}$$

We wish that the relation between $\mathbf{K}'_{k+1}$ and $\mathbf{K}_{k+1}$ is such that the estimate remains unbiased ($E\left[\tilde{\mathbf{x}}_{k+1}\right] = 0$). Deriving the expression for $\tilde{\mathbf{x}}_k + 1$ by using (2.34) and simultaneously adding and subtracting $\mathbf{K}'_{k+1}\hat{\mathbf{x}}_k$ we get

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{K}'_{k+1}\hat{\mathbf{x}}_k + \mathbf{K}_{k+1}\left(\mathbf{C}_{k+1}\mathbf{x}_{k+1} + \mathbf{w}_{k+1}\right) - \mathbf{x}_{k+1} + \mathbf{K}'_{k+1}\hat{\mathbf{x}}_k - \mathbf{K}'_{k+1}\hat{\mathbf{x}}_k. \tag{2.52}$$

The rearrangement of these terms gives

$$
\begin{aligned}
\tilde{\mathbf{x}}_{k+1} = \mathbf{K}'_{k+1}\left[\hat{\mathbf{x}}_k - \mathbf{x}_k\right] + \mathbf{K}_{k+1}\left(\mathbf{K}_{k+1}\mathbf{C}_{k+1}\Phi_{k+1} - \Phi_{k+1} + \mathbf{K}'_{k+1}\right)\mathbf{x}_k \\
+ \left(\mathbf{K}_k + 1\mathbf{C}_{k+1} - \mathbf{I}\right)\mathbf{w}_{k+1} + \mathbf{K}_{k+1}\mathbf{v}_k.
\end{aligned}
\tag{2.53}
$$

The expected value of (2.53) needs to be equal to zero, and giving the assumption that $E\left[\mathbf{x}_k - \hat{\mathbf{x}}_k\right] = 0$ and also that $\mathbf{v}$ and $\mathbf{w}$ have null means, this implies

$$\mathbf{K}_{k+1}\mathbf{C}_{k+1}\mathbf{\Phi}_{k+1} - \mathbf{\Phi}_{k+1} + \mathbf{K}'_{k+1} = 0 \tag{2.54}$$

$$\mathbf{K}'_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{\Phi}_{k+1}, \tag{2.55}$$

which leads to the state update equation

$$\hat{\mathbf{x}}_{k+1} = \mathbf{\Phi}_{k+1}\hat{\mathbf{x}}_k + \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\Phi_{k+1}\hat{\mathbf{x}}_k\right). \tag{2.56}$$

This equation is in its final form, however the value of $\mathbf{K}_{k+1}$ for which the covariance of the estimation error is minimized still needs to be found. This will be achieved by minimizing the trace of the estimation error covariance matrix $tr(\mathbf{P}_{k+1|k+1})$, being $\mathbf{P}_{k+1|k+1} = E\left[\tilde{\mathbf{x}}_{k+1|k+1}\tilde{\mathbf{x}}^T_{k+1|k+1}\right]$ [1]. This needs to be computed iteratively so we would expect that $\mathbf{P}_{k+1}$ would be related to $\mathbf{P}^-_{k+1}$ and P_k which are the covariance of the error of each estimate $\hat{\mathbf{x}}^-_{k+1}$ and $\hat{\mathbf{x}}_k$ respectively, where $\mathbf{x}^-_k$ is the predicted state vector at any given k time-step before the measurements $\mathbf{y}_k$ are considered

$$\hat{\mathbf{x}}^-_{k+1} = \mathbf{\Phi}_{k+1}\hat{\mathbf{x}}_k. \tag{2.57}$$

By using (2.57) and (2.56) follows

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{\Phi}_{k+1}\tilde{\mathbf{x}}_k + \mathbf{w}_{k+1}, \tag{2.58}$$

which leads to the covariance of the estimate error prediction equation

$$
\begin{aligned}
\mathbf{P}^-_{k+1} = & \\
&= E\left\{\tilde{\mathbf{x}}_{k+1}\tilde{\mathbf{x}}^T_{k+1}\right\} \\
&= \mathbf{\Phi}_{k+1}E\left\{\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}^T_k\right\}\mathbf{\Phi}^T_{k+1} + E\left\{\mathbf{w}_{k+1}\mathbf{w}^T_{k+1}\right\} \\
&= \mathbf{\Phi}_{k+1}\mathbf{P}_k\mathbf{\Phi}^T_{k+1} + \mathbf{Q}_{k+1}.
\end{aligned}
\tag{2.59}
$$

By using the update part of (2.56) found before

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}^-_{k+1} - \mathbf{K}_{k+1}\left(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\hat{\mathbf{x}}^-_{k+1}\right) \tag{2.60}$$

[1] The notation $\mathbf{Y}_{t_2|t_1}$ denotes that the variable $\mathbf{Y}$ is being defined at time t_2 using observation until time t_1. Since all computations in this work are iterative, for a matter of simplicity $\mathbf{Y}_{k|k-1}$ will be denoted as $\mathbf{Y}^-_k$, while $\mathbf{Y}_{k|k}$ is simple denoted as $\mathbf{Y}_k$.

in the definition of estimate error $\tilde{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}$ gives

$$\tilde{\mathbf{x}}_{k+1} = [\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}] + \mathbf{K}_{k+1}\mathbf{y}_{k+1} - \mathbf{x}_{k+1} \tag{2.61}$$

$$= \left[\hat{\mathbf{x}}_{k+1}^{-} - \mathbf{x}_{k+1}\right] - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\left[\hat{\mathbf{x}}_{k+1}^{-} - \mathbf{x}_{k+1}\right] + \mathbf{K}_{k+1}\mathbf{v}_{k+1} \quad = [\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]\,\tilde{\mathbf{x}}_{k+1}^{-} - \mathbf{K}_{k+1}\mathbf{v}_{k+1}. \tag{2.62}$$

When taking the expected value of the last expression in both sides, the covariance of the estimate error update equation follows

$$\begin{aligned} \mathbf{P}_{k+1} &= E\left[\tilde{\mathbf{x}}_{k+1}\tilde{\mathbf{x}}_{k+1}^{T}\right] \\ &= [\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]\,\mathbf{P}_{k+1}^{-}\,[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]^{T} + \mathbf{K}_{k+1}\mathbf{R}_{k+1}\mathbf{K}_{k+1}^{T}. \end{aligned} \tag{2.63}$$

Now that all the equations for state and error covariance matrix update and prediction are derived in order of $\mathbf{K}_{k+1}$, the value of the latter for which the trace of $\mathbf{P}_{k+1}$ is minimum must be found. Using the mathematical rules applied to the trace of matrices we can find the expression for $tr\left(\mathbf{P}_{k+1}\right)$. Factorizing (2.63) and then applying the trace operator we get

$$\begin{aligned} \mathbf{P}_{k+1} &= (\mathbf{I} - \mathbf{K})\,\mathbf{P}\,(\mathbf{I} - \mathbf{K})^{T} + \mathbf{K}\mathbf{R}\mathbf{K}^{T} \\ &= \mathbf{P} - \mathbf{K}\mathbf{C}\mathbf{P} - \mathbf{P}\mathbf{C}^{T}\mathbf{K}^{T} + \mathbf{K}\mathbf{C}\mathbf{P}\mathbf{C}^{T}\mathbf{K}^{T} + \mathbf{K}\mathbf{R}\mathbf{K}^{T} \end{aligned}$$

$$tr\left(\mathbf{P}_{k+1}\right) = tr\left(\mathbf{P}\right) - 2tr\left(\mathbf{K}\mathbf{C}\mathbf{P}\right) + tr\left(\mathbf{K}\left(\mathbf{C}\mathbf{P}\mathbf{C}^{T}\right)\mathbf{K}^{T}\right) + tr\left(\mathbf{K}\mathbf{R}\mathbf{K}^{T}\right), \tag{2.64}$$

where the shorthand operators $\mathbf{P}$, $\mathbf{K}$, $\mathbf{C}$ and $\mathbf{R}$ replaced $\mathbf{P}_{k+1}^{-}$, $\mathbf{K}_{k+1}$, $\mathbf{R}_{k+1}$ and $\mathbf{C}_{k+1}$ respectively for an easier reading. Simplifying (2.64) and taking its derivative in respect to $\mathbf{K}$ we get

$$\frac{\partial tr\left(\mathbf{P}_{k+1}\right)}{\partial \mathbf{K}} = -2\mathbf{P}\mathbf{C}^{T} + 2\mathbf{K}\mathbf{C}\mathbf{P}\mathbf{C}^{T} + 2\mathbf{K}. \tag{2.65}$$

Setting this equation equal to zero and solving in order to $\mathbf{K}$ yields the value of it for which $\mathbf{P}_{k+1}$ is minimum.

$$\mathbf{K} = \mathbf{P}\mathbf{C}^{T}\left(\mathbf{C}\mathbf{P}\mathbf{C}^{T} + \mathbf{R}\right)^{-1}. \tag{2.66}$$

Now that the optimum Kalman gain is derived, the equation for updating the error covariance matrix can be simplyfied

$$\mathbf{P}_{k+1} = [\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]\,\mathbf{P}_{k+1}^{-}. \tag{2.67}$$

However shorter this expression might be, it often presents numerical problems. This being the case, the implemented equation will be the one depicted in (2.63). The iterative process was all derived and so a brief summary is now shown, being that the process is divided into 2 stages: prediction, where only the dynamics of the state vector influence the estimate and the error covariance matrix, and update, when the mentioned quantities are influenced by the measurement at the target time of the prediction step, thus correcting the belief.

Prediction Step

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1}$$

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_k \mathbf{P}_{k-1|k-1} \mathbf{\Phi}_k^T + \mathbf{Q}_k$$

Update Step

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \left[\mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T \right]^{-1}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_{k|k-1})$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$$

2.4 Filtering Solution

This section briefly describes both Kalman Filters used in the attitude and position estimation and the connection between them. The attitude estimate dynamics follows the usual Kalman dynamics

$$\hat{\theta}_k = \mathbf{\Phi}^\theta(T)\hat{\theta}_{k-1} + \mathbf{G}^\theta(T)\omega_k + \mathbf{K}_k^\theta \left[\bar{\psi}_k - \hat{\psi}_k \right], \tag{2.68}$$

where $\mathbf{K}_k^\theta$ is the Kalman gain computed for the attitude system in time t_k. The Kalman gain is dynamically computed at each iteration, i.e., every time a landmark is detected by the vision algorithm

$$\mathbf{K}_k^\theta = \mathbf{P}_k^\theta \mathbf{C}^{\theta T} \left[\mathbf{C}^\theta \mathbf{P}_k^\theta \mathbf{C}^{\theta T} \right]^{-1}, \tag{2.69}$$

where $\mathbf{P}^\theta{}_k = E\left[\tilde{\theta}\,\tilde{\theta}^T\right]$ is the estimation error covariance matrix and $\tilde{\theta}$ is the estimation error. This matrix is also dynamically computed by (2.70a) whenever information from the wheels is available and (2.70b) when the landmark is detected by the image processing module

$$\mathbf{P}_{k|k-1}^\theta = \mathbf{\Phi}_k^\theta \mathbf{P}_{k-1|k-1}^\theta \mathbf{\Phi}_k^{\theta T} + \mathbf{Q}_k^\theta, \tag{2.70a}$$

$$\mathbf{P}_{k|k}^\theta = (\mathbf{I} - \mathbf{K}_k^\theta \mathbf{C}_k^\theta) \mathbf{P}_{k|k-1}^\theta (\mathbf{I} - \mathbf{K}_k^\theta \mathbf{C}_k^{\theta T}) + \mathbf{K}_k^\theta \mathbf{R}_k^\theta \mathbf{K}_k^{\theta T}. \tag{2.70b}$$

The position estimator will perform a sub-optimal estimation since the angular velocity that parametrizes the state transition matrix of this system is meant to take into account the estimated angular slippage $\hat{s}$ since $\hat{\omega} = \bar{\omega} + \hat{s}$. Nevertheless, the equation that describes the estimate dynamics is similar to the one used for the attitude system estimate and is expressed by (2.71)

$$\hat{\mathbf{x}}_k = \mathbf{\Phi}^\mathbf{x}(\hat{\omega}_k, T)\hat{\mathbf{x}}_{k-1} + \mathbf{B}^\mathbf{x} v_k + \mathbf{K}_k^\mathbf{x} \left[\bar{\mathbf{e}}_k - \hat{\mathbf{e}}_k \right]. \tag{2.71}$$

The Kalman gain for this estimator is calculated in the exact same way as in the attitude estimator, using (2.69), only using the appropriate matrices $\mathbf{R}^{\mathbf{x}}$, $\mathbf{C}^{\mathbf{x}}$, $\mathbf{P}^{\mathbf{x}}$ and $\mathbf{Q}^{\mathbf{x}}$.

Chapter 3

Nonlinear Control Design

Linear control design is a mature subject with many powerful methods and a variety of successful implementation cases. However, the scientific community has, in some areas, been showing a growing interest in the development of methodologies for the design and analysis of nonlinear systems. The world is inherently nonlinear and some systems do not confer desired characteristics after linearisation, since linear control methods are often only valid for small range operations, thus exhibiting poor or unstable behaviour under large range operations due to not being able to compensate for the nonlinearities present. Also, robots performing under linear control designs often have to keep a slower rate of work so as to not cause the nonlinearities to have a significant effect in the motion of it, since the controller is not designed to cope with them. Surprisingly, non-linear control systems design can often be simpler than linear control design, since they can address some of the physics in the plant explicitly, fact that often provides an intuitive flavour to nonlinear control design and analysis, as will be evident throughout the chapter.

The most used stability concept was brought by a Russian mathematician Aleksandr Lyapunov, who developed the idea of equilibrium state stability and deduced powerful methods for the analysis of nonlinear systems. Lyapunov Stability Theory, comprises two main methods: the linearisation method which studies the stability of the linearised systems and tries to draw conclusions in the local stability of the original non-linear system; and the direct method, which analyses the time variation of a scalar function of the system state vector in order to determine its stability properties. Special emphasis will then be given to the direct method, due to the nature of the present work.

3.1 Lyapunov Stability

Nonlinear systems can be classified in two major groups: **non-autonomous** and **autonomous** systems, depending on whether it, respectively, explicitly depends on time or not. Here the focus will be on autonomous systems. Non-autonomous behaviours can be originated either in the plant or in the control law of a system such as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \tag{3.1}$$

If the control law is to take the form $\mathbf{u} = \mathbf{g}(\mathbf{x}, t)$ then the system will exhibit non-autonomous behaviour ($\mathbf{f}^*(\mathbf{x}, t)$), but if the time dependency drops, then (3.1) takes the simpler form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}). \tag{3.2}$$

Consider now an autonomous system such as (3.2) and let $\mathbf{x}^*$ be an equilibrium point of the system, i.e., $\mathbf{f}(\mathbf{x}^*) = 0$. This definition of equilibrium point entails that once $\mathbf{x} = \mathbf{x}^*$, it will remain so for all future time. Without any loss of generality, any *specific* equilibrium point can be transformed into the origin by taking $\mathbf{y} = \mathbf{x} - \mathbf{x}^*$. Actually, once can notice that $\mathbf{y} = \mathbf{0}$ is an equilibrium point of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{y} + \mathbf{x}^*)$. If the problem is the stability of a motion, with a trajectory defined as $\mathbf{x}^*(t)$, corresponding to an initial condition $\mathbf{x}^*(0) = \mathbf{x}_0^*$, one can also study the stability of the *disturbance* of the movement relative to the desired trajectory, turning the problem again into the study of the stability of a system around the origin, since for $\mathbf{r}(t) = \mathbf{x}(t) - \mathbf{x}^*(t)$ the differential equation becomes

$$\dot{\mathbf{r}} = \mathbf{f}(\mathbf{x}^*(t) + \mathbf{r}) - \mathbf{f}(\mathbf{x}^*(t)) = \mathbf{g}(\mathbf{r}, t),$$

thus inheriting the non-autonomous behaviour due to the progression of the desired trajectory with time. The stability of a non-linear system, unlike their linear counterparts, is not analysed for the system as a whole, but rather for each equilibrium point, since non-linear systems can have several of such.

Definition 3.1 (Lyapunov Stability)**.** *The equilibrium point $\mathbf{x}^* = 0$ is considered **stable** if, for any $R > 0$, there exists $r(R) > 0$ such that*

$$\|\mathbf{x}(0)\| < r \implies \|\mathbf{x}(t)\| < R, \quad \forall t > 0,$$

*and is otherwise **unstable**. Moreover, the equilibrium point is considered **asymptotically stable** if in addition to being stable, there exists some $r > 0$, such that*

$$\|\mathbf{x}(0)\| < r \implies \mathbf{x} \to 0 \quad as \quad t \to \infty,$$

*and it is considered **exponentially stable** if there exist two strictly positive numbers α and λ such that*

$$\forall t > 0, \quad \|\mathbf{x}(t)\| \leq \alpha \|\mathbf{x}(0)\| e^{-\lambda t},$$

in a ball $B_r = \{\mathbf{x} \in \mathbb{R}^n : \quad \|\mathbf{x}\| < r\}$.

Stability of an equilibrium point (stability in the Lyapunov sense) means that any trajectory starting arbitrarily close to the origin (equilibrium point) will remain arbitrarily close to it, or in other words, for any initial condition $\mathbf{x}(0) \in B_r$, if $\mathbf{x} \in B_R$ then the origin is *stable*. If the state converges to the origin as $t \to \infty$, then the stability is *asymptotic*, making B_r a *domain of attraction*, being that the set of all points for which this happens is *the* domain of attraction of the equilibrium point. Such domain can be written as $\Omega^* = \{\mathbf{x}(0) \in \mathbb{R}^n : \lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*\}$. Note that any equilibrium point that is stable but not asymptotically stable is denoted as *marginally stable*. Finally, if the convergence of the state to zero is exponentially fast, then the origin is *exponentially stable*. In many applications, local stability, as defined

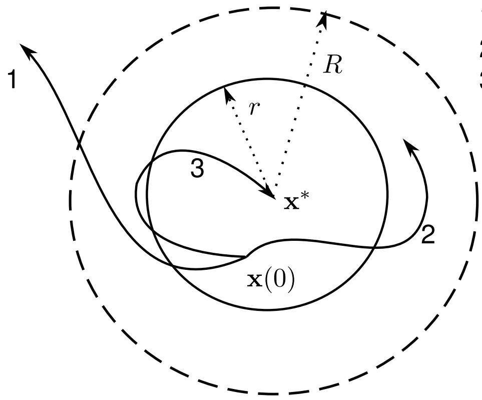


Figure 3.1: Lyapunov Stability

above, is not an enough strong property, meaning that it is desirable that the stability properties are maintained regardless of the initial conditions.

Definition 3.2 (Global stability)**.** *If the asymptotic (or exponential) stability holds for any initial conditions of the system, the equilibrium point is said to be **globally** asymptotically (or exponentially) stable.*

Another perspective of global stability is allowing for the set Ω^* to be the whole state space $\mathbb{R}^n$, while still holding the asymptotic or exponential stability. Note that marginal stability does not make sense in global terms.

3.1.1 Linearisation Method

The Lyapunov's linearisation method is, to some extent, a formalisation of the intuitive idea that a non-linear system behaves the same way as its linearised counterpart for small range motions near an equilibrium point. It also is used as a justification for the use of linear control techniques. Consider that, in the system (3.1), $\mathbf{f}(\mathbf{x})$ is continuously differentiable in $\mathbf{x}$ and $\mathbf{u}$ so that the linearisation of the system around an equilibrium point $\mathbf{x}^*$ (disregarding any terms with order higher than 1) is expressed by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.3}$$

where

$$\mathbf{A} = \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{(\mathbf{x}=\mathbf{x}^*,\mathbf{u}=\mathbf{u}^*)}, \quad \mathbf{B} = \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{(\mathbf{x}=\mathbf{x}^*,\mathbf{u}=\mathbf{u}^*)} \tag{3.4}$$

It is then usual to use this technique in order to apply linear control design strategies by choosing $\mathbf{u} = \mathbf{u}(\mathbf{x})$. Note that now a potential non-autonomous has become a closed-loop autonomous system, since $\mathbf{u}$ does not explicitly depend on time. Linearising the control law

$$\mathbf{u} \approx \left.\frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{x}}\right|_{(\mathbf{x}=\mathbf{x}^*)} \mathbf{x} = \mathbf{G}\mathbf{x},$$

yields the closed-loop system dynamics approximation

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{B}\mathbf{G})\,\mathbf{x},$$

which can also be acquired by linearising the original nonlinear closed-loop dynamics. The Lyapunov Linearisation method draws conclusions on the nonlinear system behaviour near the equilibrium point by analysing the stability properties of the linearised system.

Theorem 3.1 (Lyapunov's Linearisation Method)**.**

- *if the linearised system is strictly stable, the equilibrium point is asymptotically stable for the non-linear system.*
- *if the linearised system is unstable, the equilibrium point for the nonlinear system is also unstable.*
- *if the linearised system is marginally stable, nothing can be concluded by the linear approximation.*

Proof. the proof of the above theorem can be found in Chapter 4 of [31]. □

It is intuitive that for marginally stable linear systems one cannot draw conclusions on the stability properties of the original systems, since the disregarded higher order terms can have a significant effect on them. It is possible to have asymptotically stable systems of while the linearised approximation is only marginally stable. Although many times useful, the Lyapunov's linearisation method is has the limitability of a short range validity.

3.1.2 Lyapunov's Direct Method

Lyapunov's direct method, often called second method, is based on the idea that, in a conservative system, the total energy must converge to a minima, i.e., a stable equilibrium point, while the maxima of that energy correspond to an unstable equilibrium point. Therefore the stability of such systems can be analysed by examining the behaviour of a scalar (energy-like) function $V(\mathbf{x})$ (or $V(\mathbf{x}, t)$) of the system's state, without then requiring to know the specific solution of the dynamics equation.

Definition 3.3. *A scalar continuous function $V(\mathbf{x})$ is*

- *considered* ***locally positive definite*** *if $V(0) = 0$ and in a ball B_{R_0}, $V(\mathbf{x}) > 0$ for $\mathbf{x} \neq 0$.*
- *considered* ***locally positive semi-definite*** *if $V(0) = 0$ and in a ball B_{R_0}, $V(\mathbf{x}) \geq 0$ for $\mathbf{x} \neq 0$.*

Additionally, if $V(0) = 0$ and the above properties hold as $R_0 \to \infty$, then $V(\mathbf{x})$ is considered ***globally positive (semi-)definite****.*

It is straightforward to define negative definiteness since $V(\mathbf{x})$ is negative (semi-)definite if $-V(\mathbf{x})$ is positive (semi-)definite. Using this as a tool to study the time derivative of the scalar function $V(\mathbf{x})$ is the basis of the direct method. The derivative of this function is expressed by the chain rule

$$\dot{V}(\mathbf{x}) = \frac{\partial V(\mathbf{x})}{\partial t} + \frac{\partial V(\mathbf{x})}{\partial x}\dot{\mathbf{x}}. \tag{3.5}$$

By restricting the focus of this work to autonomous systems, (3.5) becomes the derivative of V along the system's trajectory

$$\dot{V}(\mathbf{x}) = \frac{\partial V(\mathbf{x})}{\partial t}\mathbf{f}(\mathbf{x}). \tag{3.6}$$

Such functions are called Lyapunov functions if, locally, they are positive-definite and their time derivative is negative semi-definite along the system trajectory.

Theorem 3.2 (Local Stability)**.** *If there exists a continuous function $V(\mathbf{x})$ with continuous first partial derivatives such that, locally in B_{R_0},*

1. *$V(\mathbf{x})$ is positive definite*

2. *$\dot{V}(\mathbf{x})$ is negative semi-definite*

then the origin equilibrium point is stable. If $\dot{V}(\mathbf{x})$ is actually negative definite, then it is asymptotically stable. Additionally, if B_{R_0} can be extended to the whole state-space and, combined with 1. we have

3. *$\dot{V}(\mathbf{x})$ is negative definite*

4. *$V(\mathbf{x}) \to \infty$ as $\|\mathbf{x}\| \to \infty$ (radial unboundedness)*

then the equilibrium at the origin is globally asymptotically stable.

Proof. proof of this theorem can be found in Chapter 3 of [32]. □

Note that these theorems are sufficiency theorems and that there can be numerous Lyapunov function candidates to evaluate the stability properties of a system, meaning that, if, for a particular choice of scalar function, the conditions for the theorem are not met, one cannot draw an conclusions regarding stability properties.

3.1.3 Invariance Principle

The negative definiteness of $\dot{V}$ can be restrictive, but for cases when $\dot{V}$ is only negative semi-definite, LaSalle's invariance theorem is the method of choice to prove asymptotic stability for autonomous systems. The central concept of the theorem is that of an invariant set which is the generalization of an equilibrium point.

Definition 3.4 (Invariant Set)**.** *A set G is said to be an* ***invariant set*** *if*

$$\mathbf{x}(0) \in M \implies \mathbf{x}(t) \in M, \forall t \in \mathbb{R}$$

That is, if a solution belongs to the set M at some instant, then it belongs to M for all future and past time. It is, however, a *positively invariant set*, if a solution only remains in M for all future time. Having introduced the concept of an invariant set, the enunciation of LaSalle's theorem follows.

Theorem 3.3 (LaSalle)**.** *Let $\Omega \subset D$ be a compact positively invariant set with respect to (3.2). Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(\mathbf{x}) \leq 0$ in Ω. Let E be the set of all points within Ω where $\dot{V}(\mathbf{x}) = 0$. Let M be the largest invariant set in E. Then every $\mathbf{x}(t)$ starting in Ω approaches M as $t \to \infty$.*

Proof. proof of the above theorem can be found in Chapter 4 in [31]. □

As stated before, LaSalle's theorem alleviates the need of negative definiteness of $\dot{V}$ for proving asymptotic stability and it also does not require $V(\mathbf{x})$ positive definite, unlike Theorem 3.2. The interest is sometimes to show that the solution $\mathbf{x} \to \mathbf{0}$ as $t \to \infty$, i.e., prove that the largest invariant set in E is actually the origin. This need gave origin to the following two corollaries.

Corollary 3.1. *Let $\mathbf{x} = \mathbf{0}$ be an equilibrium point of system (3.2). Let $V : D \to \mathbb{R}$ be a continuously differentiable positive definite function on a domain D that contains the origin $\mathbf{x} = \mathbf{0}$, such that $\dot{V}(\mathbf{x}) \leq 0$ in D. Let $S = \{\mathbf{x} \in D \quad | \quad \dot{V}(\mathbf{x}) = 0\}$ and suppose that the only solution that can stay identically in S is $\mathbf{x} = \mathbf{0}$. Then the origin is asymptotically stable.*

Corollary 3.2. *Let $\mathbf{x} = \mathbf{0}$ be an equilibrium point of system (3.2). Let $V : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable, radially unbounded, positive definite function such that $\dot{V}(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. Let $S = \{\mathbf{x} \in \mathbb{R}^n \quad | \quad \dot{V}(\mathbf{x}) = 0\}$ and suppose that the only solution that can stay identically in S is $\mathbf{x} = \mathbf{0}$. Then the origin is globally asymptotically stable.*

These corollaries are known as theorems of Barbashin and Krasovskii. Under the conditions of the above corollaries, requiring that E contains no trajectories other than the origin, and that, then if $V(\mathbf{x})$ is positive definite in a neighbourhood of the origin Ω^*, a domain of attraction of the equilibrium point is the largest connected region $\Omega_c = \{\mathbf{x} \in \Omega^* : V(\mathbf{x}) < c\}$.

3.2 Docking problem

Let the state vector to be controlled be $\mathbf{z} = [e_x \; e_y \; \psi]^T$, from (2.30) and (2.10) we get that its kinematics can be expressed as

$$\dot{\mathbf{z}} = \mathbf{f}_\omega(\mathbf{z})\omega + \mathbf{f}_v(\mathbf{z})v, \tag{3.7}$$

where

$$\mathbf{f}_\omega = \begin{bmatrix} e_y \\ -e_x \\ -1 \end{bmatrix}, \quad \mathbf{f}_v = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}.$$

The present work follows Lyapunov's direct method of finding a scalar energy-like function $V(\mathbf{z})$ and devise a control law $\mathbf{u}(\mathbf{z})$ that ensures the resulting closed-loop system is asymptotically stable (the goal is to park the vehicle in a position $\mathbf{z}^*$). This leads to a smooth and time invariant control law. However, a theorem developed by Brockett shown in [33], states that, for systems in the structure

$$\dot{\mathbf{z}} = \sum_{i=1}^{m} \mathbf{f}_i(\mathbf{z})u_i,$$

with vectors $\mathbf{f}_i(\mathbf{z})$ being linearly independent and continuously differentiable at a point z^*, then there exists a stabilization solution, with a smooth and time invariant feedback law, if and only if $m = n$, where n is the order of the system, meaning there need to be the same number of control parameters as the dimension of the state vector to be controlled. The system in (3.7) does not respect the last condition and clearly has $\mathbf{f}_\omega$ and $\mathbf{f}_v$ independent at the origin. This would then require the use of time-varying or discontinuous control laws in order to achieve the desired stabilization. Seeing as the need of stabilizing n states at a point $\mathbf{z}^*$ is still the objective, then only a system with singularities is of interest. With this in mind, a new system is proposed in [34]. The said system represented in (3.8) is based on a state vector that is isomorphic with the one in (3.7), characterized by the isomorphism $g : \mathbb{R}^3 \setminus \{\mathbf{0}\} \mapsto \mathbb{R}^3 \setminus \{\mathbf{0}\}$

$$\begin{cases} e = \sqrt{e_x^2 + e_y^2} \\ \alpha = atan(e_y/e_x) \\ \phi = atan(e_y/e_x) - \psi \end{cases},$$

leading to the kinematics

$$\begin{cases} \dot{e} = -v\cos\alpha \\ \dot{\alpha} = -\omega + v\frac{\sin\alpha}{e} \\ \dot{\phi} = v\frac{\sin\alpha}{e} \end{cases}. \tag{3.8}$$

The new state vector is depicted in Figure 3.2. Due to the singularity at the origin, Brockett's theorem no longer applies, since the regularity assumptions do not hold, and so the asymptotic stabilization of 3.8 is possible. One then cannot formally use the definition of equilibrium point to describe the origin, since it is now located in the frontier of the open set of validity of the system dynamics. The objective of the control law is then to asymptotically drive the system to $\mathbf{z_p}^* = [0 \quad 0 \quad 0]^T$ without attaining $e = 0$ in a finite time, where $\mathbf{z_p} = [e \quad \alpha \quad \phi]$ is henceforth the notation used for the new state vector. A simple

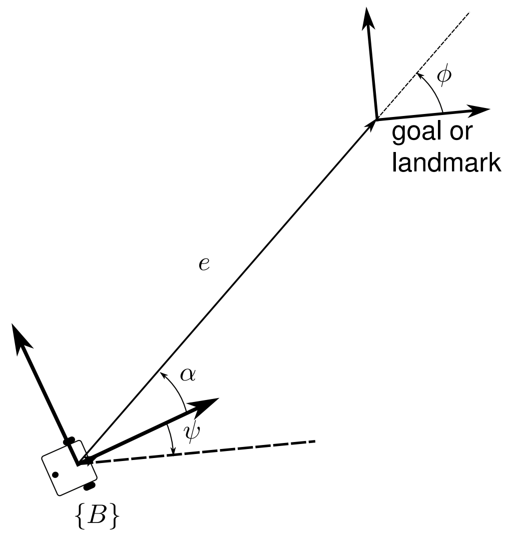


Figure 3.2: Depiction of new state vector.

choice for a candidate Lyapunov function is the often used quadratic error form

$$V(\mathbf{z}) = \underbrace{\frac{1}{2}\lambda e^2}_{V_1} + \underbrace{\frac{1}{2}\alpha^2 + \frac{1}{2}h\phi^2}_{V_2}, \quad \lambda, h > 0 \tag{3.9}$$

where λ and h are positive weighting constants that will later on help shape the control law. By separating the scalar function in two terms, we have that the first term refers to the error in distance to the target position, and the second term corresponds to the to a "alignment vector" error $[\alpha \quad h\phi]$. It is clear by now that the a candidate of a scalar function has been chosen and then a function $\mathbf{u}(\mathbf{z_p})$ will be derived in order for the behaviour of V along the trajectory of (3.8) to drive the state asymptotically to the origin. Taking then the derivative $\dot{V}$, given by

$$\begin{aligned}\dot{V} &= \lambda e\dot{e} + \left(\alpha\dot{\alpha} + h\phi\dot{\phi}\right) \\ &= \lambda ev\cos\alpha + \alpha\left[-\omega + v\frac{\sin\alpha(\alpha + h\phi)}{\alpha e}\right].\end{aligned} \tag{3.10}$$

The first term of (3.10) can be made non-positive by letting

$$v = \gamma e\cos\alpha, \quad \gamma > 0, \tag{3.11}$$

leading to

$$\dot{V}_1 = -\lambda\gamma\cos^2\alpha e^2 \leq 0. \tag{3.12}$$

This choice of linear velocity control law ensures that the validity of (3.8) throughout the parking problem, since V_1 is lower bounded and non-increasing, making it asymptotically converge to a non negative finite limit, thus ensuring e exhibits the same behaviour. The same strategy is applied to the second term, and so expression for the angular velocity control law is

$$\begin{aligned} \omega &= k\alpha + v\frac{\sin\alpha(\alpha + h\phi)}{\alpha e} \\ &\overset{(3.11)}{=} k\alpha + \gamma\frac{\cos\alpha\sin\alpha(\alpha+h\phi)}{\alpha}. \end{aligned} \tag{3.13}$$

The derivative of the total Lyapunov function then becomes

$$\dot{V} = -\gamma\left(\cos^2\alpha\right)e^2 - k\alpha^2 \leq 0, \tag{3.14}$$

which is negative semi definite. A fundamental result of calculus lets us establish that, given the non increasing nature of V and given that its lower boundedness by zero, then it converges to a non negative limit. This in turn, together with the radial unboundedness of V, assures the boundedness of the state trajectories for any bounded initial conditions. By then using LaSalle's Theorem 3.3, and noting that $E = \{\mathbf{z_p} \in \mathbb{R}^3 : \quad \dot{V}(\mathbf{z_p}) = 0\}$, inspection of (3.14) indicates that the state trajectory will converge to the largest invariant set M characterized by a state of the form $[0 \quad 0 \quad \phi]$. To prove that the origin is the only solution that can identically stay in E, the closed-loop kinematics system must be inspected

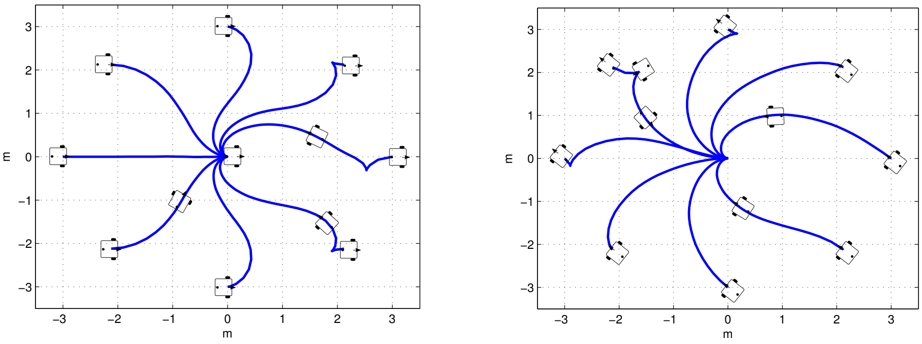
$$\begin{cases} \dot{e} = \gamma e\cos^2\alpha \\ \dot{\alpha} = -k\alpha - \gamma h\frac{\cos\alpha\sin\alpha}{\alpha}\phi & \qquad e > 0. \\ \dot{\phi} = \gamma\cos\alpha\sin\alpha \end{cases} \tag{3.15}$$

Given the convergence of e and α to zero, then both time derivates $\dot{e}$ and $\dot{\phi}$ converge to zero. The convergence of $\dot{\phi}$ allied with the boundedness of the trajectory asserts that ϕ must converge to a finite limit $\bar{\phi}$ over time. It then follows that $\dot{\alpha}$ tends to the limit $-\gamma h\bar{\phi}$. Given the bounded nature of the trajectory, the uniform continuity of $\dot{\alpha}$ follows, allowing for the use of Barbalat's Lemma which results in $\dot{\alpha}$ converging to zero, which in turn confirms that $\bar{\phi} = 0$.

Lemma 3.1 (Barbalat's Lemma)**.** *If the differential function $f(t)$ has a finite limit as $t \to \infty$, and if $\dot{f}$ is uniformly continuous, then $\dot{f} \to 0$ as $t \to \infty$.*

Proof. proof of this lemma can be found in Chapter 4 of [32]. □

This then proves that the largest invariant set M in E is the origin, thus making the origin globally asymptotically stable. Note that the objective is to have the vehicle dock in a certain station with positive linear velocity, but it is possible to obtain different trajectories by simply changing the goal objective (to for instance $\mathbf{z_p}^* = [0, \pm\pi, \pm\pi]$). A depiction of the trajectories performed by the system are depicted in Figure 3.3, where a simulation was performed with $\gamma = 3$, $h = 1$ and $k = 6$. As intended, the vehicle always arrives at the target location facing, which goes accordingly with state vector converging to the origin. An infinite number of trajectories are possible by modifying the parameters present in the feedback laws, namely h, k and γ, seeing as λ has no effect in the parking problem. These parameters



(a) $\psi(0) = 0$

(b) $\psi(0) = 3\pi/4$

Figure 3.3: Trajectories performed with $e(0) = 3$

have an effect on the rate of convergence of each of the three states, which can be studied by analysing the linearisation of (3.15)

$$\begin{bmatrix} \dot{e} \\ \dot{\alpha} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -\gamma & 0 & 0 \\ 0 & -k & -h\gamma \\ 0 & \gamma & 0 \end{bmatrix} \begin{bmatrix} e \\ \alpha \\ \phi \end{bmatrix}, \tag{3.16}$$

as well as the linearisation of the feedback law, in this study regarded as being an output

$$v = \gamma e, \tag{3.17}$$

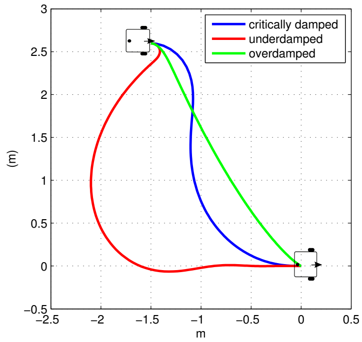
$$\omega = (k + \gamma)\,\alpha + h\gamma\phi. \tag{3.18}$$

The poles for the linearised system are $p_1 = -\gamma$, which refers to the behaviour of e and consequently of v, $p_{2,3} = 0.5\left(-k \pm \sqrt{k^2 - 4h\gamma^2}\right)$ and so it is easy to see that e converges to zero as $e^{-\gamma t}$ while α and ϕ converge with $e^{\sigma t}$, being $-\sigma$ the real part of the dominant pole of the decoupled linear α, ϕ system. We can then conclude that the curvature $c = \omega/v$ can only converge to zero if $\sigma > \gamma$, which is identical to say that the parameters h, γ and k must follow

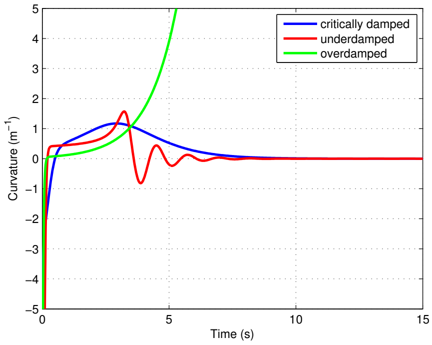
$$h > 1; \quad 2\gamma < k < (h + 1)\,\gamma. \tag{3.19}$$

Both α and ϕ poles are coupled and can be mostly manipulated by choosing different values for h and k. Figure 3.5 shows the trajectory behaviour for different sets of values chosen for the aforementioned parameters. Bringing into account the designed controller will be receiving estimates of the states through the estimator described in Chapter 2, it is straightforward to come to the conclusion that a compromise must be met between the time progression of α, being it ideally as low as possible, and the curvature of the trajectory imposed by the controller, which desirably zero when approaching the parking goal. By analysing the curvature expression

$$c = \frac{\omega}{v} = \frac{(k/\gamma + 1)\,\alpha + h\phi}{e}, \tag{3.20}$$



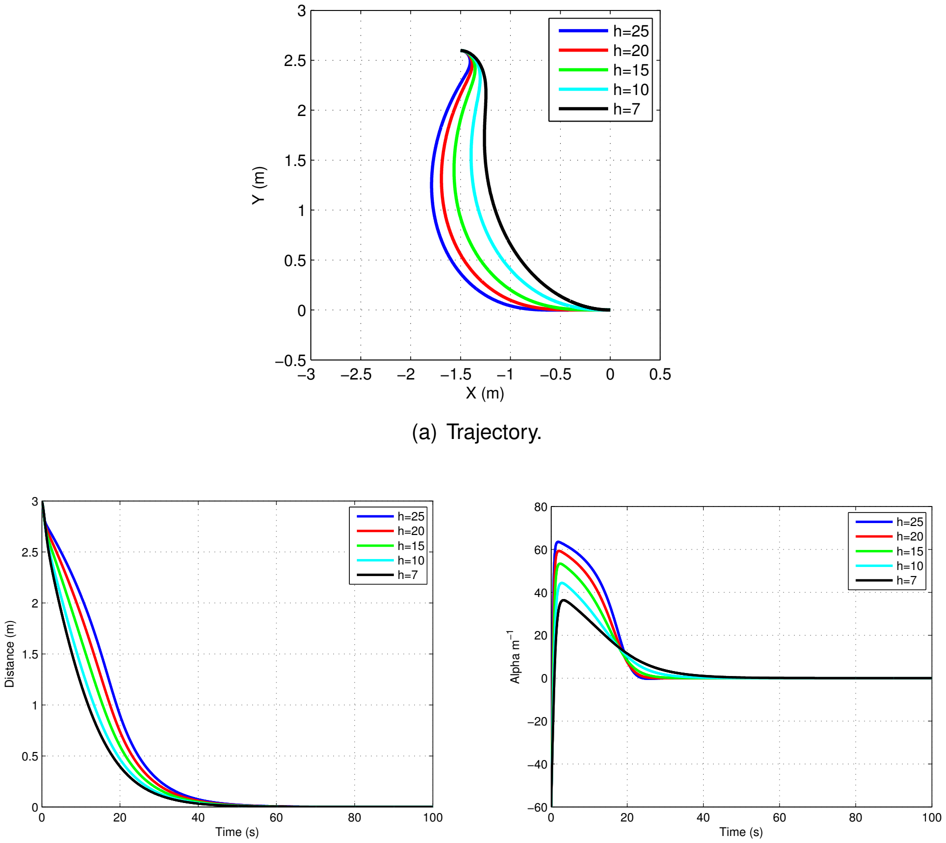
(a) Trajectory.



(b) Curvature

Figure 3.4: Critically damped ($h = 4, k = 4, \gamma = 1$), underdamped ($h = 30, k = 4, \gamma = 1$) and overdamped ($h = 4, k = 30, \gamma = 1$) trajectories.

one realises the curvature remains constant if k/γ is kept constant. Experimenting on several values of h gives an intuitive idea of the ideal parameter relation between h and k/γ. Figure 3.5 depicts the trajectories resultant for $k/\gamma = 8$



(a) Trajectory.

(b) Distance e.

(c) Angle of view α

Figure 3.5: Study on several values for h with a fixed $k/\gamma = 8$.

Chapter 4

Implementation

Previous chapters described the complete theoretical basis for the docking system comprising a localisation module and a feedback control module which make use of a vision module that provides the localisation system with measurements. This chapter means to describe the whole control and localisation modules, the connections between them, as well as the inner workings of each module regarding practical details and adjustments made to create a real-time implementation for a validation of the system as a whole. Firstly an architecture of the localisation system is presented in Section 4.1, following a brief description of the motors and motor driver board in Section 4.2. Then, a description of the implemented vision algorithm is made in Section 4.3 and finally some solutions for the real system implementation issues are presented in Section 4.4.

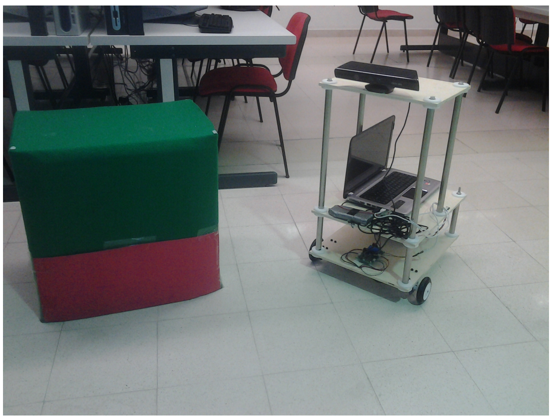


Figure 4.1: Mobile robot prototype and structured visual landmark.

4.1 Localisation Sensor

As stated before, the goal is to localise the landmark in the body-fixed frame with the help of a Microsoft Kinect™ and a pair of incremental optical encoders. In this section, the connection between both filters is clarified as well as their relation to the modules that are responsible of the landmark detection and encoder reading.

In Fig. 4.2 a diagram of the total localisation sensor is illustrated. The MD25 motor driver will periodi-

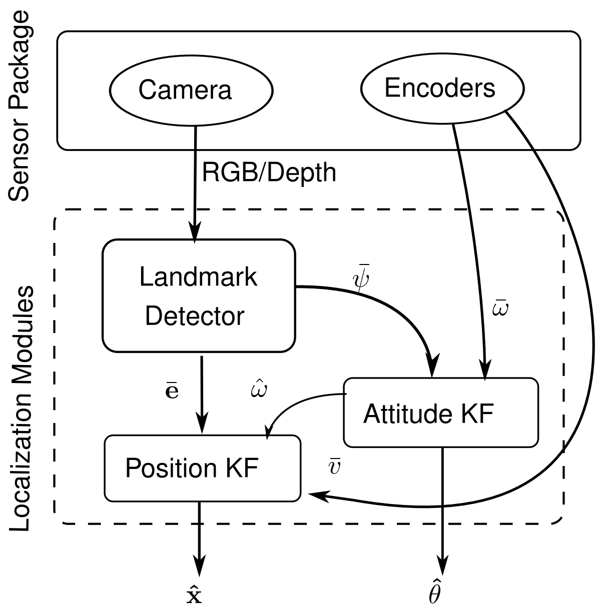


Figure 4.2: Localisation System.

cally collect the encoders readings and feed both Kalman Filters with the measured angular and linear velocities, $\bar{\omega}$ and $\bar{v}$ respectively. These velocities are not necessarily the true ones and so a slow varying slippage of both will be estimated, angular slippage $\hat{s}$ in the Attitude Filter and a linear slippage $\hat{b}$ in the Position Filter. Particularly the angular slippage is used to correct the angular velocity used in the Position Filter, thus parametrizing the state transition matrix with an estimate of the angular velocity.

$$\hat{\omega} = \bar{\omega} + \hat{s} \tag{4.1}$$

The estimate depicted in (4.1) is then used in (2.20) yielding

$$\mathbf{x}_k = \mathbf{\Phi}_k(\hat{\omega}_k)\mathbf{x}_{k-1} + \mathbf{G}_k v_{k-1} + \mathbf{v}_k \tag{4.2}$$

Both the filters will receive information from the landmark detector in order to perform the Kalman Filter update step, in which a correction of the attitude and position will be made, as well as an estimate of both angular and linear velocity slippage will be made. Slippage estimates are only dependant on the innovation processes and the Kalman gains at the time of camera data sampling. It is possible to ignore the estimated angular slippage and simply parametrize the state transition matrix with the raw measured angular velocity.

Both Kalman filters are receiving sensor inputs from 2 different agents, the encoders and the RGB camera, and both these agents are being sampled at an approximately fixed rate but independently from each other, thus making the common case the one where a landmark observation is received at a different time of the encoders measurement one. This calls for a slight change of the update stage of the Kalman filter in order to take into account the fact that the measurements from the camera are not synced with the encoder readings. Assuming the last encoder reading arrived at time t_k and that a landmark observation is received at time $t_k + \delta t$, the update stage would then need a pre-prediction to be done beforehand.

Pre-Update stage (Prediction) This stage basically means to transfer the prediction to the same time as the landmark arrival so that a modularized update stage is possible.

$$\mathbf{P}(t_{k-1} + \delta t|k-1) = \mathbf{\Phi}(\delta t, \hat{\omega}_k)\mathbf{P}(t_{k-1} + \delta t|t_{k-1})\mathbf{\Phi}(\delta t, \hat{\omega}_k) + \mathbf{Q}\delta t \tag{4.3}$$

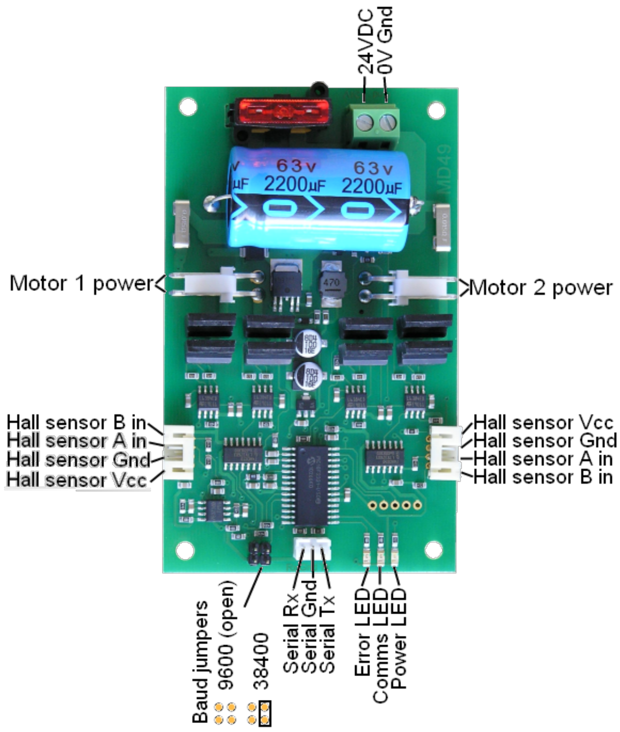
$$\hat{\mathbf{x}}(t_{k-1} + \delta t|t_{k-1}) = \mathbf{\Phi}(\delta t, \hat{\omega}_k)\hat{\mathbf{x}}(t_{k-1}|t_{k-1}) + \mathbf{B}v_{k-1} \tag{4.4}$$

This being done, the update stage can then be performed as normal at time $t_{k-1} + \delta t$, after which a prediction stage would have to be done using the starting point at the latter time.

4.2 Motors and Drivers

The EMG49 includes a $24V$ motor equipped with encoders and a 49:1 gearbox. The optical encoders are coupled in the motor axis and they basically consists of a mechanical light chopper that produces a sine-like wave which can then be transformed into a square wave, thus producing a way of counting the cycles and incrementally computing the wheel displacement.

(a) EMG49A motor.



(b) MD49 serial dual motor driver.

Figure 4.3: Prototype Locomotion Hardware [1].

This particular type of encoder used is a *quadrature encoder* that makes use of two fine grids with a 90^o angular displacement relative to each other which allows to know in which direction the motor is spinning by detecting which of twin square waves will produce a rising edge first. This type of optical encoder also allows to quadruple the angular displacement resolution by taking into account the combination of the 2 possible states of each light detector (high or low). Both the motors are controlled using a MD49 serial dual-motor driver which already has its own velocity controller implemented and adapted to the EMG49 motors, thus allowing for a stable use of the motors simply by sending commands of desired velocity to each one of them. This driver also makes it possible to read the encoder counters values and the current sent to each motor by using a query protocol. Using these values it is possible to find the linear and angular velocities then used to parametrize the LPV Kalman Filter.

$$\Delta s_{l/r}(k) = \frac{2\pi r}{ECST}(enc_{l/r}(k) - enc_{l/r}(k-1)) \tag{4.5}$$

$$\bar{w}_k = \frac{\Delta s_r - \Delta s_l}{L(t_k - t_{k-1})} \tag{4.6}$$

$$\bar{v}_k = \frac{\Delta s_r + \Delta s_l}{2(t_k - t_{k-1})} \tag{4.7}$$

In the last set of equations, $ECST$ is the Encoder Count per Shaft Turn, enc is the encoder counter for either the left or right wheel and Δs is the distance travelled by each wheel. The value for $\bar{v}_k$ and $\bar{w}_k$ can also be obtained by reading the velocity command for each wheel saved in the registers or simply using the one sent in each command. Each method has its disadvantages. Calculating the velocity of the wheel by encoder readings might lead to augmenting any "noisy" reading due to the subtraction used to do so. Using the commands relies on the good behaviour of the communications of the real-time system and also on the good calibration of a linear parameter making the transition between a velocity command defined from 0 to 255 to the actual velocity. In the present work the chosen method is based on encoder counter readings.

4.3 Landmark Identification and Localisation Sensor

For the localisation of the landmark in the environment a Microsoft Kinect™ RGB-D camera is used. It is controlled using the open-source openni drivers [35] built especially to be used under ROS (Robotic Operating System) [36]. In Fig. 4.4 a simple diagram of the process that starts when the raw data is received until a landmark position and orientation is available for the Kalman Filter to process is depicted. The RGB and Depth images are first captured by the openni driver. Afterwards both are

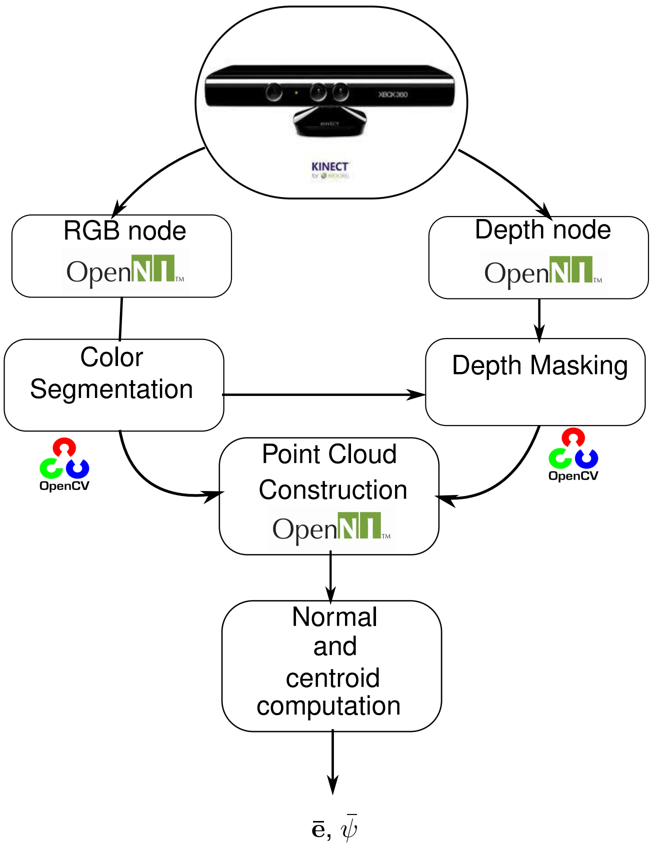
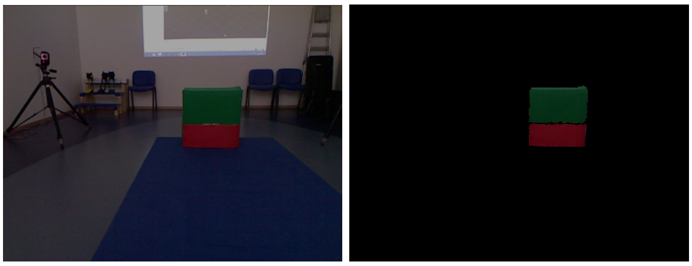


Figure 4.4: Landmark detection scheme.

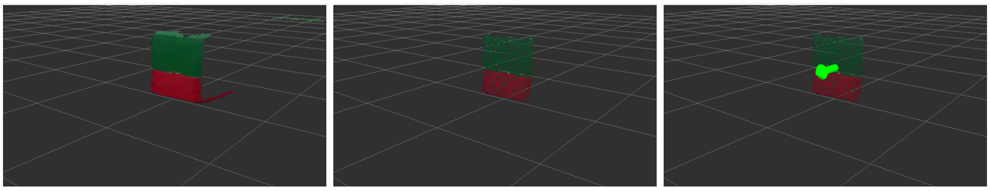
masked by using the RGB image to find the pixels that correspond to the landmark. These are found by defining the contours in the image that involve pixels of desired colours (in this case, bright green and red were used), filling those contours and, with the resulting pixels, building a mask. It is important to refer that many green and red contours are found in an image, so a selection of those needs to be made, since only two represent the real ones. The landmark structure is known and so the segmentation can prepared for the structured relative position between red and green centres, thus making it easy to retrieve the landmark from the environment and avoid labelling random artefacts as being the target.



(a) RGB image. (b) Masked Image.

Figure 4.5: Illustration of segmentation

After the mask is built it is sent to a node where the depth image will be simply masked so that later in the chain only the pixels that represent a landmark are used to build a Point Could, thus saving computational power. After the said Point Cloud is built, it is sent to the "Centroid and Normal Computation" node, where it will first be downsampled by using a voxel grid, which divides the 3 dimensional space in a grid of boxes with a defined size. In this case the choice was to downsample the point cloud with a grid of cubes of $1\ cm$ of side. Inside each cube, only the mean point will remain, allowing for a manageable computation of the centroid and also the normal of the planar landmark. For the centroid computation a simple computation of mean values is used. As for the normal calculation, by applying a RANSAC algorithm (open-source black box [37]), the most probable planar coefficients that build the equation $ax + by + cz = d$ can be found and the normal vector to that plane is known to be $\mathbf{n} = (a\mathbf{i}, b\mathbf{j}, c\mathbf{k})$ by taking the three dimensional gradient of the planar surface.



(a) Initial point cloud. (b) Segmented plane in point cloud with RANSAC. (c) Calculated normal and centroid.

Figure 4.6: Centroid and normal computation process

In the particular case of this work it also known that landmark pose is 2 dimensional and so by taking the transformation between the camera and the centre of both wheels of the mobile robot where the body-fixed frame was defined, it is possible to get a landmark attitude angle.

$$\mathbf{n}^B = {}^B\mathbf{R}_C \mathbf{n}^C \tag{4.8}$$

$$\mathbf{p}_l^B = \mathbf{p}_l^{C\,B}\mathbf{R}_C^T + \mathbf{p}_c^B \tag{4.9}$$

In (4.8) the transformation of the normal vector written in the camera frame $\{C\}$ to the body-fixed frame is depicted, and in (4.9) the transformation of the landmark centroid, where $\mathbf{p}_c^B$ is the camera position in $\{B\}$. The observations the fed to both position and attitude filters are, respectively, $\mathbf{p}_l^B$ and $\bar{\psi} = atan(\mathbf{n}_y^B/\mathbf{n}_x^B)$.

4.4 Implementation Issues

This section addresses situations that occur only due to the limitations of the physical system working in real-time and explores their respective solutions.

4.4.1 Landmark Latency

The theory for the Kalman Filtering assumes a very well defined time-step which, in some real-time systems, is a bold assumption. In the case of this particular implementation, landmark observations suffer from latency that is almost always higher than the encoder mean sampling period.

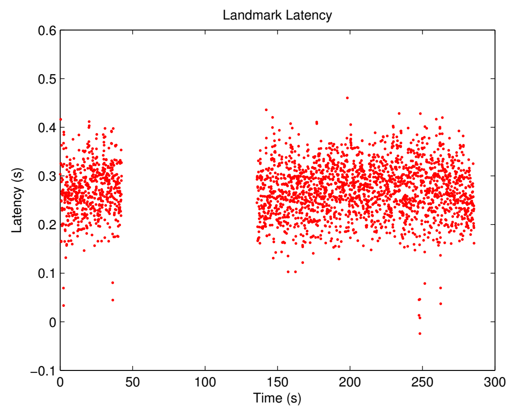


Figure 4.7: Landmark Processing over a real-time experiment.

Because it is not plausible to have a real-time localisation system with an average delay of $0.27\ s$, value taken from the experiment depicted in Fig. 4.7, the prediction steps are done independently of the landmarks. Also, it requires that both the states and covariance matrices and the parametrizing parameters ω and T_k are stored in memory in order to correctly place an update stage in the landmark measuring time, i.e., the time when the RGB and Depth images were collected from the hardware. Taking into account that a normal PC will be used to control the robot, memory is not a bottleneck in this work, so state predictions are saved and each time a landmark measurement is received, both states will return to a the latest stored state time-stamped before the landmark was measured, and perform the necessary time adjustments in order to compute the update correctly. Afterwards, the state is returned to the last encoder saved, thus performing several prediction steps. This strategy avoids then the big latency of landmark measurements and the only disadvantage is that while the landmark is being detected in the images, the robot will be navigating only using prediction parametrized with encoder data which consequently can also lead to slightly greater corrections in the estimate upon landmark observations from the controller perspective, as depicted in Figure 4.8.

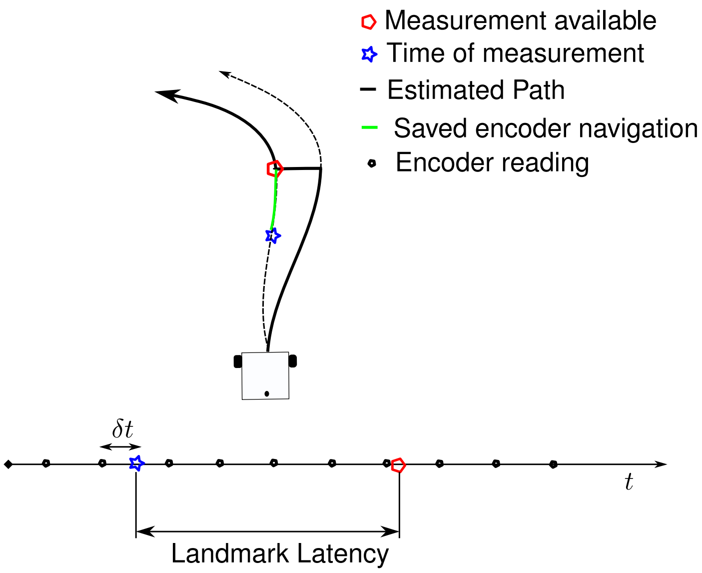


Figure 4.8: Illustration of processing and estimate time delays.

This strategy ensures a small delay from the time the localisation sensor receives data to the time the controller unit is able to broadcast commands to the motor driver, as depicted in Fig. 4.9, which depicts the values of this delay in a real-time docking experiment.

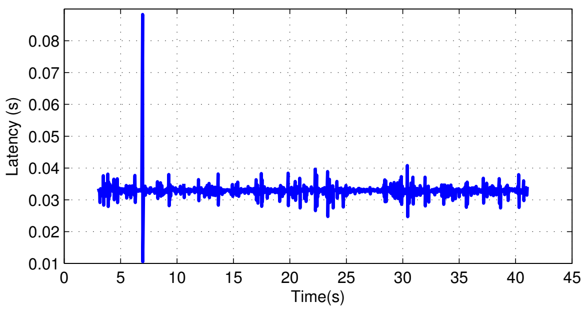


Figure 4.9: Time delays since measurement to command broadcast.

4.4.2 Saturation and Curvature of Trajectories

As stated before, wheel velocities can only take values ranging from $c = 0$ to $c = 255$, where $c = 128$ means that the wheel is not moving. This then gives a limited range of possible values for the angular and linear velocities. To better make use of the parameters used to tune the state feedback law presented in Section 3.2, it is beneficial to treat the saturations in such a way that the curvature of the trajectory remains constant. Moreover, in order to allow for the localisation system to converge properly and to avoid slippages, it is also beneficial to keep the linear and angular velocities saturated at a certain value. The aforementioned saturations are illustrated in Fig. 4.10. The saturation strategy then comes down

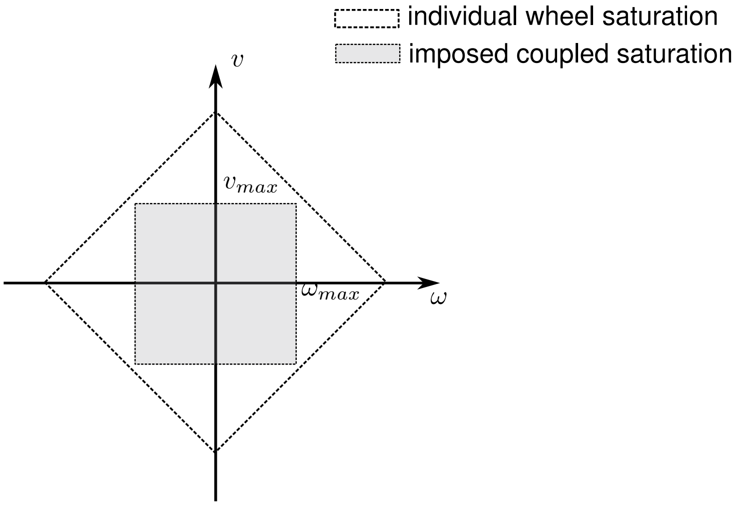


Figure 4.10: Wheel saturation.

to choosing appropriate values for ω_{max} and v_{max} and assuring that whenever either linear or angular velocities saturate, the new commands follow

$$\frac{\omega_{sat}}{v_{sat}} = \frac{\omega}{v} = R = cte. \tag{4.10}$$

Chapter 5

Simulation

This chapter focuses in the simulation of both localiser and controller units. Testing of the localiser will focus on the analysis of its errors in the presence of noisy landmark observations, under periods of landmark unavailability. These situations will be tested while the virtual robot carries out a trajectory defined in the inertial frame. Some simulations were set with imposed slippages so as to assess the localisation system behaviour under such conditions with and without landmark visualisations. The controller unit will be tested for a set of parameter values and also with a suited wheel saturation strategy imposed for a better functioning of the localiser.

5.1 Trajectory generation

The trajectory followed by the virtual body system is generated through built v and ω signals which are then integrated using the expressions for a differential drive mobile robot 5.1, yielding a path expressed in the inertial frame

$$x(t) = x(0) + \int_{t_0}^{t} v(\tau)\cos(\theta(\tau))d\tau \tag{5.1a}$$

$$y(t) = y(0) + \int_{t_0}^{t} v(\tau)\sin(\theta(\tau))d\tau \tag{5.1b}$$

$$\theta(t) = \theta(0) + \int_{t_0}^{t} \omega(\tau)d\tau \tag{5.1c}$$

where x and y refer to the robot's position in $\{I\}$ and θ refers to the attitude of it.

5.2 Landmark Observation Simulation

As stated before, for this work a RGB-D camera will be used and so the measurements at disposal will be the position and attitude of the landmark in the sensor-based frame. In order to simulate measurements we have then to compute the relative position of the landmark and the mobile robot. The landmark position and orientation in the inertial frame are known beforehand and will be denoted as $[x_l\ y_l\ \theta_l]$. As for the mobile robot position in the inertial frame, it is denoted as $[x\ y\ \theta]$. The simulated measurements will the follow (5.2)

$$\bar{e}_x = r\cos(\alpha - \theta) \tag{5.2a}$$

$$\bar{e}_y = r\sin(\alpha - \theta) \tag{5.2b}$$

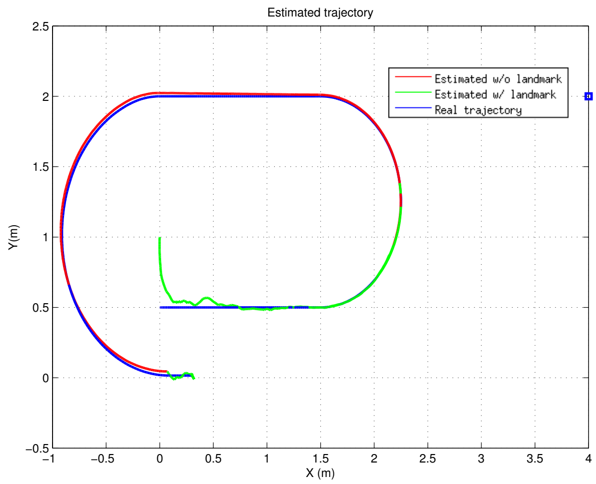
$$\bar{\psi} = \theta_l - \theta \tag{5.2c}$$

where $r = \sqrt{(x - x_l)^2 + (y - y_l)^2}$, $\alpha = atan2(y_l - y, x_l - x)$ and $e_{x,y}$ represent the x, y component of $\mathbf{e}(t)$. The simulated measurements also take into account the Kinect camera range and angle of view, which are taken to be approximately $5\,m$ and 52^o respectively.

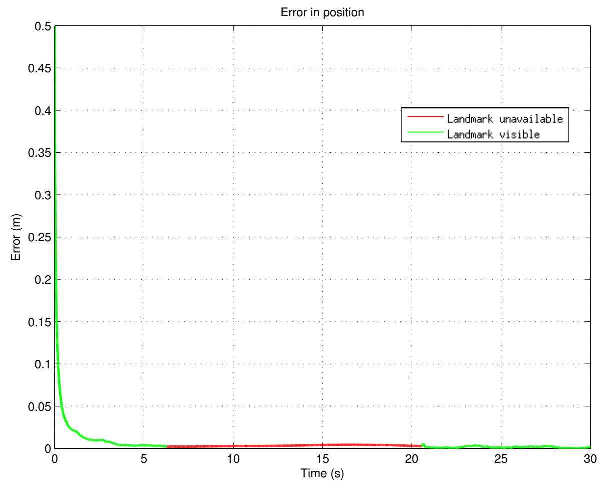
5.3 Simulation Results

5.3.1 Continuous Model

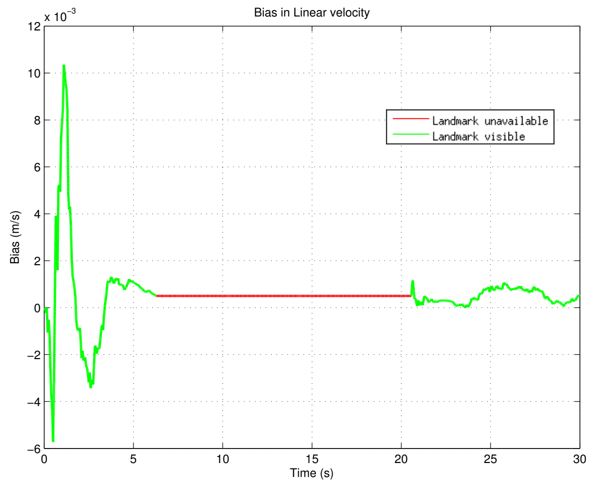
The filter dynamics are assumed to be affected by a standard deviation of $1\,cm/s$ and $0.1\,cm/s$ both in x and y axis as well as $0.1\,cm/s^2$ in the velocity bias. Also, the camera sensor and detection algorithm will be assumed to have a standard deviation of $1cm$ in both axis when localising the landmark. These values are used to build both $\mathbf{Q}$ and $\mathbf{R}$ diagonal matrices, thus making them $\mathbf{Q} = diag(1e-4, 1e-6, 1e-6)$ and $\mathbf{R} = diag(1e-4, 1e-4)$. Firstly a simulation was conducted using the continuous time equations and imposing no slippages in the velocity readings, resulting in the estimated trajectory shown in 5.1.



(a) Real and estimated trajectory.

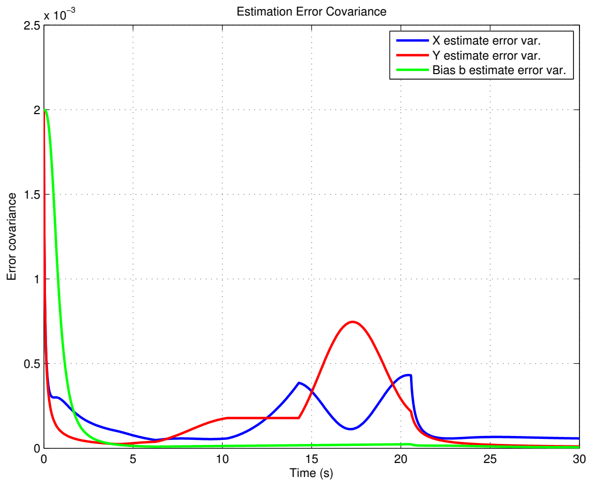


(b) Position error norm.

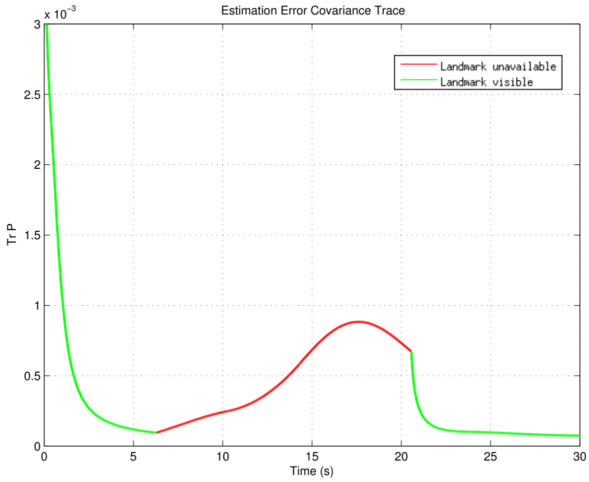


(c) Estimated linear slippage.

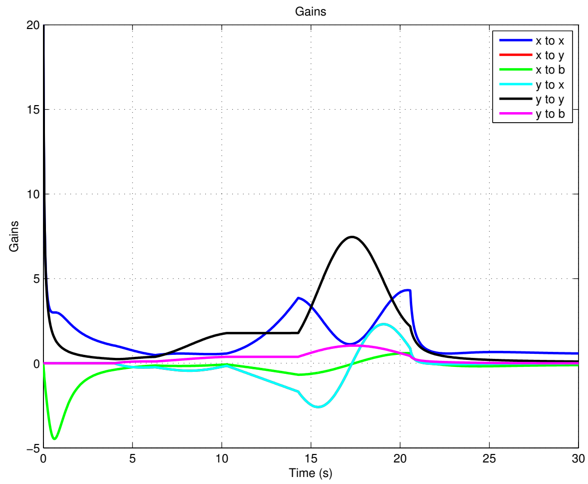
Figure 5.1: Results for simulation with no imposed slippage.



(a) Estimate error covariance.



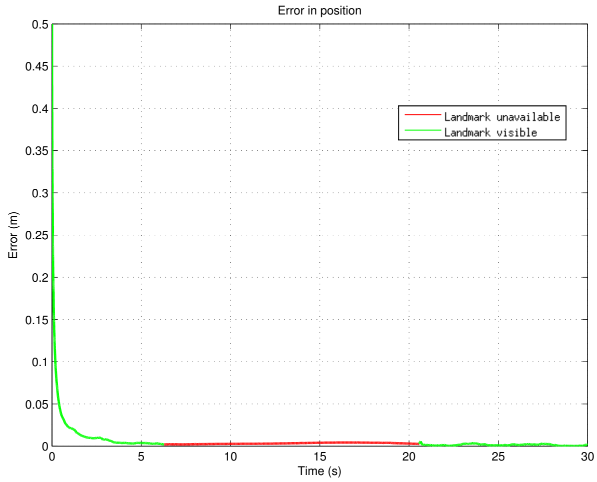
(b) Trace of covariance matrix.



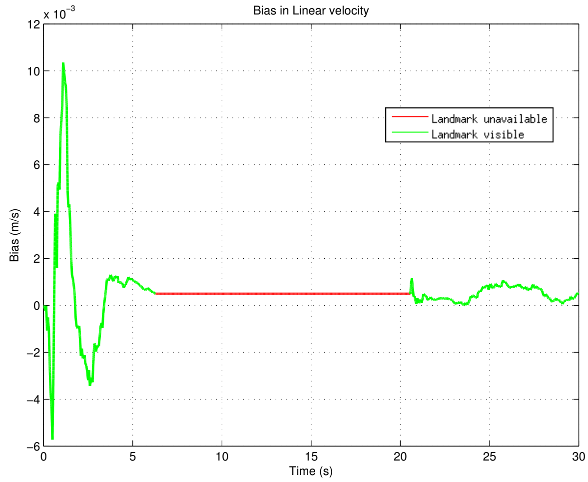
(c) Kalman matrix gains.

Figure 5.2: Error Covariance and Kalman gains illustration

The above performed simulation was meant to simply validate the model used and confirm that the filter has a correct behaviour. Note that there is a transfer of error covariance from one axis to the other as the vehicle turns. The simulation in Figure 5.3 will already test the behaviour of the slippage estimate in order to check if it is being well estimated, so assuming that the linear velocity readings were affected by a bias $b = 0.05\ m/s$, the results are the ones shown below.



(a) Position estimate error norm.



(b) Linear slippage estimate.

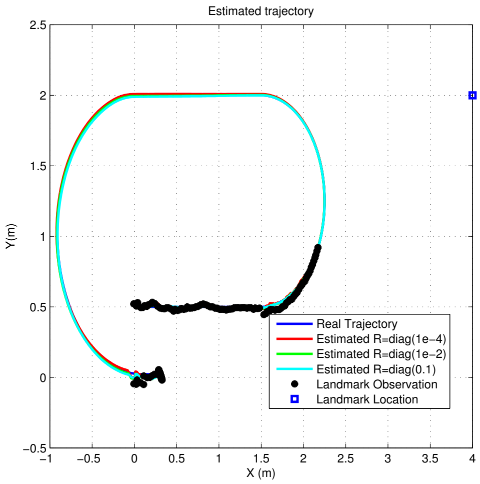
Figure 5.3: Depiction of slippage estimate with imposed $b = 0.05$.

The difference is negligible especially in the x and y axis localisation. The reason is that the constant linear slippage is being estimated in the model, as can be seen in Fig. 5.3. The observability of b is thus validated.

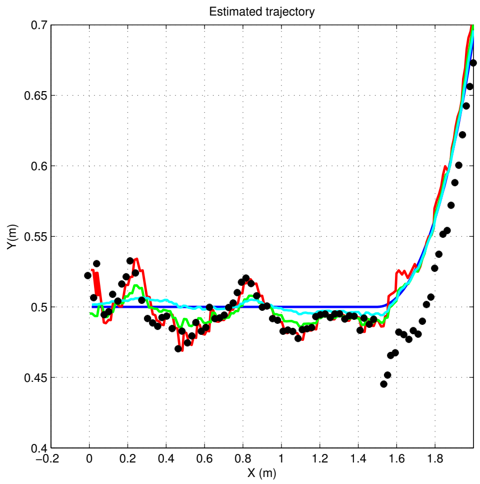
5.3.2 Discrete Model

This section shows the results for the discrete filtering system, also testing its behaviour under landmark unavailability and imposed slippage values. Figures 5.4 and 5.5 show simulations where three different values of sensor noise covariance matrix $\mathbf{R}_k$ are experimented in order to assess which value would hold the most smooth estimate.

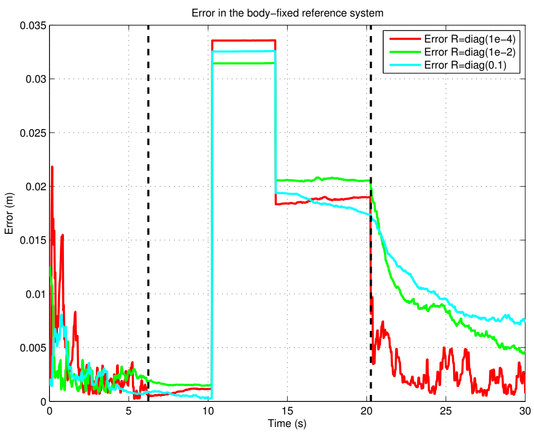
In both situations, $\mathbf{R} = diag(1 \times 10^{-2})$ seems to be a good compromise between a fast convergence and a good noise rejection filtering. Although the transient values for the estimation error are noticeable in Fig. 5.5 because of the initial wrong estimate $\hat{b}(0)$, it still converges to zero in an appropriate time frame.



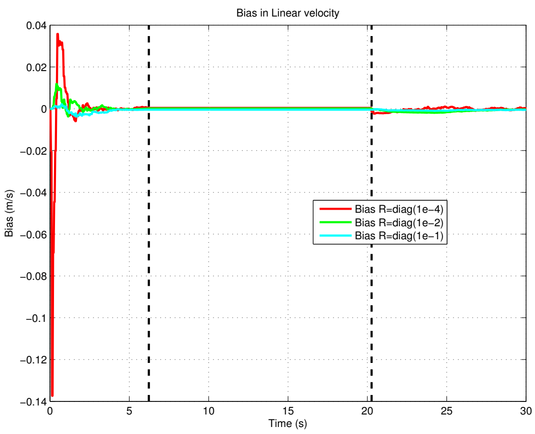
(a) Real and estimated trajectory.



(b) Zoom of trajectory.

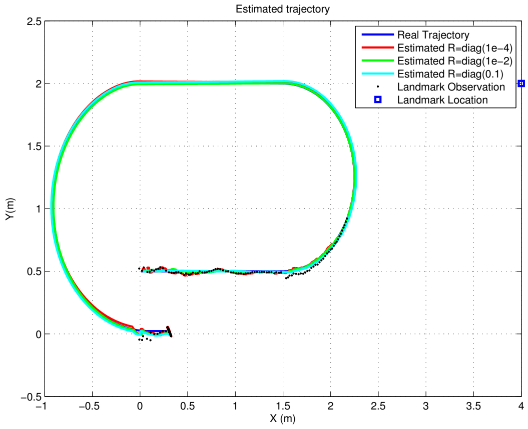


(c) Position estimate error norm.

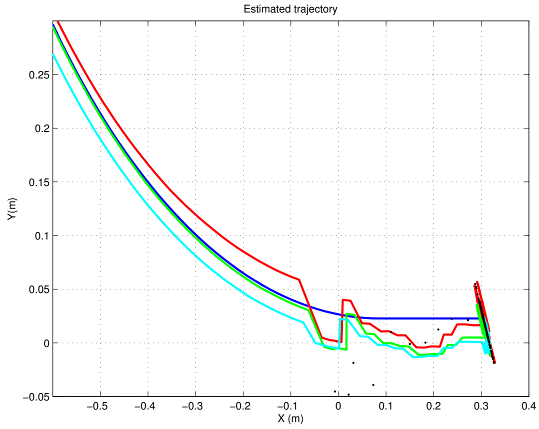


(d) Linear slippage estimate.

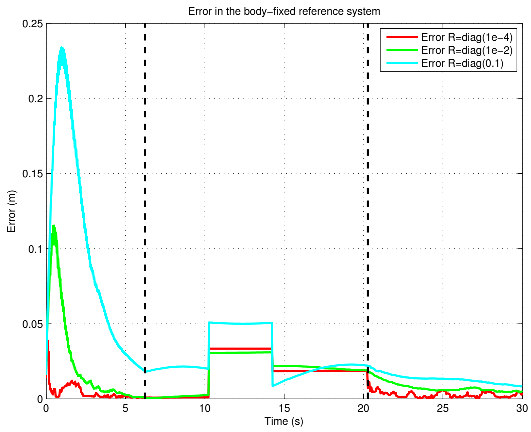
Figure 5.4: Simulation with no imposed slippage.



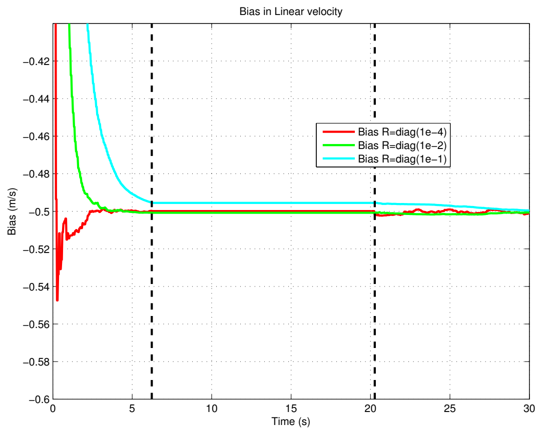
(a) Real and estimated trajectory.



(b) Zoom of trajectory.



(c) Position estimate error norm.



(d) Linear slippage estimate.

Figure 5.5: Simulation with imposed slippage of $b = 0.5$.

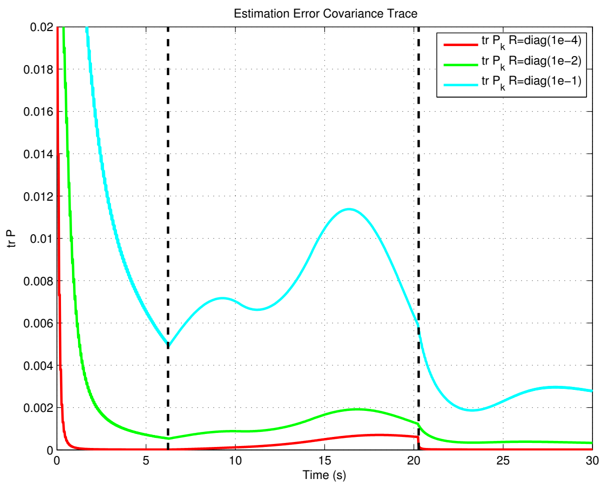
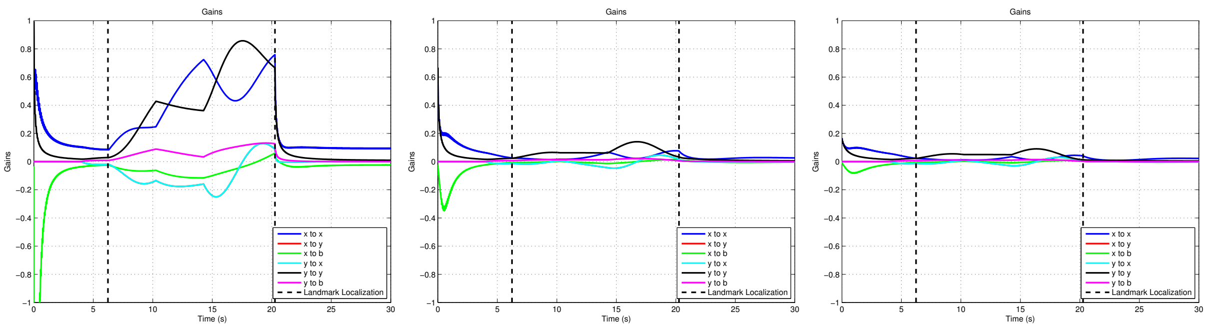


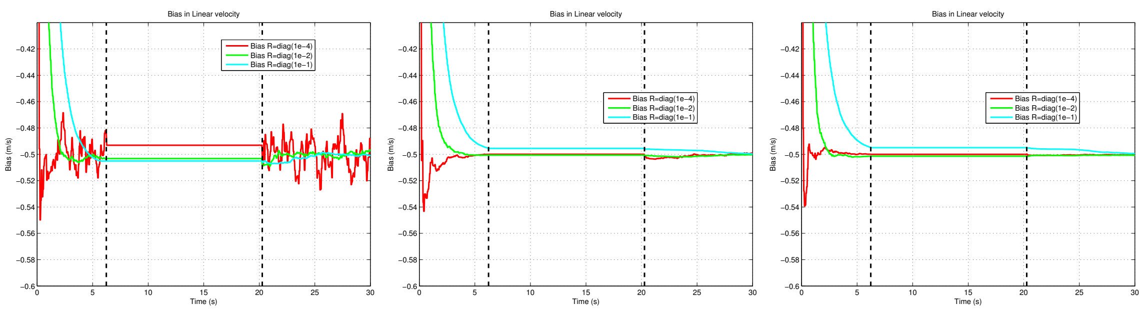
Figure 5.6: Error Covariance Trace.



(a) $\mathbf{R} = diag(1 \times 10^{-4})$ (b) $\mathbf{R} = diag(1 \times 10^{-2})$ (c) $\mathbf{R} = diag(1 \times 10^{-1})$

Figure 5.7: Kalman gains illustration.

Before fixing any values for $\mathbf{R}$ though, it is worth testing some values for the confidence that we have in the linear bias model $\dot{b} = 0$, because it needs to respond to a wrong initial value an stabilize at the correct level, but it can't suffer from bad landmark observations. Fig. 5.8 illustrates the behaviour of the bias estimate with three different q_{33} values (denoting the covariance of the plant noise regarding the slippage model). A noisy estimate should be avoided and yet one that takes too long the initial transient to go to the correct value is also not desirable so $\mathbf{R}$ and $\mathbf{Q}$ must be chosen in a way to reach a compromise between a low convergence time and good rejection of outlier measurements. The chosen values were $\mathbf{Q} = diag(1 \times 10^{-5}, 1 \times 10^{-5}, 1 \times 10^{-9})$ and $\mathbf{R} = diag(1 \times 10^{-4}, 1 \times 10^{-4})$, although these choices may be refined during experimental analysis.

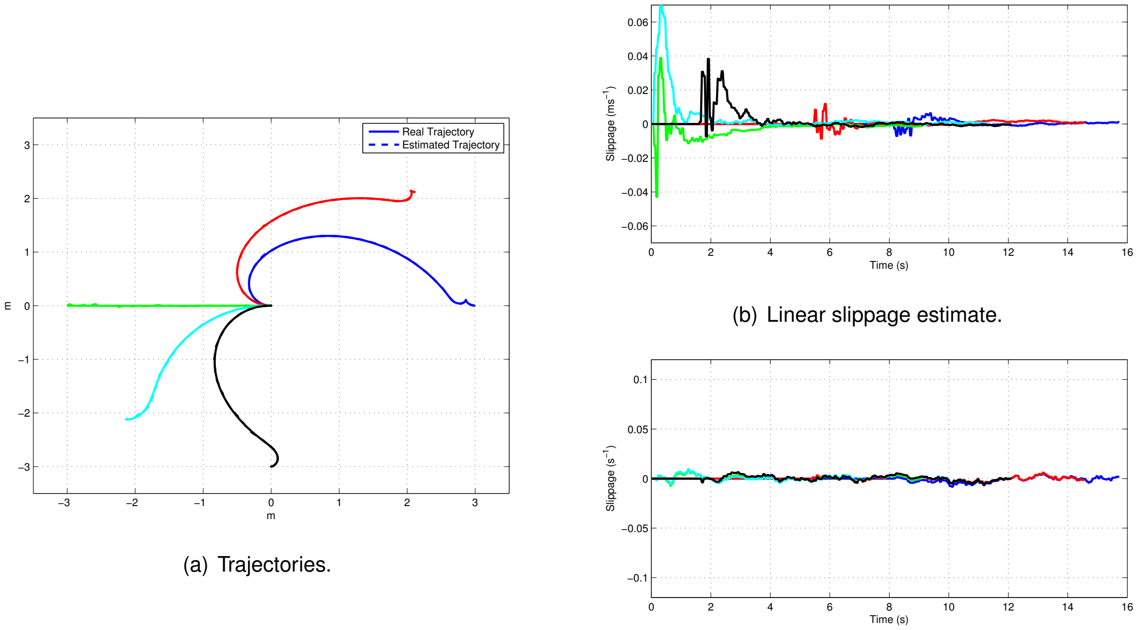


(a) $q_{33} = 1 \times 10^{-3}$ (b) $q_{33} = 1 \times 10^{-6}$ (c) $q_{33} = 1 \times 10^{-9}$

Figure 5.8: Influence of the plant noise covariance in the bias estimate.

5.4 Simultaneous Localisation and Control

This section will focus in presenting results of simulations where the automatic control is activated. Several initial positions are taken and some imposed slippages are tested in order to assess the situations in which the localiser can or cannot give the right information to the controller. Note that all filtering parameters are maintained from the previous sections. For all the following simulations, the localisation system started with the correct initial position given that the localisation is only globally asymptotically stable when the presence of observations is guaranteed, i.e., the system converges to the origin in the presence of a wrong initialisation only if the landmark is in the range of camera operability at some point during the manoeuvre. Fig. 5.9 shows a simple simulation where no slippages are induced, and so the estimate is nearly overlapped with the real trajectory, allowing for a correct parking of the vehicle.



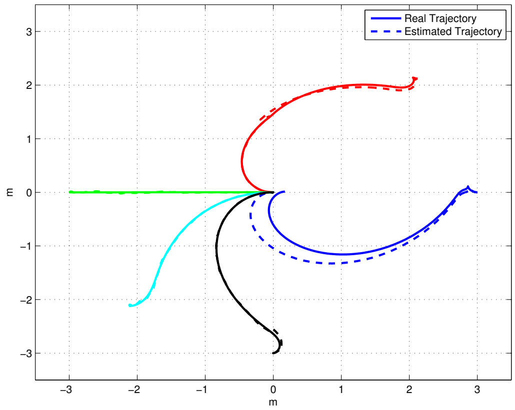
(a) Trajectories.

(b) Linear slippage estimate.

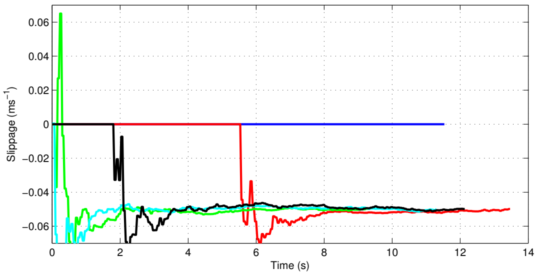
(c) Angular slippage estimate.

Figure 5.9: Parking with no imposed slippages.

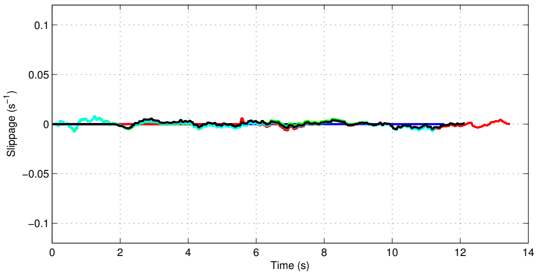
In the set of figures from 5.10 to 5.12 different combinations of values for the angular and linear slippages were imposed in the model, in order to assert under which conditions the controller would be able to drive the real and estimated pose to the origin. In situations where the controller never drives the vehicle into a position under which he receives a landmark observation, then the correct slippages will never be estimated and therefore the position and attitude estimates will the driven to the origin but the real vehicle will not. This gives emphasis to the argument that the work here presented can be envisioned as a module that is activated once a given target is recognised by an algorithm such as a task planner.



(a) Trajectories.

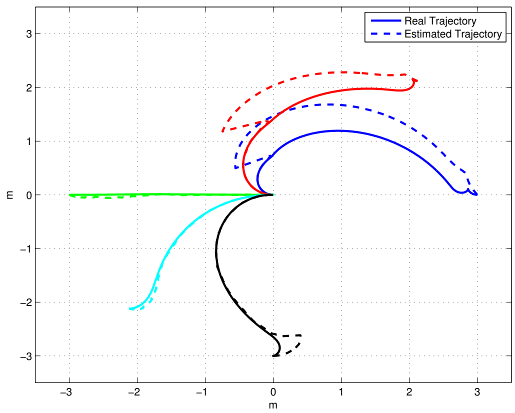


(b) Linear slippage estimate.

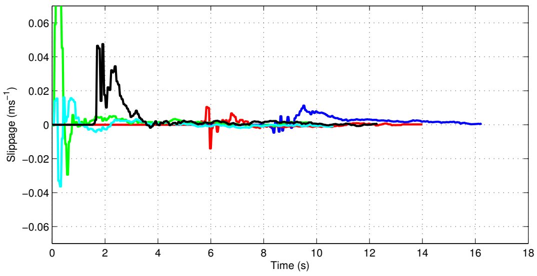


(c) Angular slippage estimate.

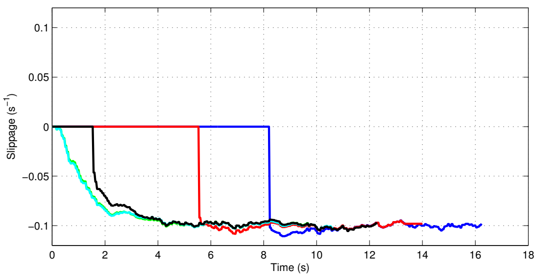
Figure 5.10: Parking with imposed $b = -0.05$.



(a) Trajectories.

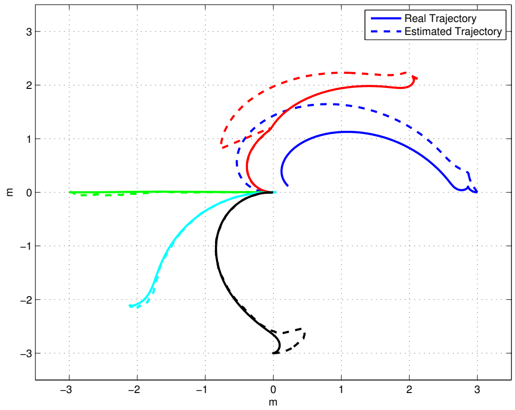


(b) Linear slippage estimate.

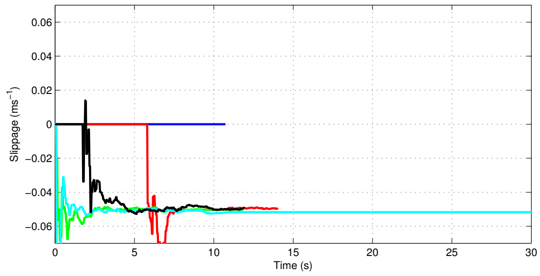


(c) Angular slippage estimate.

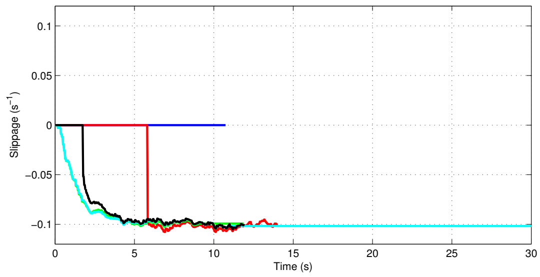
Figure 5.11: Parking with imposed $s = -0.1$.



(a) Trajectories.



(b) Linear slippage estimate.



(c) Angular slippage estimate.

Figure 5.12: Parking with imposed $b = -0.05$ and $s = -0.1$.

The performed simulations validate the idea that the vehicle can still complete the parking task even if the estimate deviates from the real position during a manoeuvre, by correctly estimating any slippage occurring in the wheels. Figure 5.12 shows one extreme case where the real vehicle did not park in the correct place seeing as the landmark was never visualised, not allowing for any slippage to be estimated, leading to an accumulation of error that led the control laws to drive the vehicle to a position where the landmark was not in sight.

Chapter 6

Experimental Results

Every scientific effort that has some practical purpose and which purpose goes towards an implementation in a real world scenario must go through some sort of experimental validation. This is particularly true when it comes to industrial applications, even if the simulation results are satisfactory under realistic assumptions, and the reason is because there are always unpredictable factors such as the effect of some flaw in the raw data processing or some unforeseen environmental property that distorts the sensor's proper operation. In this chapter, firstly the experimental setup is explained in Section 6.1. Secondly, the experimental results performed to validate the localisation system with real-time data are presented in Section 6.2, and finally, in Section 6.3, some real-time tests are carried out with the closed-loop feedback law activated, making it possible to verify the functioning of the docking system as a whole.

6.1 Experimental Setup

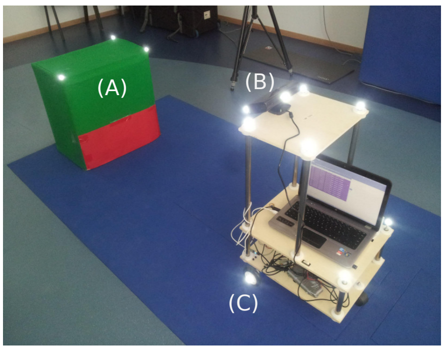


Figure 6.1: Biomechanics Laboratory of Lisbon (Landmark (A), Robot prototype (B), Kinect Camera (C)).

The ground truth validation data acquisition system used consists of a Qualisys™ Motion Tracking [38] system that uses 14 different cameras to track the position of reflectors placed upon the mobile robot. The characteristics of the tracking system are listed in Table 6.1.

Cameras	14 Qualisys Pro Reflex 1000
Frequency	100 Hz
Markers	19 mm diam. passive retroreflectors
Precision	$< 1mm$ after calibration

Table 6.1: Qualysis Motion Tracking system characteristics.

The robot prototype and landmark setup are shown in Fig. 6.1. Several passive retroreflectors, which are highlighted by the camera flash, were placed on the robot and landmark to provide redundant ground truth data. Below is a summary of the parameters and initialization of both Kalman Filters.

- Camera noise covariance: $\mathbf{R}^{\mathbf{x}} = 1 \times 10^{-2}\mathbf{I}_2$ and $\mathbf{R}^{\theta} = 1 \times 10^{-2}$
- Plant noise covariance: $\mathbf{Q}^{\mathbf{x}} = diag(4.1 \times 10^{-6}\mathbf{I}_2, 1 \times 10^{-8})$ and $\mathbf{Q}^{\theta} = diag(2 \times 10^{-5}, 1 \times 10^{-8})$
- Initial covariance matrix: $\mathbf{P}_0^{\mathbf{x}} = 1\mathbf{I}_3$ and $\mathbf{P}_0^{\theta} = 0.1\mathbf{I}_3$
- Initial conditions: ê and $\hat{\theta}$ were set to the real initial position, and both bias estimates $\hat{b}$ and $\hat{s}$ were set to zero.

It is also important to bear in mind the position of the camera frame $\{C\}$ relative to the body-fixed frame $\{B\}$, defined by a translation and a rotation particularized below

$$\mathbf{p}_C^B = [0.090 \quad 0.03 \quad 0.775]^T\,(m),$$
$${}^{B}\mathbf{R}_C = \begin{bmatrix} c(\theta) & -s(\theta) & 0 \\ s(\theta) & c(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\theta = 0.216\,(rad)$.

6.2 Experimental Validation of Localisation System

This section comprises the localisation results for two separate trajectories performed in a laboratory equipped with a Qualisys motion tracking system. For an easier visualisation, only a portion of the first trajectory tested is depicted in Fig. 6.2, where *Estimate* refers to a correct initialisation and *Estimate 2* to a wrong one, for global stability validation. The comparison between ground truth data expressed in $\{B\}$ and the estimate is depicted in Figures 6.3 to 6.5. In these results neither of the slippages were being estimated, so the open-loop is carried-out with odometry data alone. The landmark is not always visible from the robot. The maximum difference between the estimated trajectory occurs after an unavailability of landmarks during a period of 30 seconds and also due to sensor faulty measurements at the end of the experiment. A statistical representation of these differences can be seen in Fig. 6.6.

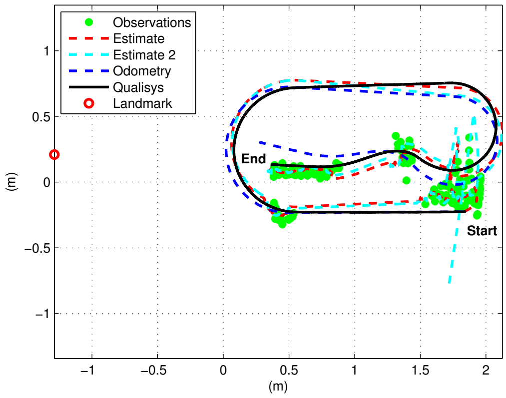
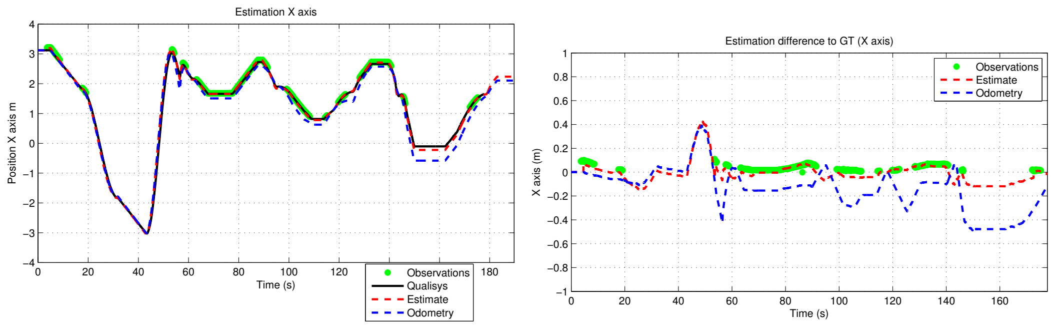


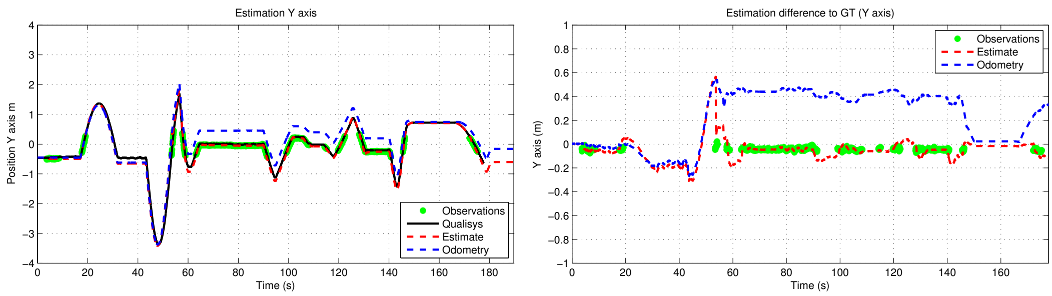
Figure 6.2: Ground truth and Estimate (no slippage estimation).



(a) Trajectory.

(b) Estimation error, relative to ground truth information.

Figure 6.3: Estimate and Ground Truth in X axis (no slippage estimation).



(a) Trajectory.

(b) Estimation error, relative to ground truth information.

Figure 6.4: Estimate and Ground Truth in Y axis (no slippage estimation).

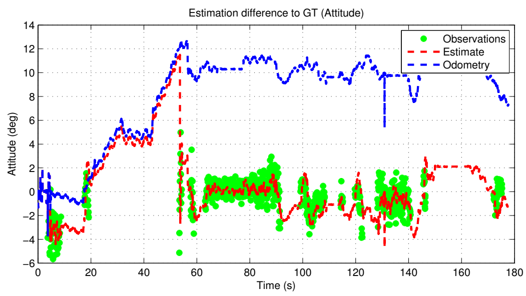


Figure 6.5: Attitude estimate error, relative to ground truth information (no slippage estimation).

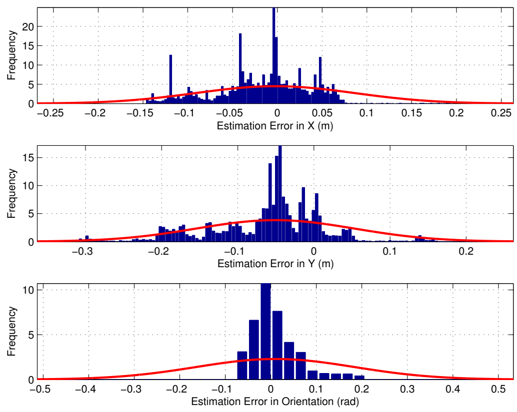
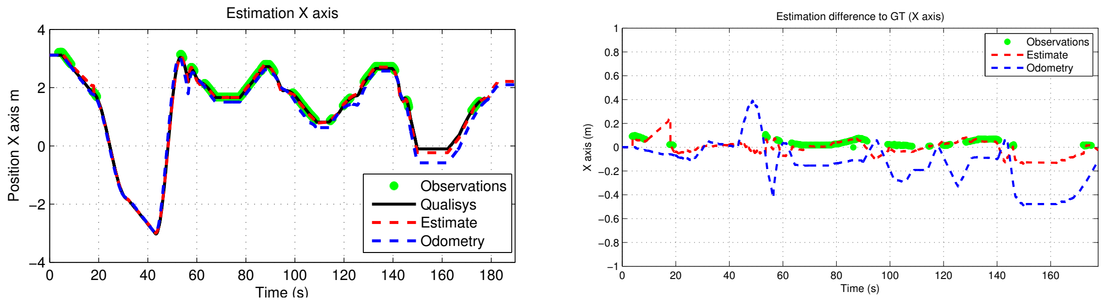


Figure 6.6: Statistical study of estimation error relative to ground truth.

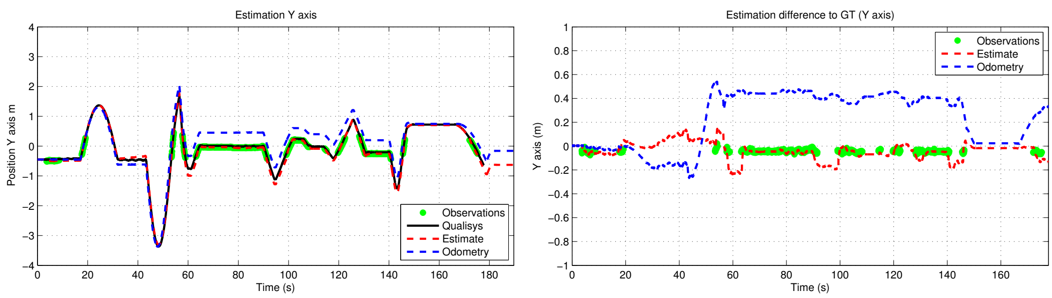
When the same trajectory data is processed while also estimating the slippages, some improvement can be noticed, especially in the portion of the trajectory where both angular and linear velocities are maintained. The mentioned data set goes from 20 seconds to 50 seconds from the beginning of the experiment and the slippage estimation effect can be seen when comparing between Figures 6.3 to 6.5 with Figures 6.7 to 6.9., being the latter the ones representing the experiment where the slippages were estimated.



(a) Trajectory.

(b) Estimate error, relative to ground truth information.

Figure 6.7: Estimate and ground truth in X axis.



(a) Trajectory.

(b) Estimate error, relative to ground truth information.

Figure 6.8: Estimate and ground truth in Y axis.

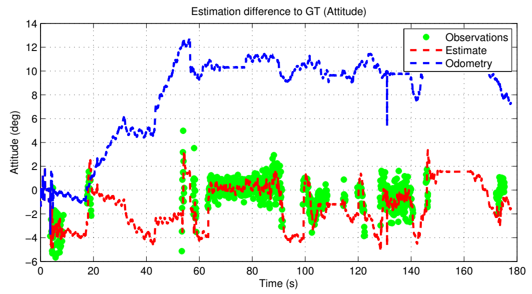
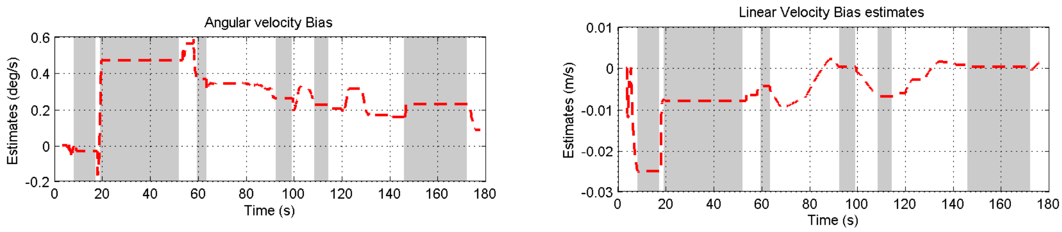


Figure 6.9: Attitude Estimate error, relative to ground truth information.

Since the robot kept both velocities nearly constant, the slippage estimation that took place until the 20 second mark was suitable until the 50 second mark, allowing for a reduced open-loop estimation error. The slippage estimations are depicted in Fig. 6.10, where the shaded areas correspond to the time periods of landmark unavailability.



(a) Angular slippage.

(b) Linear slippage.

Figure 6.10: Slippage estimates

With the same data, a different test was conducted, this time forcing $b = -0.01\ (m/s)$. The slippage estimate behaviour is depicted in Fig. 6.11.

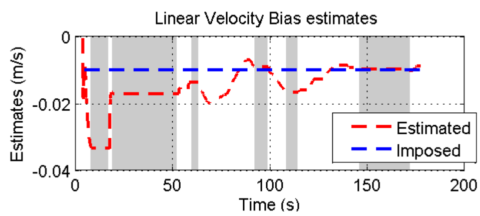


Figure 6.11: Linear slippage estimation.

A more intuitive demonstration of the slippage estimation effect in the path estimation is depicted in Fig. 6.12 which shows the data relative to a second simpler trajectory, where the robot maintained its linear velocity throughout the whole experiment, *Estimate* refers to an experiment without slippage estimation and *Estimate 2* is influenced by the slippage estimate.

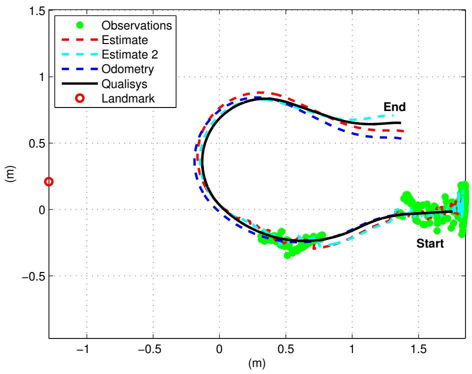
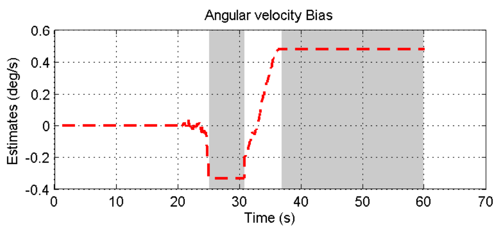
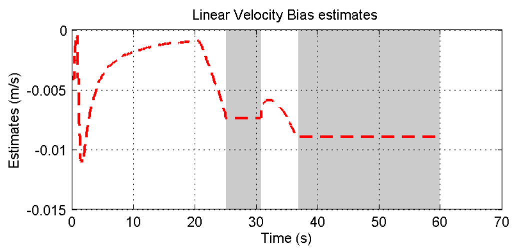


Figure 6.12: Trajectory uninfluenced by slippage estimation.

The slippage estimation both in the linear and angular velocity respective to the trajectory in Fig. 6.12 are depicted in Fig. 6.13.



(a) Angular slippage.

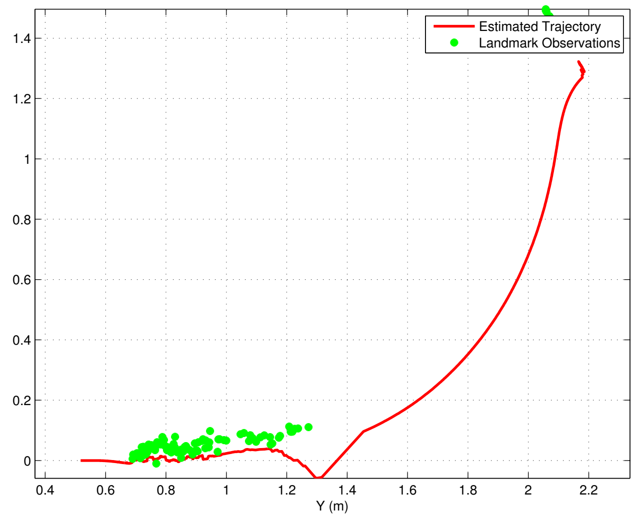


(b) Linear slippage

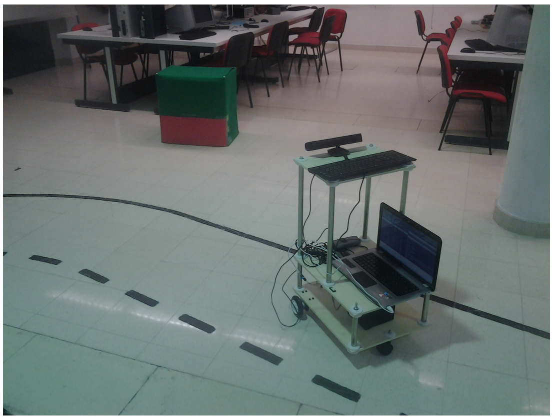
Figure 6.13: Slippage estimate.

6.3 Docking System Validation

In this section some tests regarding the full docking system are presented. The localisation system uses the same parameters as in previous sections, the landmark is considered to be the origin of the inertial frame and the goal of every experiment presented in this section was set $0.5\,m$ in front of the real landmark object being used, effectively representing the origin of the closed-loop system. Also, in every experiment, unless stated otherwise, the initial estimate of the position is set to be near the real position of the mobile robot, seeing as the estimation filters were active before the feedback loop was enabled. In the first experiment saturations of $\omega_{max} = 0.2$ and $v_{max} = 0.2$ were applied.



(a) Estimated trajectory.

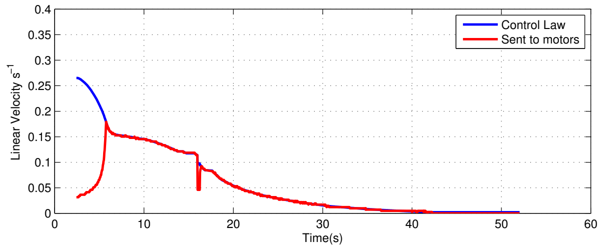


(b) Initial position.

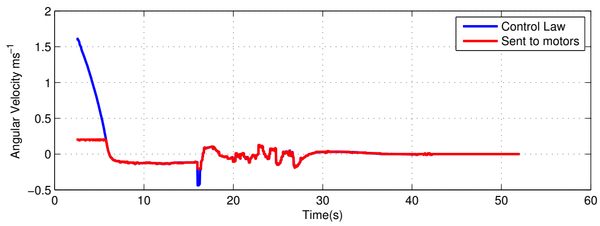
(c) Final position.

Figure 6.14: Docking with no saturation ($h = 20$, $k = 1$, $\gamma = 0.125$).

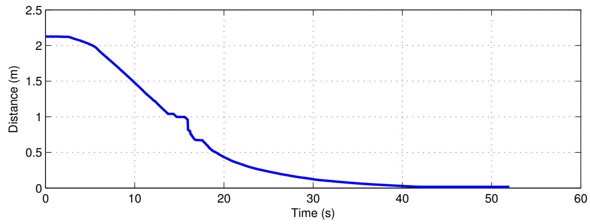
It is possible to notice that the estimate shows no curvature at the end of the docking process, which goes towards conclusions drawn in simulations previously performed. Also, in Fig. 6.15 one can observe the effect of the correction of the estimate in the command around second 15.



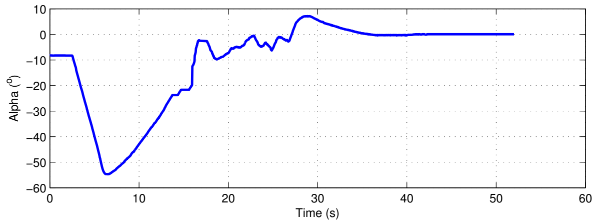
(a) Linear velocity command



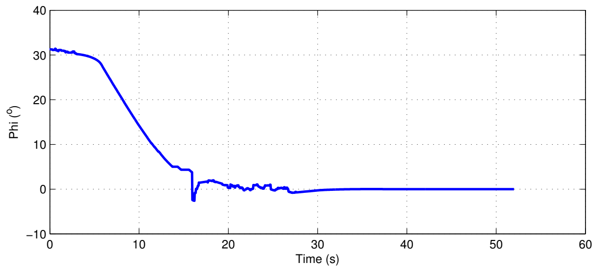
(b) Angular velocity command



(c) e.



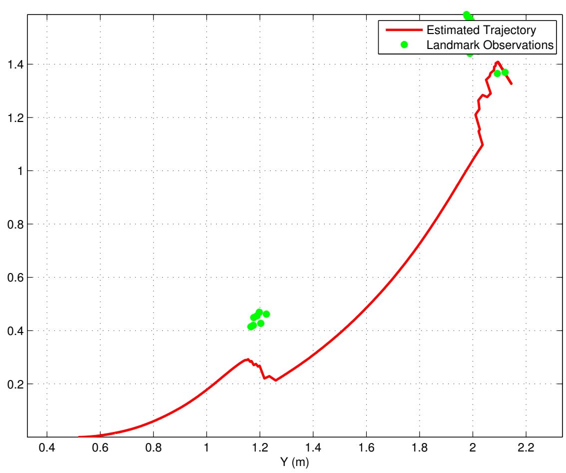
(d) α.



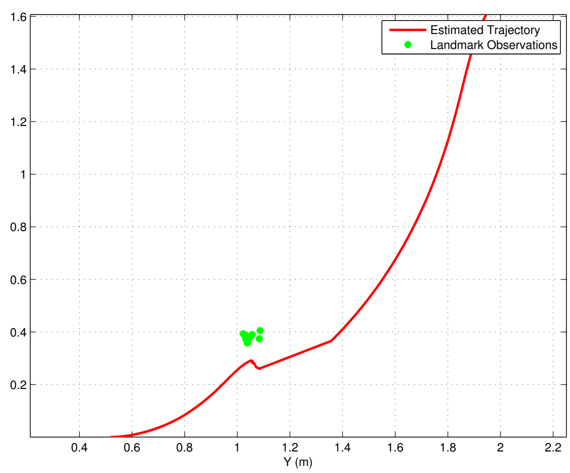
(e) ϕ.

Figure 6.15: Progression of state vector and commands values for a docking manoeuvre.

In the next set of tests one can observe the effect of changing h to 7.5, which is near the limit of the curvature constraint in (3.19), as well as observe the effect of saturation in v and ω.



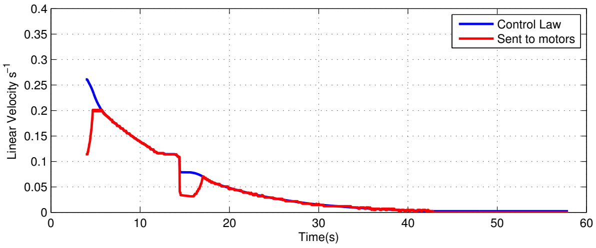
(a) With unsaturated commands.



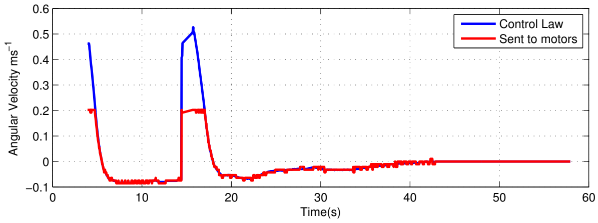
(b) With saturated commands.

Figure 6.16: Docking with ($h = 7.5$, $k = 1$, $\gamma = 0.125$.)

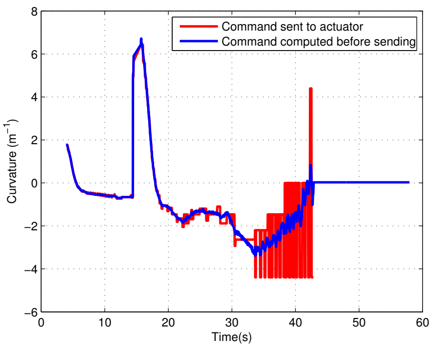
Performing a lower curvature will cause the landmark availability to decrease in similar situation to the one in Fig. 6.16, illustrating that higher values of h are prone to perform better, but also, on a positive note, that the saturation used has little effect on the resultant trajectory. Moreover, it is very clear here the effect of the latency in the landmark measurement and the solution proposed to smooth its effect described in Section 4.1. Fig. 6.17 shows the effect of saturation on the commands, where it is observable how the linear velocity behaves when the angular velocity is saturated, in a way to maintain the curvature.



(a) Linear velocity.



(b) Angular velocity.



(c) Curvature before and after quantization and saturation.

Figure 6.17: Saturation effect on commands.

Notice that after the velocity hits low values because the robot is approaching the goal, the angular velocity suffers some noise due to the quantization effect. Finally, Fig. 6.18 shows several successful docking trajectories with different starting points, including some where the robot was not facing the landmark at the beginning, and even one where the initial estimate was rather wrong. The landmark was placed at the origin of the reference frame in order to visualise these results.

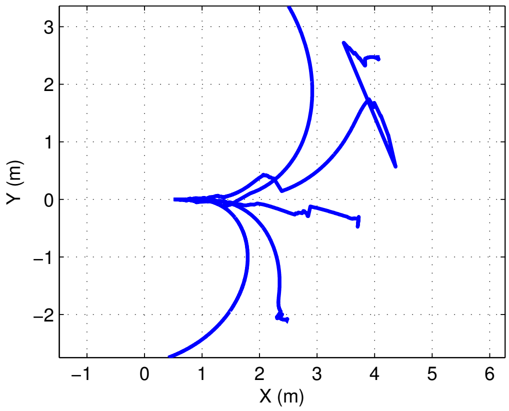


Figure 6.18: Several successful trajectories.

Chapter 7

Conclusions

A sensor-based positioning system based on measurements from optical encoders and from feature recognition using an RGB-D camera is proposed and experimentally validated. The proposed estimation system is able to localise a certain feature in the environment, tracking it even if it is not in sight, by estimating any slippage that might be occurring in the wheels for a better open-loop navigation. The Kalman filtering solution makes use of a new linear differential drive mobile robot kinematics by representing the movement of the environment in the robot frame instead of the inverse, allowing for a sub-optimal linear estimation and noise reduction due to the fact that no rotation is executed. The estimate is seen to converge rapidly once a landmark is in sight and also to be globally stable in faulty initializations or kidnapping scenarios. The slippage estimation contributes positively for the localisation in open-loop if the robot does not change its speed too drastically, seing as the slippage takes some time to be estimated due to the choice of values for $\mathbf{Q}$ and $\mathbf{R}$, which were chosen so as to smooth the estimate and also avoid the slippage estimate to respond to noise or faulty measurements. The devised feedback control successfully drives the robot to a given goal and is tunable to the needs of the localisation system, seeing as it is possible to require the robot to take the same path under different speeds by just adjusting the parameters that tune the feedback. It is also possible to achieve good results by imposing a saturation in linear and angular velocities as to ensure convergence of the localisation system within the manoeuvre time frame. The robot is able to drive itself to the correct goal even when a wrong initial position and attitude estimate occurs, given that the landmark is visible. Also the localisation system is able to reject some of the corrupt camera measurements allowing for a smoother driving.

7.1 Future Work

Notwithstanding the fact that the present thesis carried out its purpose, there is still, as always, room for improvement. Regarding the theoretical work, the localisation was devised to track a static landmark, giving the opportunity for future work to address the problem of localising a moving robot by also using a on-board vision based system. Also, now regarding the control law, the devised feedback law is easily extendable to the case where the goal is moving through the inertial frame by following in works in

[34]. It is also possible to extend the current thesis to the 3D scenario, as is the case with autonomous underwater under actuated vehicles.

The chosen strategy for visually recognising a landmark showed to be a sufficient tool to validate both localisation and feedback laws, however, color segmentation is very prone to fail under environments with changing light values. Also, during experimental tests, there is a chance for the environment to contain a pattern similar to the one expected on the landmark, thus misleading the localisation system to guide the robot to an undesired location. Therefore, there is room to improve on the robustness of the vision system, going towards a more selective feature recognition possibly based on 3D point cloud feature recognition. Another approach would be to completely avoid 3D reconstruction such as is done in [39].

This work is meant to be applied in a modularized approach, however a global localisation can be devised by simply using several landmarks and allowing for a parallel filtering of several Kalman filter units, one for each landmark of interest, taking an approach similar to FastSLAM algorithms.

Regarding now peripheral aspects, it is also possible to extend this work by using the proposed control law to carry out exploration missions such as described in [40], or even use the capabilities provided by this thesis to implement a multi-agent cooperative system (see [41] for a state-of-the-art in distributed autonomous mobile robotics).

Bibliography

[1] Md49 and emg49 locomotion hardware. http://www.robot-electronics.co.uk/htm/md49tech.htm. Accessed: 2012-02-10.

[2] Uluc Saranli, Martin Buehler, and Daniel E Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[3] David Wooden, Matthew Malchano, Kevin Blankespoor, Andrew Howardy, Alfred A Rizzi, and Marc Raibert. Autonomous navigation for bigdog. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4736–4741. IEEE, 2010.

[4] Paul Blaer and Peter Allen. Topological mobile robot localization using fast vision techniques. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 1031–1036. IEEE, 2002.

[5] Henrik Andreasson and Tom Duckett. Topological localization for mobile robots using omnidirectional vision and local features. In *Proc. IAV*. Citeseer, 2004.

[6] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *Robotics & Automation Magazine, IEEE*, 13(3):108–117, 2006.

[7] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[8] Robert Sim, Pantelis Elinas, and James J Little. A study of the rao-blackwellised particle filter for efficient and accurate vision-based slam. *International Journal of Computer Vision*, 74(3):303–318, 2007.

[9] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loops. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 655–660. IEEE, 2005.

[10] Nathaniel Fairfield, George Kantor, and David Wettergreen. Towards particle filter slam with three dimensional evidence grids in a flooded subterranean environment. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3575–3580. IEEE, 2006.

[11] Robert Sim, Pantelis Elinas, Matt Griffin, James J Little, et al. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, volume 14, pages 9–16, 2005.

[12] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.

[13] Andrea Garulli, Antonio Giannitrapani, Andrea Rossi, and Antonio Vicino. Mobile robot slam for line-based environment representation. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2041–2046. IEEE, 2005.

[14] Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The international Journal of robotics Research*, 21(8):735–758, 2002.

[15] John J Leonard and Hugh F Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *Robotics and Automation, IEEE Transactions on*, 7(3):376–382, 1991.

[16] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696. IEEE, 2012.

[17] Yasushi Hada and Shin'ichi Yuta. A first-stage experiment of long term activity of autonomous mobile robot—result of repetitive base-docking over a week. In *Experimental Robotics VII*, pages 229–238. Springer, 2001.

[18] Gerald J Agin. *Real time control of a robot with a mobile camera*. SRI International, 1979.

[19] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *Robotics and Automation, IEEE Transactions on*, 8(3):313–326, 1992.

[20] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5):651–670, 1996.

[21] Claude Samson, Bernard Espiau, and Michel Le Borgne. *Robot control: the task function approach*. Oxford University Press, 1991.

[22] Steven B Skaar, Issac Yalda-Mooshabad, and William H Brockman. Nonholonomic camera-space manipulation. *Robotics and Automation, IEEE Transactions on*, 8(4):464–479, 1992.

[23] Olivier Lefebvre and Florent Lamiraux. Docking task for nonholonomic mobile robots. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3736–3741. IEEE, 2006.

[24] Pedro Batista, Carlos Silvestre, and Paulo Oliveira. A two-step control strategy for docking of autonomous underwater vehicles. In *American Control Conference (ACC), 2012*, pages 5395–5400. IEEE, 2012.

[25] Michael D Feezor, F Yates Sorrell, Paul R Blankinship, and James G Bellingham. Autonomous underwater vehicle homing/docking via electromagnetic guidance. *Oceanic Engineering, IEEE Journal of*, 26(4):515–521, 2001.

[26] Jin-Yeong Park, Bong-Huan Jun, Pan-Mook Lee, Fill-Youb Lee, and Jun-ho Oh. Experiment on underwater docking of an autonomous underwater vehicleisimi'using optical terminal guidance. In *OCEANS 2007-Europe*, pages 1–6. IEEE, 2007.

[27] Chris McCarthy, Nick Barnes, and Robert Mahony. A robust docking strategy for a mobile robot using flow field divergence. *Robotics, IEEE Transactions on*, 24(4):832–842, 2008.

[28] Myungsik Kim and Nak Young Chong. Direction sensing rfid reader for mobile robot navigation. *Automation Science and Engineering, IEEE Transactions on*, 6(1):44–54, 2009.

[29] Jian Chen, Warren E Dixon, M Dawson, and Michael McIntyre. Homography-based visual servo tracking control of a wheeled mobile robot. *Robotics, IEEE Transactions on*, 22(2):406–415, 2006.

[30] Ren C Luo, Chung T Liao, Kuo Lan Su, and Kuei C Lin. Automatic docking and recharging system for autonomous security robot. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2953–2958. IEEE, 2005.

[31] Hassan K Khalil. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, 2002.

[32] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall New Jersey, 1991.

[33] Roger W Brockett et al. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, pages 181–191. Defense Technical Information Center, 1983.

[34] Michele Aicardi, Giuseppe Casalino, Antonio Bicchi, and Aldo Balestrino. Closed loop steering of unicycle like vehicles via lyapunov techniques. *Robotics & Automation Magazine, IEEE*, 2(1):27–35, 1995.

[35] Openni drivers. http://www.openni.org. Accessed: 2012-06-15.

[36] Robotics operating system. http://www.ros.org/. Accessed: 2012-02-15.

[37] Ransac plane indice extraction. http://pointclouds.org/documentation. Accessed: 2012-03-25.

[38] Qualysis motion capture system. http://www.qualisys.com. Accessed: 2013-07-10.

[39] José Santos-Victor and Giulio Sandini. Visual behaviors for docking. *Computer Vision and Image Understanding*, 67(3):223–238, 1997.

[40] Alexander Zelinsky. A mobile robot exploration algorithm. *Robotics and Automation, IEEE Transactions on*, 8(6):707–717, 1992.

[41] Lynne E Parker. Current state of the art in distributed autonomous mobile robotics. In *Distributed Autonomous Robotic Systems 4*, pages 3–12. Springer, 2000.

[42] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robotos*. The MIT press, 2nd edition, 2011.

[43] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[44] Pedro Batista, Carlos Silvestre, and Paulo Oliveira. Optimal position and velocity navigation filters for autonomous vehicles. *Automatica*, 46(4):767–774, 2010.

[45] Pedro Batista, Carlos Silvestre, and Paulo Oliveira. On the observability of linear motion quantities in navigation systems. *Systems & Control Letters*, 60(2):101–110, 2011.

[46] Yoshiaki Shirai and Hirochika Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern recognition*, 5(2):99–108, 1973.