Set-membership estimation techniques
for mobile robotics applications

Ph.D. Thesis

Mauro Di Marco

Coordinatore: Prof. Giovanni Marro
Tutori: Prof. Alberto Tesi
         Prof. Antonio Vicino
To my parents.
I would like to sincerely thank my advisors Prof. Alberto Tesi, who introduced me to scientific research, and Prof. Antonio Vicino, who helped me to widen my interests. Their high ethical standards, bright intellect, interest to the research activity and dedication to students made them perfect advisors and role models.

Andrea Garulli provided such an enthusiastic and active participation to my research work, that I hardly believe this thesis would have been the same without his efforts.

Simon Lacroix, at LAAS (Toulouse) gave me precious insights on several robotics problems: thank you.

I would like to thank the students of my Ph.D. course: Gianni Bianchini and Lorenzo Giovanardi at DSI (Firenze) and Graziano Chesi at DII (Siena) for their advice and help.

Thanks to my parents, who always encouraged and supported me by any available means.

Thanks to Veronica, who was always by my side.

There are, of course, many others to whom I owe my thanks. No matter how many I list, there will always be more. To all of you my most heartfelt thanks.

Firenze, 11/1/2001
# Contents

List of Figures \hspace{2cm} iv

List of Tables \hspace{2cm} vii

Notation \hspace{2cm} viii

1 Introduction \hspace{2cm} 1

2 Set-Membership Estimation \hspace{2cm} 5
   2.1 Estimation Problem \hspace{2cm} 5
   2.2 Set-Membership Approach \hspace{2cm} 7
      2.2.1 Estimation errors \hspace{2cm} 9
      2.2.2 Some properties of pointwise estimation algorithms \hspace{2cm} 10
   2.3 Pointwise Estimators \hspace{2cm} 11
      2.3.1 Central estimator \hspace{2cm} 11
      2.3.2 Projection estimator \hspace{2cm} 12
   2.4 Set-valued Estimators \hspace{2cm} 13
   2.5 Recursive State Approximation \hspace{2cm} 14
      2.5.1 Recursive state estimation \hspace{2cm} 15
      2.5.2 Set approximation for state estimation \hspace{2cm} 16

3 Pose Estimation in Known Environments \hspace{2cm} 23
   3.1 Approaches to Localization and Pose Estimation \hspace{2cm} 23
      3.1.1 Local techniques \hspace{2cm} 24
      3.1.2 Global techniques \hspace{2cm} 25
   3.2 Localization in a Set-Theoretic Framework \hspace{2cm} 27
   3.3 Localization from Visual Angle Measurements \hspace{2cm} 34
      3.3.1 Approximation through boxes \hspace{2cm} 37
## CONTENTS

3.3.2 Approximation through paralleloptopes ................................................. 38  
3.4 Pose Estimation from Relative Orientation Measurements .......................... 41  
  3.4.1 Orientation set approximation ............................................................. 44  
  3.4.2 Recursive refinement of the approximating set ...................................... 45  

4 Simultaneous Localization And Map Building ............................................. 49  
  4.1 Approaches to Concurrent Mapping and Localization .................................. 49  
    4.1.1 Grid-based approach ........................................................................... 50  
    4.1.2 Feature-based approach ..................................................................... 50  
    4.1.3 Topological approach ........................................................................ 52  
  4.2 Set-Membership SLAM Solution ................................................................ 52  
    4.2.1 Models and problem formulation ....................................................... 53  
  4.3 State Decomposition ................................................................................. 56  
  4.4 Set Approximation ..................................................................................... 59  
    4.4.1 Approximation through boxes ........................................................... 59  
    4.4.2 Approximation through paralleloptopes ............................................. 62  
    4.4.3 Initialization and computational complexity ....................................... 64  

5 Cooperative Localization and Map Building ............................................... 67  
  5.1 Approaches to Cooperative Localization and Mapping .............................. 68  
  5.2 Set-Membership CLAM .............................................................................. 69  
  5.3 Map Fusion in the Set-Membership Framework ......................................... 73  

6 Estimation of Time to Contact ................................................................. 77  
  6.1 Approaches to Time-to-Contact Estimation .............................................. 77  
  6.2 Contour Tracking ....................................................................................... 78  
    6.2.1 Planar affine transformation of contours ........................................... 79  
    6.2.2 Discrete time dynamics of image contour .......................................... 81  
    6.2.3 Contour tracking via set-membership filtering .................................. 82  
  6.3 Computation of the time to contact ........................................................... 83  
  6.4 Exact Error Bounds on the Time-to-Contact Estimate .............................. 85  

7 Simulation Experiments .................................................................................. 89  
  7.1 Pose Estimation in Known Environments ................................................. 89  
    7.1.1 Localization from visual angle measurements .................................. 89  
    7.1.2 Pose estimation from relative orientation measurements ................ 95  
  7.2 Simultaneous Localization and Map Building .......................................... 97
CONTENTS

7.3 Cooperative Localization and Map Building . . . . . . . . . . . . . . . . . . 103
7.4 Time-to-Contact Estimation . . . . . . . . . . . . . . . . . . . . . . . . . . 108
  7.4.1 Comparison with Kalman filter approach . . . . . . . . . . . . . . 111

8 Conclusions 115

Bibliography 120
# List of Figures

2.1 Scheme of the generic estimation problem. .................................. 7
2.2 Central estimator for $\ell_2$-norm and $\ell_\infty$-norm ......................... 12
2.3 Representation of the projection algorithm in the case $\|y\|_Y = \ell_2$, and $F(x) = Fx$. ................................................................. 13
2.4 Inner and outer approximations of $FSS_y$. .................................. 14
3.1 Example of the state decomposition technique. ............................. 33
3.2 The vehicle performing one visual angle measurement. ................. 34
3.3 An example of “thickened ring” $C_{ij}$. .................................... 35
3.4 An example of measurement set $M(k)$. ...................................... 35
3.5 Suboptimal approximation algorithm for 3 landmarks. .................... 36
3.6 The robot performing two relative orientation measurements with respect to the fixed direction $X$. ............................................ 41
3.7 Measurement set $M_i$ associated to a relative orientation measurement. ......................................................... 42
3.8 Intersection between two distinct measurement sets and its projection on the $(X,Y)$ plane. .......................................................... 43
3.9 Geometric setting for orientation estimation. ................................ 44
3.10 Position estimation refinement: set $M_i(k,R_o(k|k))$. .................... 46
3.11 Position estimation refinement: new admissible position set. ............ 47
4.1 Distance $d$ and angle $\theta$ for a robot $p$ and a landmark $l_i$. .......... 54
4.2 Shape of measurement sets .......................................................... 58
4.3 The points in $V(M_{li})$ and $Z(M_{li})$ for three typical uncertainty set $M_{li}$. ................................................................. 61
4.4 Proof of proposition 4.3 (*: points of $\forall M = V(M_{li}) \cup \{A_{li}\}$). ................................................................. 65
5.1 Set $M_{li}(U_{k+n},R_{pj})$, sum of the two sets $R_{pj}$ and $C_{li}$. ........ 71
7.1 Localization with 5 landmarks: position error and uncertainty set areas . 90
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Localization with 10 landmarks: position error and uncertainty set areas</td>
<td>91</td>
</tr>
<tr>
<td>7.3</td>
<td>Position estimates ($p = [0 0]'$) for 100 different groups of 5 landmarks, and relative error bounds (20%).</td>
<td>92</td>
</tr>
<tr>
<td>7.4</td>
<td>Position estimates ($p = [-9.5 - 9.5]'$) for 100 different groups of 100 landmarks, and relative error bounds (20%).</td>
<td>92</td>
</tr>
<tr>
<td>7.5</td>
<td>Dynamic localization with 5 landmarks and absolute noise bounds.</td>
<td>93</td>
</tr>
<tr>
<td>7.6</td>
<td>Average uncertainty set areas and position error for dynamic localization with 5 landmarks and absolute noise bound</td>
<td>94</td>
</tr>
<tr>
<td>7.7</td>
<td>Dynamic localization with 8 landmarks and relative noise bounds.</td>
<td>94</td>
</tr>
<tr>
<td>7.8</td>
<td>Average uncertainty set areas and position error for dynamic localization with 8 landmarks and relative noise bound</td>
<td>95</td>
</tr>
<tr>
<td>7.9</td>
<td>Improvements due to the recursive refinement based on orientation estimates.</td>
<td>97</td>
</tr>
<tr>
<td>7.10</td>
<td>Dynamic pose estimation: robot location</td>
<td>98</td>
</tr>
<tr>
<td>7.11</td>
<td>Dynamic pose estimation: robot orientation</td>
<td>98</td>
</tr>
<tr>
<td>7.12</td>
<td>Typical results of the SM-SLAM algorithm.</td>
<td>99</td>
</tr>
<tr>
<td>7.13</td>
<td>Average uncertainties and errors for a single run of the SM-SLAM algorithm.</td>
<td>101</td>
</tr>
<tr>
<td>7.14</td>
<td>Average errors for 100 SLAM experiments, depending on bounds on exteroceptive sensors errors.</td>
<td>102</td>
</tr>
<tr>
<td>7.15</td>
<td>Average errors for 100 SLAM experiments, depending on bounds on proprioceptive sensors errors.</td>
<td>104</td>
</tr>
<tr>
<td>7.16</td>
<td>Results provided by the set-valued maps fusion algorithm</td>
<td>105</td>
</tr>
<tr>
<td>7.17</td>
<td>Comparison of single-robot (dashed lines) and multi-robot (solid line) SM-SLAM: total robot uncertainty.</td>
<td>107</td>
</tr>
<tr>
<td>7.18</td>
<td>Comparison of single-robot (dashed lines) and multi-robot (solid line) SM-SLAM: total landmark uncertainty.</td>
<td>107</td>
</tr>
<tr>
<td>7.19</td>
<td>Contours of the first and last images of two testing sequences.</td>
<td>108</td>
</tr>
<tr>
<td>7.20</td>
<td>Set-membership time-to-contact estimate in the case of exact a priori knowledge.</td>
<td>109</td>
</tr>
<tr>
<td>7.21</td>
<td>Set-membership time-to-contact estimate in the case of approximate a priori knowledge.</td>
<td>110</td>
</tr>
<tr>
<td>7.22</td>
<td>Time-to-contact estimate for motion not aligned with the image axis.</td>
<td>111</td>
</tr>
<tr>
<td>7.23</td>
<td>Comparison with Kalman filter: ideal case</td>
<td>112</td>
</tr>
<tr>
<td>7.24</td>
<td>Comparison with Kalman filter: $\hat{\tau}_0 &gt; \tau(0)$.</td>
<td>113</td>
</tr>
<tr>
<td>7.25</td>
<td>Comparison with Kalman filter: $\hat{\tau}_0 &lt; \tau(0)$.</td>
<td>113</td>
</tr>
</tbody>
</table>
List of Tables

4.1 The $k$-th step of the orthotope-based recursive SLAM algorithm. . . . . . . 62
4.2 The $k$-th step of the parallelootope-based recursive SLAM algorithm. . . . 66

7.1 Simulation results with 5 landmarks, after iterative refinement. . . . . . . 96
7.2 Simulation results with 10 landmarks, after iterative refinement. . . . . . 96
7.3 Percentage of total uncertainty reduction due to the fusion algorithm, with respect to the average total uncertainty over all initial maps. . . . . . . . 106
7.4 Percentage of total uncertainty reduction due to the fusion algorithm, with respect to the uncertainty of the best initial map. . . . . . . . . . . . . . . . 106
LIST OF TABLES
Notation

Below is a list of notations used, and a list of common abbreviations has been included as well. Generally, lower case letters (e.g., $x$, $\xi$) are used to indicate vectors, indices, functions. Uppercase letters (e.g., $X$) are used for spaces, operators, linear transformations and matrices. Sets are indicated with uppercase Greek letters (e.g., $\Xi$) or, when their structure is a priori known, with calligraphic uppercase Roman letters (e.g., $\mathcal{R}$).

$\mathbb{R}$: the set of real numbers.

$x, \xi$: system state.

$x'$: transpose of vector $x$.

$\text{diag}[x]$: a diagonal matrix. The element of $x$ are the diagonal, nonzero entries.

$I_n$: identity matrix. $I_n = \text{diag}[1, \ldots, 1]$.

$\| \cdot \|_X$: vector norm induced on the metric space $X$.

$\| \cdot \|_\infty$: $\ell_\infty$-norm. $\|x\|_\infty = \max_i |x_i|$.

$\| \cdot \|_\infty^w$: weighted $\ell_\infty$-norm. $\|x\|_\infty^w = \max_i |x_i/e_i|$.

$\| \cdot \|_1$: $\ell_1$-norm. $\|x\|_1 = \sum_i |x_i|$.

$K \otimes P$: cartesian product between sets $K$ and $P$.

$K \cap P$: intersection between sets $K$ and $P$.

$\text{co}\{P\}$: convex hull of set $P$.

$\text{cen}(P)$: Chebichev center of set $P$. See Definition 2.

$\text{rad}(P)$: Chebichev radius of set $P$. See Definition 3.

$\text{diam}(P)$: diameter of set $P$. See Definition 4.

$\mathcal{M}$: measurement set.
\( \mathcal{B} \): orthotope (box). See Definition 16.
\( \mathcal{V}(\mathcal{B}) \): set of the vertices of the box \( \mathcal{B} \).
\( \mathcal{B}_\infty \): unit ball in \( \ell_\infty \)-norm.
\( \mathcal{P} \): parallelootope. See Definition 17.
\( \mathcal{V}(\mathcal{P}) \): set of the vertices of the parallelootope \( \mathcal{P} \).
\( \mathcal{T} \): parpolygon. See Definition 18.
\( \mathcal{S} \): strip. See Equation (2.17).
\( \mathcal{R}(\mathcal{P}) \): minimum volume set in class \( \mathcal{R} \) containing the set \( \mathcal{P} \).
\( \mathcal{S}(u) \): closed B–spline.

**Common Abbreviations**

\( \text{MUS}_y \): Measurement Uncertainty Set. See Section 2.2.
\( \text{EUS}_\phi \): Estimate Uncertainty Set. See Section 2.2.
\( \text{FPS}_y \): Feasible Problem Element Set. See Section 2.2.
\( \text{FSS}_y \): Feasible Solution Set. See Section 2.2.
\( \text{UBB} \): Unknown But Bounded.
\( \text{SM} \): Set Membership.
\( \text{LP} \): Linear Programming.
\( \text{EKF} \): Extended Kalman Filter.
\( \text{SMAL} \): Simultaneous Localization And Map building.
\( \text{CLAM} \): Cooperative Localization And Map building.
n.i.u.d: independent uniformly distributed.
Chapter 1

Introduction

The main aim in the mobile robotics research field is the study and development of real autonomous robots, i.e. agents able to move in and interact with a possibly unknown environment, without any human intervention.

To be autonomous, the robot needs to be able to perform several different tasks. Among them, navigation is of fundamental importance. As a matter of fact, though this problem has been studied extensively by the research community, many fundamental questions are still unanswered. According to Leonard and Durrant-Whyte [54], the navigation problem can be summarized by the three questions: “where am I?”, “where am I going?” and “how do I get there?”. However, all of these questions usually require the knowledge of the environment the agent is navigating in. This kind of information (e.g., the location of environmental features in a given reference system) is generally called a map. Thus, mapping and localizing can be regarded as the two fundamental issues in navigation. Both these tasks can be cast as state estimation problems, the main difference between them being the dimension of the state to be estimated (see, e.g., [46, 53, 77]). Given some noisy measurements performed on the robot itself (e.g., data provided by proprioceptive sensors, giving information on displacements, velocities and attitude) and by the robot on the environment (e.g., measurements of orientation and distance provided by exteroceptive sensors), one has to deduce knowledge about the agent and the explored environment. Additionally, when long range navigation is required, the capability to provide, along with the nominal estimates, a measure of the quality of such estimates becomes of paramount importance.

Several techniques can be exploited to solve the aforementioned estimation problems: the most common approach is based on Extended Kalman Filters (EKFs). Kalman filters provide an efficient way to perform both localization and map building; the reason for their
popularity is due to the fact that they provide both a recursive solution to the navigation problem and a technique for evaluating the uncertainty of the estimates. However, in order to obtain optimal results, all the uncertainties (disturbances, model errors) affecting the system must have a Gaussian distribution. Unfortunately, real-world disturbances include correlated, biased noises whose statistical properties are seldom easy to evaluate.

In this thesis, a different approach to the estimation problem is adopted. No statistical assumption is made on errors affecting the system: the only assumption is that they are bounded in norm by some quantity. This hypothesis is motivated by the fact that a bound on measurement and/or model errors is often the only available information in several practical situations. This leads quite naturally to a set-theoretic approach to the problem: all estimates are derived in terms of feasible uncertainty sets, defined as those regions where the quantities to be estimated are guaranteed to lie, according to the available information.

Recently developed for linear estimation and identification [60, 62, 89], the set-theoretic approach is mostly addressed in a worst-case setting (i.e. estimation errors are evaluated with respect to the worst element in the feasible set), thus assuring robustness of the estimates. The main aim of this thesis is to apply these estimation techniques and devise new strategies for mobile robotics problems, that are usually nonlinear. Actually, for these nonlinear problems, approximation procedures, providing guaranteed estimates, can be devised. The computational burden of those algorithms turns out to be comparable to that of Kalman filters. Algorithms for localization, simultaneous localization and map building performed by a single robot or a colony of agents can be devised using this technique. Set-theoretic estimation can also be applied to other estimation problems, such as the recovery of the time to collision in dynamic vision applications.

The structure of the thesis is as follows.

Chapter 2 introduces the basic concepts of the set-membership estimation problem. Some fundamental definitions and properties are provided. In particular, pointwise and set-valued estimators are provided. Some algorithms for the solution of the recursive state estimation problem are presented. In particular, set approximation algorithms for linear state estimation are analyzed.

Chapter 3 deals with the problem of localization and pose estimation in known environments, i.e. when an a priori description of the environment is available to the navigating robot. After a review of the viable approaches to the problem, the problem is formulated as a nonlinear recursive state estimation. Set-membership approximation algorithms are developed for both position and orientation estimation. An algorithm providing set-valued estimates of the robot pose is presented: this algorithm is based on subsequent refine-
ments, thus allowing for the desired trade-off between quality and computational burden necessary to retrieve the sought estimate.

Chapter 4 tackles the more challenging Simultaneous Localization and Map Building (SLAM) problem, i.e. the case when an agent operates in an unknown environment. In this case, the robot must exploit the relative measurements performed on the environment to tackle simultaneously two tasks: self localization and map building. The main issue with SLAM algorithms is their complexity due to the high dimensionality of the estimation problem. The proposed algorithm, based on the set-membership approach, has a complexity which is at worst linear in the number of landmarks present in the environment.

Chapter 5 extends the aforementioned algorithm to the case of teams of cooperative robots, the so-called Collective Localization and Map Building (CLAM) problem. Both the static fusion of set-valued maps and the centralized dynamic updating of shared maps are considered.

Chapter 6 deals with another important problem in robot visual navigation: the estimation of time to contact (time needed for the agent to reach a fixed object, under the hypothesis that the relative velocity is constant). The proposed set-membership approach allows for the calculation of guaranteed upper and lower bounds on the estimated time to contact.

Chapter 7 presents a collection of simulation experiments, with the aim of illustrating and evaluating the features of the algorithms developed in the previous chapters.

Finally, Chapter 8 presents a summary of the thesis and its main contributions.
Chapter 2

Set-Membership Approach to the Estimation Theory

In this chapter, the general estimation problem in the set-membership framework is illustrated. The chapter is organized as follows: in Section 2.1 the estimation problem is introduced, along with all its characteristic elements (spaces, operators). Section 2.2 introduces the set-membership approach to the estimation problem, and some properties of the estimation algorithms. In Sections 2.3 and 2.4 pointwise and set-valued estimators are presented. Finally, in Section 2.5 the problem of recursive state estimation is described: particular attention is devoted to linear estimation problems and to set-approximation techniques that allow for the development of fast set-membership estimation algorithms.

2.1 Estimation Problem

Let us consider the following three different metric linear spaces:

- \( X \): problem element space,
- \( Y \): measurement space,
- \( Z \): solution space,

whose dimensions are \( n, m \) and \( p \), respectively; let \( \| \cdot \|_X, \| \cdot \|_Y \) and \( \| \cdot \|_Z \) be the relative norms.

Let us introduce a subset \( K \) of \( X \) and let the generic problem element \( x \) belong to \( K \) (i.e., \( K \) contains the whole \textit{a priori} information about the problem):

\[
x \in K \subset X.
\]
In addition, let us introduce two operators:

- **Solution operator** \( S(\cdot) : X \to Z \). This function maps a generic point \( x \in X \) of the problem element space in a solution element \( z = S(x) \). \( S(x) \) is the quantity that must be estimated. In several problems \( z = x \), i.e., \( S(\cdot) = I \).

- **Information operator** \( F(\cdot) : X \to Y \). This function returns all the measurements that are performed on the problem element \( x \).

Usually the quantity \( F(x) \) is not exactly known, since the measurement process is corrupted by some noise \( e \). In this context, we will assume that \( e \) is additive, hence

\[
y = F(x) + e.
\]

Under these assumptions, the generic estimation problem can be stated as follows:

*Given an unknown element \( x \in K \), find an estimate of \( S(x) \), given the a priori knowledge on \( K \) and some measurements \( F(x) \), affected by an additive noise \( e \).*

We point out that the solution of the problem depends on functions \( F(\cdot) \) and \( S(\cdot) \), and on the hypotheses on the measurement noise.

In the present work, we will consider an Unknown-But-Bounded (UBB) noise model, which can be characterized as follows:

**Definition 1 (UBB noise)** Noise \( e \in Y \) is Unknown-But-Bounded when:

- No statistical property of \( e \) is known,

- \( e \) is bounded in norm, i.e. there exists an a priori known \( \epsilon > 0 \) such that

\[
\| e \|_Y \leq \epsilon.
\]

The estimation problem consists in finding an estimator (or estimation algorithm) \( \Phi(\cdot) \), \( \Phi : Y \to Z \), which provides an approximation \( \Phi(y) \) of the quantity \( S(x) \). Said another way, one wants to determine a (generally non-linear) algorithm \( \Phi \), such that \( \Phi(y) \approx S(x) \). As a matter of fact, the output of \( \Phi(\cdot) \) can be either a single element of \( Z \) (in this case \( \Phi(\cdot) \) is called pointwise estimator) or a set \( R \) of points in \( Z \) (set-valued estimator).

A representation of the estimation problem, is depicted in Fig. 2.1.
The UBB noise assumption allows one to characterize all the elements of the estimation problem in terms of sets. In fact, provided that the noise $e$ is bounded, it is possible to evaluate all the elements in $y$ that could be the correct (noiseless) measurements. Extending this idea, we introduce some relevant sets that are usually used in the set-membership estimation theory:

- **Measurement Uncertainty Set**

$$MUS_y = \{ \hat{y} \in Y : \|y - \hat{y}\|_Y \leq \epsilon \}.$$  

This is the set containing all elements in the measurement space whose distance from the current measurement $y$, is less than $\epsilon$. Notice that $MUS_y$ is composed by all the possible “true” measurements that could generate the noisy measurement $y$. Indeed, if $x$ is the unknown problem element that produced the measurement $y$, then $F(x) \in MUS_y$.

- **Estimate Uncertainty Set**

$$EUS_\Phi = \Phi(MUS_y).$$

Given an estimation algorithm $\Phi$, this is the set of all possible “exact” answers to the estimation problem, using the noisy measurement $y$. We note that this set depends on the choice of the estimation algorithm.
- **Feasible Problem Element Set**

\[ FPS_y = \{ x \in K : \|y - F(x)\|_Y \leq \epsilon \}. \]  

(2.1)

This set contains all the problem elements that are compatible with the available information: the operator \( F(\cdot) \), the UBB hypothesis on the noise, the \textit{a priori} information \( K \) and the measurement \( y \). We observe that, should \( FPS_y \) be empty, then the measurement \( y \) is not consistent with the problem formulation (i.e., \( F(\cdot) \), \( K \) and/or \( \epsilon \) are not correct).

- **Feasible Solution Set**

\[ FSS_y = S(FPS_y) \]

This set contains all the solution elements that are compatible with the available information. If \( S(\cdot) \) is the identity operator, then \( FSS_y = FPS_y \), and the set is generally called \textit{Feasible Set}.

The structure of the aforementioned sets depends on the information operator \( F(\cdot) \): in the general case, they can be very complex (non-connected, non-convex sets). There is a special case, when the information operator is a linear function (i.e., \( F(x) = Fx \)) and \( K = X \), in which the structure of \( FPS_y \) depends exclusively on the norm chosen for the \( Y \) space. More specifically, if the \( \ell_\infty \)-norm or the \( \ell_1 \)-norm are adopted for \( Y \), then \( FPS_y \) is a polytope. On the other hand, if the \( \ell_2 \)-norm is chosen, then \( FPS_y \) is an ellipsoid. Linear set-membership estimation problem has been studied extensively by the research community, with particular reference to linear system identification [60, 62, 88, 89]. Efficient algorithms have been devised for the representation (exact or approximated) of polytopic [14, 19, 20, 64] and ellipsoidal [18, 35] sets.

When dealing with sets, it is common to introduce some geometrical features, such as \textit{center}, \textit{radius} and \textit{diameter}.

**Definition 2 (Chebichev center)** Given a set \( P \), the Chebichev center is the center of the minimum radius ball (in \( Z \)-norm) containing \( P \), i.e.,

\[ cen(P) = \arg \inf_{z \in Z} \sup_{\tilde{z} \in P} \|z - \tilde{z}\|_Z. \]

**Definition 3 (Chebichev radius)** Given a set \( P \), the Chebichev radius is given by

\[ rad(P) = \sup_{\tilde{z} \in P} \|cen(P) - \tilde{z}\|_Z. \]

Generally, \( cen(P) \) may be non-unique, and may lie outside of \( P \), even though \( P \) is a convex set. We point out that, if the set \( P \) has a symmetry center, than \( cen(P) \) coincides with it.
2.2. SET-MEMBERSHIP APPROACH

Definition 4 (Diameter) Given a set $P$, its diameter is given by

$$\text{diam}(P) = \sup_{z_1, z_2 \in P} \| z_1 - z_2 \|_Z.$$ 

It is easy to verify that

$$\text{rad}(P) \leq \text{diam}(P) \leq 2\text{rad}(P), \quad \forall P \subset Z.$$ 

2.2.1 Estimation errors

Given an estimation problem, the main issue is to choose a “good” algorithm $\Phi$. In the case of pointwise estimators, a way to measure the quality of the estimates provided by $\Phi$ is to evaluate the distance

$$\| S(x) - \Phi(y) \|_Z,$$ 

which depends both on the unknown problem element $x$ and the measurement $y$. Since the set-theoretic approach describes all the admissible problem elements ($FPS_y$) given a measurement $y$, and all the admissible measurements ($MUS_{F(x)}$) once that the problem element is known, eq. (2.2) is usually evaluated in the worst case:

Definition 5 (Y-local error) Given an estimator $\Phi$, a measurement $y$ and an error bound $\epsilon$, the Y-local error of $\Phi$ is given by

$$E_y(\Phi, \epsilon) = \sup_{x \in FPS_y} \| S(x) - \Phi(y) \|_Z$$

The Y-local error is an $a$ posteriori measurement of the quality of $\Phi$, because it implies the knowledge of the measurements and the set $FPS_y$.

Definition 6 (X-local error) Given an estimator $\Phi$, a problem element $x$ and an error bound $\epsilon$, the X-local error of $\Phi$ is given by

$$E_x(\Phi, \epsilon) = \sup_{y \in MUS_{F(x)}} \| S(x) - \Phi(y) \|_Z$$

The X-local error is an $a$ priori error measure, since it depends on the problem element (that is generally unknown), but it does not depend on the measurements $y$.

Definition 7 (Global error) Given an estimator $\Phi$ and an error bound $\epsilon$, the global error is the supremum of the X-local (or Y-local) error, i.e.,

$$E(\Phi, \epsilon) = \sup_{x \in X} E_x(\Phi, \epsilon) = \sup_{y \in Y_0} E_y(\Phi, \epsilon)$$
where $Y_0 = \{ y \in Y : FPS_y \neq \emptyset \}$.

The minimum global error of an estimator is called \textit{global information radius}

$$R(\epsilon) = \inf_{\Phi} E(\Phi, \epsilon).$$

Since every $z \in FSS_y$ could be a possible solution of the estimation problem, it can be proven that $R(\epsilon)$ can be evaluated in the following way [63]:

$$R(\epsilon) = \sup_{y \in Y_0} \text{rad}(FSS_y).$$

### 2.2.2 Some properties of pointwise estimation algorithms

Let us introduce some relevant properties of pointwise estimators.

**Definition 8 (X-local optimality)** An estimator $\Phi^*$ is said to be X-locally optimal if

$$E_x(\Phi^*, \epsilon) \leq E_x(\Phi, \epsilon), \quad \forall x \in X, \forall \Phi.$$ 

When there is no information from measurements, an X-locally optimal algorithm provides the best estimate, corresponding to the worst $y$, no matter the value of $x \in X$.

**Definition 9 (Y-local optimality)** An estimator $\Phi^*$ is said to be Y-locally optimal if

$$E_y(\Phi^*, \epsilon) \leq E_y(\Phi, \epsilon), \quad \forall y \in Y_0, \forall \Phi.$$ 

Conversely, when the measurement $y$ is available, the best algorithm is the one which minimizes the estimation error corresponding to the worst $x \in FPS_y$.

**Definition 10 (Global optimality)** An estimator $\Phi^*$ is said to be globally optimal if

$$E(\Phi^*, \epsilon) \leq E(\Phi, \epsilon), \quad \forall \Phi.$$ 

Notice that global optimality is a property weaker than local optimality. In many applications, the computational burden of optimal algorithms becomes too high, so suboptimal estimators are often pursued, provided that one is able to quantify the error introduced by them.

**Definition 11 ($\alpha$-optimality)** An estimator $\Phi^*$ is $\alpha$-optimal if

$$E(\Phi^*, \epsilon) \leq \alpha \inf_{\Phi} E(\Phi, \epsilon) \leq \alpha R(\epsilon).$$

**Definition 12 (Correctness)** An algorithm $\Phi$ is correct if

$$\Phi(F(x)) = S(x), \quad \forall x \in X.$$
Consequently, an algorithm is correct if it provides the exact solution when applied to noiseless measurements.

**Definition 13 (Interpolatory algorithm)** An algorithm $\Phi$ is said to be interpolatory if

$$\Phi(y) \in FSS_y.$$  

In other words, an interpolatory algorithm always provides an estimation compatible with the available information.

### 2.3 Pointwise Estimators

We present two important classes of pointwise estimators: the *central algorithms* and the *projection algorithms*. We recall that the set-membership theory allows one to completely describe the set of the admissible solutions of an estimation problem. Hence, the definition of a pointwise estimator boils down to the choice of a single solution. This is usually done by means of a suitable criterion, related to the Feasible Solution Set representative element in the solution set.

#### 2.3.1 Central estimator

A natural choice for the pointwise estimator is the Chebichev center of the set:

**Definition 14 (Central estimator)** A central estimator $\Phi_c$ is defined by

$$\Phi_c = cen(FSS_y).$$

The main result about central estimators is the following theorem [62].

**Theorem 2.1** A central algorithm $\Phi_c$ is $Y$-locally optimal (and, consequently, globally optimal). The minimum $Y$-local error is

$$E_y(\Phi_c, e) = rad(FSS_y).$$

More specific properties of the central algorithms depend on the structure of the function $F(\cdot)$ and the norms adopted in $Y$ and $Z$ (see, e.g., Fig. 2.2). For instance, when $\|\cdot\|_Y = \ell_{\infty}$, the following result [61] holds.

**Theorem 2.2** Let $\|\cdot\|_Z = \ell_{\infty}$. The central algorithm $\Phi_c$ can be evaluated as

$$\Phi_c(y) = cen(FSS_y) = \frac{z_i + \tilde{z}_i}{2} \quad i = 1, \ldots, p$$
Figure 2.2: Central estimator for $\ell_2$-norm (left) and $\ell_\infty$-norm (right); in the latter case the central estimation is not unique, since any point on the line between $\Phi_{c_1}(y)$ and $\Phi_{c_2}(y)$ is a center of a minimum radius ball containing $FSS_y$.

where

$$
\overline{z}_i = \sup_{x \in FSS \_y} S_i(x) \\
\underline{z}_i = \inf_{x \in FSS \_y} S_i(x)
$$

and $S_i$ is the $i$-th component of the function $S$. The $Y$-local error of $\Phi_c$ is

$$
\text{rad}(FSS_y) = \max_{i=1,\ldots,p} \frac{\overline{z}_i - \underline{z}_i}{2}
$$

In the case of linear estimation problems (i.e., $F(x) = Fx$, $F \in \mathbb{R}^{m \times n}$ and $S(x) = Sx$, $S \in \mathbb{R}^{p \times n}$), some stronger results on central estimators can be proven, such as their correctness and $X$-local optimality [62].

The main issue with central algorithms is their complexity: as a matter of fact, $\Phi_c$ can be non-unique, and is generally non-linear. For example, when applying theorem 2.2, one has to solve $2p$ optimization problems (see eqs. 2.3): these problems may be non-convex and have local extrema. Approximated algorithms can be employed to solve these problems, but they do not provide any measurement of the error introduced by the approximation.

### 2.3.2 Projection estimator

Since central algorithms often require a prohibitive computational burden, some simpler estimators are usually used. A very important class is given by the **projection estimators**.

**Definition 15 (Projection estimator)** A projection estimator $\Phi_p$ is defined by

$$
\Phi_p(y) = S(x_p),
$$
2.4. SET-VALUED ESTIMATORS

Figure 2.3: Representation of the projection algorithm in the case $\| \cdot \|_Y = \ell_2$, and $F(x) = Fx$.

where

\[ x_p = \arg \min_{x \in R} \| y - F(x) \|_Y. \]

Depending on the choice of the $Y$-norm, the projection algorithm has some straightforward interpretations: if the $\ell_2$-norm is used, $\Phi_p$ is the well-known least squares estimator (see Fig. 2.3, in the case of linear information operator). Conversely, if the $\ell_1$-norm or the $\ell_\infty$-norm are chosen, we have the minimum error sum estimator and the minimum maximum error estimator, respectively.

The main features of the projection algorithms can be summarized by the following results [83].

**Theorem 2.3** A projection estimator is always interpolatory.

**Theorem 2.4** A projection estimator is always $Y$-locally (and globally) 2-optimal.

2.4 Set-valued Estimators

In Section 2.1, we have introduced a set-valued estimator as a multi-valued function $\Phi(\cdot)$ which provides a set $P$ in the solution space.

\[ \Phi(y) = P \subset Z. \]

Since in the set-membership theory all the information concerning the estimation problem is expressed using sets, it is quite natural to describe the estimate through a set of admissible solutions. Set-valued estimators are usually divided in two classes:
CHAPTER 2. SET-MEMBERSHIP ESTIMATION

Figure 2.4: Inner and outer approximation of $FSS_y$, using ellipsoids (left) and axis aligned boxes (right).

- exact algorithms,
- approximated algorithms.

The first class aims at describing the true set of admissible solutions,
\[ \Phi(y) = FSS_y. \]

Once again, the main problem with this approach is the computational burden. As a matter of fact, in the general case, $FSS_y$ is very complicated (non-linear, non-connected regions). This is the reason why the research community has extensively studied several set approximation techniques. Approximating algorithms can be divided in inner (see, e.g., [88] and references therein) and outer approximations. The first approaches introduced in the literature use ellipsoids [26] or orthotopes [71] as approximating regions, depending on the choice of the $Y$-norm. Efficient algorithms to evaluate the minimum value outer approximations ($mOE$, $mOB$), or the maximum volume inner estimates ($MIE$, $MIB$) are usually provided (see, e.g., Fig. 2.4). Other approaches use parallelotopes [20, 87] or polytopes of fixed complexity [14].

### 2.5 Recursive State Approximation

As it will be shown in the following chapters, a large number of estimation problems in robotics applications involve dynamic system state estimation. As a matter of fact, several well-known problems (e.g., localization and/or map building in navigation applications) can be cast as state estimation of a suitable dynamic system. Since the output of these
estimation procedures is usually needed to perform other tasks (such as path planning or obstacle avoidance, in the aforementioned cases), recursive algorithms, able to update the estimates at each time step, are often pursued.

2.5.1 Recursive state estimation

The general set-membership estimation scheme, presented in the previous sections, can be easily applied to recursive state estimation. Let us consider a generic discrete-time system

\[ x(k + 1) = F(x(k), k) + w_x(k), \quad (2.4) \]

where \( F : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n \) is a discrete (time-varying) map, modeling the evolution of the system, and is generally called state transition function. The signal \( w_x(k) \in \mathbb{R}^n \) takes into account the model errors. Since a set-theoretic approach to this problem is adopted, \( w_x(k) \) is assumed to be unknown-but-bounded in some norm, i.e.,

\[ |w_x(k)| \leq \epsilon^{w_x}(k), \quad i = 1, \ldots, n. \quad (2.5) \]

In addition, let us suppose that some measurements are performed on the system state,

\[ y(k) = G(x(k), k) + v_y(k). \quad (2.6) \]

In equation (2.6), \( G : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m \) is the measurement model, and \( v_y(k) \) is an unknown-but-bounded error vector, which takes into account the measurement errors,

\[ |v_y(i)| \leq \epsilon^{v_y}(i), \quad i = 1, \ldots, m. \quad (2.7) \]

Equations (2.5) and (2.7) can also be written down in compact form as

\[ \|w_x(k)\|_\infty \leq 1 \quad (2.8) \]

\[ \|v_y(k)\|_\infty \leq 1 \]

where \( \|v\|_\infty \) is the weighted \( \ell_\infty \) norm of a vector \( v \in \mathbb{R}^n \), defined as \( \|v\|_\infty = \max_{i=1,\ldots,n} |v_i/\epsilon_i| \), \( \epsilon^{w_x}_k \in \mathbb{R}^n = [\epsilon^{w_x}_1(k) \ldots \epsilon^{w_x}_n(k)] \) and \( \epsilon^{v_y}_k \in \mathbb{R}^m = [\epsilon^{v_y}_1(k) \ldots \epsilon^{v_y}_m(k)] \).

Being the noise bounded in norm, it is possible to cast the recursive state estimation problem in terms of recursive estimation of feasible sets:

**Recursive Set-Membership State Estimation (RSMSE) Problem:** Let \( \Xi(0) \subseteq K \subseteq \mathbb{R}^n \) be a set containing the initial value of \( x(0) \). Given the dynamic model (2.4) and the measurement model (2.6), find a feasible state set \( \Xi(k|k) \subset K \) containing all the state values \( x(k) \) that are compatible with the system dynamics, the assumptions (2.8) on the disturbances, and the measurements collected up to time \( k \).
A recursive algorithm solving the RSMSE problem can be devised as shown in the following

Let $\Xi(k-1|k-1)$ be the set of all the state values that solve the RSMSE problem at time $k-1$. The algorithm evaluates $\Xi(k|k)$ in two different steps.

1. **Time Update:** during this step, using the state transition function, the algorithm evaluates all the feasible state values at time $k$ that are compatible with the state dynamics (2.4) and all the information up to time $k-1$:

$$
\Xi(k|k-1) = [F(\Xi(k-1|k-1), k-1) + \text{Diag}[\epsilon_{k-1}^w]B_{\infty}] \cap K. \quad (2.9)
$$

All the operators in eq. (2.9) are to be intended as set operators, i.e.,

- $F(\Xi(k-1|k-1), k-1)$ is the set generated by mapping each element of $\Xi(k-1|k-1)$ through the operator $F(x(k), k)$.

- $\text{Diag}[\epsilon_{k-1}^w]B_{\infty}$ is the product of a diagonal matrix, whose entries are the weights of the weighted $\ell_\infty$ norm in the state space, and the unit ball in $\ell_\infty$-norm (i.e., $B_{\infty} = \{x \in \mathbb{R}^n : \|x\|_{\infty} \leq 1\}$). This set describes all the disturbances compatible with assumption (2.7).

- The sum operator is the sum of two sets, i.e., the set obtained by adding any two elements, each taken from one of the sets. Since these sums could be outside the a priori information $K$, intersection with that set has to be performed.

2. **Measurement Update:** in the second step, the algorithm evaluates all the feasible states that are compatible with the measurement collected at time $k$. This is achieved by performing the intersection between $\Xi(k|k-1)$ and the feasible problem element set at time $k$:

$$
\Xi(k|k) = \Xi(k|k-1) \cap FPS_{y(k)}. \quad (2.10)
$$

We recall that

$$FPS_{y(k)} = \{x \in K : \|y(k) - G(x, k)\|_{\infty}^e \leq 1\}.$$

### 2.5.2 Set approximation for state estimation

The main problem with the algorithm defined by eqs. (2.9)-(2.10) concerns its computational complexity. As a matter of fact, even when $F(\cdot)$ and $G(\cdot)$ are linear (or affine) time-invariant functions, the set evaluations required by the algorithm may be a very hard task. A common way to deal with this problem is to employ one of the set approximation
2.5. **RECURSIVE STATE APPROXIMATION**

Techniques presented in the previous section. This can be achieved using suitable elements in a class of regions of fixed complexity. Since the objective is to recursively approximate the set of all feasible state values, minimum volume outer approximations of the feasible sets are usually pursued. If we denote by $R$ a class of regions with fixed structure, and by $\mathcal{R}(P)$ the minimum volume element in the class $R$ containing the set $P$, then the approximate version of the algorithm presented in the previous section can be obtained by the following recursion

$$
\mathcal{R}(0) = \mathcal{R}(\Xi(0)),
$$

(2.11)

$$
\mathcal{R}(k|k-1) = \mathcal{R}\left\{ F(\mathcal{R}(k-1|k-1), k-1) + \text{Diag}[^{\epsilon}_{k-1}]B_{\infty} \cap K \right\},
$$

(2.12)

$$
\mathcal{R}(k|k) = \mathcal{R}(\mathcal{R}(k-1) \cap FPS_y(k)).
$$

(2.13)

Recursion (2.11)-(2.13) is computationally less demanding than (2.9)-(2.10), since one has to find the approximation of sets that are obtained applying the same operators to elements of the class $\mathcal{R}$. Nevertheless, evaluating (2.12) and (2.13) can still be a challenging task. In addition, we point out that each step of the approximate algorithm introduces a certain degree of conservativeness (depending on the class $\mathcal{R}$ and the functions $F(\cdot)$ and $G(\cdot)$) leading to approximated feasible sets that can be remarkably larger than the true ones.

The choice of the class $\mathcal{R}$ may be driven by the norms used in the $X$ and $Y$ spaces. Considering the assumptions made in (2.8), boxes or parallelotopes look very appealing.

**Definition 16 (Orthotope)** An axis-aligned box (or orthotope) in $\mathbb{R}^n$ is defined as

$$
B = B(b, c) = \{ q \in \mathbb{R}^n : q = c + \text{Diag}[\epsilon]|b|, \|\alpha\|_{\infty} \leq 1 \},
$$

(2.14)

where $c \in \mathbb{R}^n$ is the center of the box and the absolute values of the elements of $b \in \mathbb{R}^n$ represent the size of the edges.

**Definition 17 (Paralleloptope)** A paralleloptope in $\mathbb{R}^n$ is defined as

$$
P = P(T, c) = \{ q \in \mathbb{R}^n : q = c + T\alpha, \|\alpha\|_{\infty} \leq 1 \},
$$

(2.15)

where $c \in \mathbb{R}^n$ is the center of the paralleloptope and $T \in \mathbb{R}^{n \times n}$ is a nonsingular matrix whose column vectors represent the edges of the paralleloptope.

The main advantage of paralleloptopes over boxes is that the orientation of boxes is fixed, while the orientation of paralleloptopes can be changed: this generally allows one to reduce the size of the approximated uncertainty sets, especially when the true feasible set stretches along some direction in the state space.
We observe that the sum and intersection of two orthotopes is still an orthotope; pre-
multiplication of an orthotope (or a parallelotope) by a nonsingular square matrix gives
a parallelotope. Moreover, it is useful to remark that a parallelotope can be rewritten as
the intersection of \( n \) strips, i.e.,
\[
P = \bigcap_{i=1}^{n} S(p_i, s_i) \tag{2.16}
\]
where
\[
S(p_i, s_i) = \{ q \in \mathbb{R}^n : |p_i q - s_i| \leq 1 \} \tag{2.17}
\]
is a strip in \( \mathbb{R}^n \) (with \( p_i \) row vector in \( \mathbb{R}^n \)) and
\[
\begin{bmatrix}
p_1 \\
\vdots \\
p_n
\end{bmatrix} = T^{-1}, \; s = T^{-1}c. \tag{2.18}
\]
A special situation that has been extensively investigated is the linear state estimation
problem, i.e., the case when \( K = \mathbb{R}^n \) and the state transition function and the measure-
ment function are linear. Moreover, the presence of a known input \( u(k) \) can be accounted
for, leading to the system
\[
\begin{align*}
x(k+1) &= A(k)x(k) + Bu(k) + G(k)d_x(k) \\
y(k) &= C(k)x(k) + d_y(k)
\end{align*}
\]
In this case, if parallelotopes (orthotopes) are considered as approximating regions \( \mathcal{R} \),
eqs. \eqref{eq:2.12}-\eqref{eq:2.13} become
\[
\begin{align*}
\mathcal{R}(0|0) &= \mathcal{R}\{\Xi(0)\}, \tag{2.19} \\
\mathcal{R}(k|k-1) &= \mathcal{R}\{A(k-1)\mathcal{R}(k-1|k-1) + B(k-1)u(k-1) + \\
&\quad G(k-1)\text{Diag}[^{\infty}_{k-1}\mathcal{B}_\infty]\}, \tag{2.20} \\
\mathcal{R}(k|k) &= \overline{\mathcal{R}}(\mathcal{R}(k|k-1) \cap FPS_{y(k)})\}. \tag{2.21}
\end{align*}
\]
In order to evaluate the output of the recursive algorithms, one has to be able to solve
the two following approximation problems:

A1 Compute the minimum volume box (parallelotope) containing the vector sum of two
parallelotopes (eq. 2.20);

A2 Compute the minimum volume box (parallelotope) containing the intersection of
a box (parallelotope) with a polytope (since in eq. \( (2.21) \), \( FPS_{y(k)} = \{ x \in \mathbb{R}^n : \\
\|y(k) - C(k)x\|_\infty \leq 1 \} \), which is a polytope, i.e. the intersection of \( m \) strips in \( \mathbb{R}^n \).
Solutions to problem A1 can be obtained using some available results from set-membership estimation theory, as it is shown in the following.

**Approximation through boxes.**

**Proposition 2.1** Let \( R(k-1) = B(b, c) \). Then

\[
R\{A(k-1)B(b, c) + B(k-1)u(k-1) + G(k-1)\text{Diag}[e_{k-1}^w]B_{\infty}\} = B(\bar{b}, \bar{c})
\]

where

\[
\bar{c} = A(k-1)c + B(k-1)u(k-1)
\]

\[
\bar{b}_i = \|\{A(k-1)\text{Diag}[b] G(k-1)\text{Diag}[e_{k-1}^w]\} e_i \|_1,
\]

for \( i = 1, \ldots, n \) and \( e_i \) denotes the \( i \)-th column of the identity matrix \( I_4 \).

**Proof:** See [41].

**Approximation through parallelotopes.**

Let us introduce the *parpolygon* set:

**Definition 18 (Parpolygon)** An \( m \)-segment parpolygon in \( \mathbb{R}^n \) is defined as

\[
T_m = T(T, c) = \{ q \in \mathbb{R}^n : q = c + T\alpha, \|\alpha\|_\infty \leq 1, T \in \mathbb{R}^{n \times m}, \alpha \in \mathbb{R}^m \}. \tag{2.22}
\]

A parpolygon is the vector sum of \( m \) segments, i.e., \( T_m = c + \sum_{i=1}^{m} T(t_i, 0) \), where \( t_i \) is the \( i \)-th column of matrix \( T \). Equivalently, a parpolygon is the intersection of \( \binom{m}{n-1} \) strips in \( \mathbb{R}^n \), that are symmetric with respect to \( c \). Clearly, a parallelotope is a special parpolygon with \( m = n \).

Approximate solutions of problem A1 for parallelotopic approximations can be obtained by exploiting results in [20, 87]. In particular, the following theorem is needed.

**Theorem 2.5** Consider the parpolygon \( T = c + \sum_{i=1}^{n+1} T(t_i, 0) \), \( t_i \in \mathbb{R}^n \), and let the matrix \( [t_1, \ldots, t_n] \) be nonsingular, with \([p'_1, \ldots, p'_n]' = [t_1, \ldots, t_n]^{-1}\). Then the parallelotope of minimum volume outbounding \( T \) is given by

\[
P(T) = c + \sum_{i=1}^{n+1} T(T_i, 0_n) = P(T^*, c)
\]
where $T^*$ is obtained by replacing the $i^*$-th column of $[t_1, \ldots, t_n]$ by $t_{n+1}$, and
\[
 i^* = \arg \max_{1 \leq i \leq n+1} |p_i t_{n+1}| \quad \text{with} \quad p_{n+1} t_{n+1} \triangleright 1
\]
\[
 \bar{T}_i = r_i t_i, \quad 1 \leq i \leq n+1, \quad i \neq i^*
\]
\[
 r_i = 1 + (|p_i t_{n+1}|/|p_{i^*} t_{n+1}|), \quad 1 \leq i \leq n+1, \quad i \neq i^*.
\]
This result can be exploited to recursively compute approximate solutions of problem A1.

Let us consider eq. (2.20) and assume $R(k|k-1) = P(T, c) \subset \mathbb{R}^n$. According to (2.22), equation (2.20) can be rewritten as
\[
 R(k|k-1) = P(T, c),
\]
where
\[
 T_{2n} = T([A(k-1)T \mid G(k-1)\text{Diag}[\epsilon_{k-1}^w]], A(k-1)c + B(k-1)u(k-1)).
\]
A suboptimal solution to problem (2.23) can be computed by recursively applying Theorem 2.5 to $T_{2n}$, according to the following algorithm.

0. Let $\bar{c} = A(k-1)c + B(k-1)u(k-1)$ and set
\[
 \bar{T}^{(0)} = A(k-1)T, \quad [t_1 \ldots t_n] = G(k-1)\text{Diag}[\epsilon_{k-1}^w];
\]
1. For $i = 1, \ldots, n$ compute
\[
 P_{\bar{T}^{(i)}} = P(T([\bar{T}^{(i-1)} | t_i], \bar{c}))
\]
using Theorem 2.5;
2. Let $R(k|k-1) = P(\bar{T}^{(n)}, \bar{c})$.

Clearly, the above construction guarantees that
\[
 R(k|k-1) = P(\bar{T}^{(n)}, \bar{c}) \supseteq A(k-1)R(k-1|k-1) + B(k-1)u(k-1) + G(k-1)\text{Diag}[\epsilon_{k-1}^w]B_{\infty}.
\]
A suboptimal solution can also be obtained for problem A2, based on the following result [87].

**Theorem 2.6** Consider the convex polytope $\mathcal{X} = \bigcap_{i=1}^{n+1} S(p_i, s_i)$, and let the matrix $[p'_1, \ldots, p'_n]'$ be nonsingular, with $[t_1, \ldots, t_n] = ([p'_1 \ldots p'_n]')^{-1}$. Then, the parallelotope of minimum volume outbounding $\mathcal{X}$ is given by
\[
 \overline{P}\{\mathcal{X}\} = \bigcap_{i=1}^{n+1} S(\bar{p}_i, \bar{s}_i) = P(T^*, c^*)
\]
where $T^* = (P^*)^{-1}$, with $P^*$ obtained by replacing the $i^*$-th row of $[\vec{p}_1 \ldots \vec{p}_n]'$ by $\vec{p}_{n+1}$, $c^* = T^* \vec{s}$, and

$$i^* = \arg \max_{1 \leq i \leq n+1} |p_{n+1} t_i| \quad \text{with} \quad p_{n+1} t_{n+1} \triangleq 1$$

$$\bar{p}_i = \frac{2}{r_i^+ - r_i^-} p_i, \quad \bar{s}_i = \frac{2}{r_i^+ - r_i^-} \left( s_i + \frac{r_i^+ + r_i^-}{2} \text{sign}(p_{n+1} t_i) \right), \quad i = 1, \ldots, n + 1$$

$$r_i^+ = \begin{cases} \min(-1 + \frac{1 - \epsilon_{n+1}^-}{|p_{n+1} t_i|}, 1), & p_{n+1} t_i \neq 0, \quad i = 1, \ldots, n \\ 1, & p_{n+1} t_i = 0 \end{cases}$$

$$r_i^- = \begin{cases} \max(1 - \frac{1 + \epsilon_{n+1}^-}{|p_{n+1} t_i|}, -1), & p_{n+1} t_i \neq 0, \quad i = 1, \ldots, n \\ -1, & p_{n+1} t_i = 0 \end{cases}$$

$$r_{n+1}^+ = \min(\epsilon_{n+1}^+, 1)$$

$$r_{n+1}^- = \max(\epsilon_{n+1}^-, -1)$$

$$\epsilon_{n+1}^+ = (p_{n+1} \hat{x} - s_{n+1}) + \sum_{i=1}^{n} |p_{n+1} t_i|$$

$$\epsilon_{n+1}^- = (p_{n+1} \hat{x} - s_{n+1}) - \sum_{i=1}^{n} |p_{n+1} t_i|.$$

This result can be employed to devise a suboptimal solution to problem A2 for both orthotopical and parallelotopical approximations. Indeed, let us assume $R(k|k-1) = \mathcal{P}(T, c)$ and let $FPS_{y(k)} = \bigcap_{i=1}^{m} S(p_i, s_i)$ be the feasible set due to the $m$ measurements at time $k$. Then, eq. (2.21) can be expressed as

$$R(k|k) = \overline{\mathcal{P}}\{\mathcal{P}(T, c) \cap FPS_{y(k)}\}, \quad (2.24)$$

i.e., the intersection of $n + m$ strips. A suboptimal solution to problem (2.24), which can be computed in polynomial time, is then obtained by recursively applying Theorem 2.6 as in the following algorithm.

0. Set $\tilde{T}^{(0)} = T$, $\tilde{c}^{(0)} = c$.

1. For $i = 1, \ldots, m$, use Theorem 2.6 to compute

$$\mathcal{P}(\tilde{T}^{(i)}, \tilde{c}^{(i)}) = \overline{\mathcal{P}}\{\mathcal{P}(\tilde{T}^{(i-1)}, \tilde{c}^{(i-1)}) \cap S(p_i, s_i)\};$$

2. Let $R(k|k) = \mathcal{P}(\tilde{T}^{(n)}, \tilde{c}^{(n)})$. 
Remark 2.1 Optimal solutions of problems (2.20)-(2.21) has exponential computational complexity in $n$, the state size and $m$, the number of measurements performed. The use of the proposed suboptimal algorithms reduces the computational complexity to $O(n^2)$, for both boxes and parallelotopes [20].

Unfortunately, when either the measurement equation or the system model is nonlinear, there is no systematic way to get the approximations in (2.12) and (2.13). Clearly, ad hoc techniques can be developed, exploiting the specific structure of the nonlinearities.
Chapter 3

Pose Estimation in Known Environments

Self-localization, i.e. giving a reliable answer to the question “Where am I?”, is the first step towards real long range autonomous navigation. The requisite of being able to know its own position (and, eventually, orientation) with respect to some known locations, implicitly requires some description of the environment or, in other words, a map.

Even when the environment where the robot is operating is stationary and the map is provided a priori, the localization task in the given map is not a trivial one. In the last three decades, the robotics research community has spent a lot of efforts to formulate successful strategies and devise efficient algorithms to tackle the localization problem.

The chapter is organized as follows: after a brief overview on the viable approaches to the pose estimation problem, in Section 3.2 the problem is formulated as a recursive state estimation in a set-theoretic framework. Set approximation algorithms, using orthotopes and parallelotopes as classes of approximating regions, are developed in Section 3.3 for the localization problem. The case when both position and orientation must be estimated is considered in Section 3.4.

3.1 Approaches to Localization and Pose Estimation

There are basically two different approaches to localization, each attacking a different problem [40]. Tracking or local techniques aim at developing algorithms that incrementally update the robot position: this is the general situation when a navigation task is to be performed. These techniques require that the initial location of the robot is (roughly) known, and they typically cannot recover if they lose track of the robot position. A different
approach is taken by global techniques. These are designed to estimate the robot position even under global uncertainty, in order to solve the so-called wake-up robot problem, i.e. to localize a robot in a known environment without any prior knowledge on its position. These methods usually have higher computational and storage requirements.

3.1.1 Local techniques

The most common and basic method for performing tracking localization is dead reckoning: this method uses the information available to the robot from proprioceptive sensors. In fact, dead reckoning (derived from “deduced reckoning” of sailing days) is a simple mathematic procedure that allows one to determine the current position of the robot by advancing some previous position, through knowledge of course and velocity information over a given length of time [12]. Unfortunately, pure dead reckoning techniques are prone to systematic errors, and the integration intrinsically implied by the method causes the absolute position error to grow without bounds [23].

Nevertheless, a great effort has been spent to refine as much as possible these methods. Borenstein and Feng [13] have developed a systematic procedure for the measurement and correction of odometry errors. Other methods require more sophisticated equipment: this is the case for inertial navigation. Initially developed for deployment on aircraft, its principle of operation involves continuous sensing of accelerations in the three directional axes and their integration over time to derive velocity and position. Barshan and Durrant-Whyte [4] develop a refined Inertial Navigation System, whose sophisticated error model is used in an Extended Kalman Filter (EKF), to drastically improve its performances. An interesting approach to dead reckoning, based on exteroceptive sensors, has been recently proposed by Mallet et al. [57]: by computing the motion parameters between two stereo frames on the basis of a set of 3D-point-to-3D-point matches, this technique is able to estimate the six parameters of the robot displacement in any kind of environment. The presence of landmarks is not required (provided that the scene is textured enough so that pixel-based stereovision works well).

A common procedure for improving tracking localization techniques is to combine dead reckoning with some additional localization methods, generally based on measurements performed on the environment. Fusion of the two classes of information is usually obtained via some filtering techniques, the EKF being the most used. The uncertainty on the robot position is generally modeled as an unimodal Gaussian distribution. All these approaches use some model for the robot motion and for the measurements process. Leonard and Durrant-Whyte [53] extract beacons from sonar scans and match them with a geometri-
3.1. APPROACHES TO LOCALIZATION AND POSE ESTIMATION

A precise description of the environment. The beacons consist of cylinders, corners and planes. Holenstein and Badreddin [46] develop a similar method, which can be applied during motion at working speed. Cox [23] matches distances measured by infrared sensors with a line description of the environment. In the above cases, white Gaussian zero-mean random processes are generally used as uncertainty models. Even when these assumptions are satisfied, while the Kalman filter is guaranteed to converge, the EKF is not. Moreover, real-world uncertainties do not always satisfy the above statistical assumptions. Therefore, some other techniques have been investigated. Hanebeck et al. [44, 45] adopt a mixed statistical/set-theoretic approach to the estimation problem, employing ellipsoids as outer approximating regions. They also show how Kalman filter can be sometimes too optimistic (thus producing erroneous estimates) when noises do not satisfy the aforementioned assumptions. Kiefer et al. [49] propose a set-valued nonlinear set estimator, with a scheme analogous to the Kalman filter, under the hypothesis of bounded errors. The approximation of the set inversion required by eqs. (2.1) and (2.10) is provided. The main issue with this approach is its complexity, due to the latter operation.

3.1.2 Global techniques

In the development of global localization techniques, several different methods have been proposed. Drumheller [30] uses a search tree to perform matching between the sensors measurement and the available map. A sonar barrier test, based on the constraints on sensor data, is introduced to check for inconsistencies in the position guesses. Talluri and Aggarwal [82] match line segments in the plane. The robot pose is computed using a Hough transform variation, to limit the number of regions of the map that must be examined. Several methods are based on the localization and identification of nearby landmarks (distinctive visual events, defining a unique, re-acquirable direction in three dimensional space [56]) and a consequent triangulation procedure. Sugihara [80] assumes that the relative directions of the landmarks can be sensed, but not their distance from the robot. He proposes an algorithm for performing localization from its data which performs at best as \(O(n^2)\), when landmarks can be distinguished. In order to reduce the computational burden of the method, Sutherland and Thompson [81] search for the best landmark triple, on which they perform the triangulation. In their geometrical analysis of the problem, extending the qualitative approach of Levitt and Lawton [56], they use an UBB error model, implicitly introducing the idea of feasible sets. Also Betke and Gurvits [8] consider the case where the landmarks are distinguishable. By representing the landmark position as a complex number, they obtain a linear-time algorithm with a least-squares error.
criterion. Mouaddib and Marhic [66] consider the case of undistinguishable landmarks: in this case, matching is performed by Interpretation Tree search. The robot’s evolution field is subdivided into rectangles, that are validated only if they contain at least one coherent configuration (i.e., a geometric configuration in which each pair landmark/measurement matches relatively to a bounded error). The subdivision process is iterated until the desired estimation quality is reached.

Elfes [32] uses an occupancy grid representation of the environment. Each cell in the grid has a score between -1 and 1 (certainly empty and certainly occupied, respectively). Localization is then performed by determining the relative position between a local and global occupancy grid that maximizes the product of the values at the corresponding cells in the grids.

In recent years, several probabilistic methods have been investigated. The Markov localization paradigm, for instance, computes a probability distribution over the space of possible robot position. Alternating movement steps (when the probability distribution is updated taking into account the additional uncertainty induced by dead-reckoning errors) to measurement steps (when the new data allows for the reduction of robot position uncertainty), these methods can be seen as “generalized” EKFs, where the probability distribution is not constrained to be normal and unimodal. Nourbakhsh et al. [69] use a partially observable Markov model to perform localization on a topological map: this approach allows them to handle ambiguous situations, while maintaining a relatively small state space. Simmons and Koenig [74] combine a partially observable Markov model with an evidence grid, in order to perform localization with both topological and metric information. Fox et al. developed an expectation-maximization algorithm, based on a fine-grained, grid-based discretization of the state space [37], which has been successfully applied also in crowded, dynamic environments [40]. They also give an active localization method (i.e., the algorithm chooses the direction the vehicle has to move to in order to get better knowledge on its location) using the Markov localization technique [39].

Another interesting technique, which can be considered as a variant of the Markovian approach, is presented by Olson [70]. His algorithm performs localization by comparing an elevation map, generated using the robot sensors at the current position (local map), to a previously generated map (global map, eventually constructed during the exploration). The two maps are compared according to a maximum-likelihood similarity measure based on the Hausdorff distance between images. The best relative position between the maps is then found using a branch-and-bound search in the robot pose space.
3.2 Localization in a Set-Theoretic Framework

In this section, the outdoor localization problem is tackled in the set-membership framework. An algorithm capable to recursively provide a guaranteed estimation of the robot position, provided that all the disturbances satisfy some bounded-in-norm constraints, is introduced. This means that the robot position is guaranteed to lie in the sets provided by the algorithm: this can be very helpful, e.g., to perform “safe” path planning.

In the following, the localization problem is cast in the form previously presented in Section 2.5. Let us consider a mobile robot navigating in a 2D environment, and let $\xi(k) \in \mathbb{R}^n$ be the state vector of the vehicle at time $k$. Depending on the assumptions made on the robot dynamics and on the quantities that must be estimated, the dimension of $\xi(k)$ will vary, generally from $n = 2$ (in the case of slow dynamics and position estimation only) to $n = 6$ (when dynamics are not negligible and also orientation has to be estimated).

Consequently, also the robot model will vary. In the case of slow dynamics, if translation (and rotation) measurements $u(k)$ are available from proprioceptive sensors, the vehicle state evolution can be described by the linear discrete-time model

$$\xi(k + 1) = \xi(k) + u(k) + G(k)w(k)$$

(3.1)

where $\xi(k) = [x(k), y(k), \theta(k)]'$ coincides with the robot pose $p(k)$ (to be estimated), $\theta(k)$ is the robot heading and $w(k) \in \mathbb{R}^3$ models the errors affecting measurements $u(k)$ (possibly shaped by a suitable matrix $G(k)$). Should the dynamics be not negligible, a quite general model is

$$\xi(k + 1) = A(k)\xi(k) + B(k)u(k) + G(k)w(k)$$

(3.2)

where $\xi(k) = [\dot{x}(k), \dot{y}(k), \dot{\theta}(k)]' = [x(k), y(k), \theta(k), \dot{x}(k), \dot{y}(k), \dot{\theta}(k)]'$, while matrices $A(k)$, $B(k)$ and $G(k)$ can be obtained from any nonlinear dynamics $\dot{\xi}(t) = F(\xi(t), u(t), w(t))$, through discretization and linearization with respect to the current robot state. Generally a linear model is used to describe the vehicle dynamics, since this allows for efficient set approximations. Notice that also linearization errors can be taken into account by suitably modifying the assumptions on $w(k)$.

The measurements model depends on the sensors used. Generally, when the robot is moving in a static environment, measurements depend only on the robot pose $p(k)$. As a consequence, the outputs of the sensors can be described by equations of the form

$$c_i(k) = \mu_i(p(k)) + v_i(k) \quad i = 1, \ldots, m$$

(3.3)

where $m$ is the number of measurements performed at time $k$, $\mu_i(p)$ is a (nonlinear) function modeling the $i$-th measurement and $v_i(k)$ is the noise affecting that measurement.
CHAPTER 3. POSE ESTIMATION IN KNOWN ENVIRONMENTS

Since the problem is stated in a set-theoretic framework, disturbances $w(k)$ and $v_i(k)$ are assumed to be unknown-but-bounded. More specifically, weighted $\ell_\infty$ bounds are considered
\[
\|w(k)\|_{\ell_\infty}^w \leq 1 \tag{3.4}
\]
\[
\|v(k)\|_{\ell_\infty}^v \leq 1
\]
where $\epsilon_k^w$ and $\epsilon_k^v$ are the two weights vectors introduced in (2.8) (i.e., $\epsilon_k^w = [\epsilon_1^w(k) \ldots \epsilon_n^w(k)]'$ and $\epsilon_k^v = [\epsilon_1^v(k) \ldots \epsilon_m^v(k)]'$). The second assumption allows one to define, for each measurement in (3.3), a set where the navigator pose is allowed to lie (i.e., the feasible set $FPS_{y_i}$), whose expression is
\[
\mathcal{M}_i = C(c_i, \epsilon_i^v) = \{p \in \mathbb{R}^3 : c_i - \epsilon_i^v \leq \mu_i(p) \leq c_i + \epsilon_i^v\} \tag{3.5}
\]
(dependence on time $k$ has been neglected).

Obviously the shape of these sets depends on the function $\mu_i(\cdot)$ in (3.3): they can generally be nonconvex, nonconnected and/or unbounded. Since there are $m$ distinct measurement equations, the pose at time $k$ is constrained to belong to all sets $\mathcal{M}_i$, $i = 1, \ldots, m$ and consequently in their intersection
\[
\mathcal{M}(k) = \bigcap_{i=1}^{m} \mathcal{M}_i(k). \tag{3.6}
\]

Notice that $\mathcal{M}(k)$, called measurement set, is the feasible pose set at time $k$, depending on the $m$ measurements.

According to the dynamics model (eq. (3.1) or (3.2)), the measurement model (3.3) and UBB assumptions (3.4), the localization problem can be written down in the form presented in section 2.5.1.

**Set-Membership Localization Problem:** Let $\Xi(0) \subset K$ be a set containing the initial state $\xi(0)$. Given the dynamics model (3.2) and the measurement model (3.3), find at each time $k = 1, 2, \ldots$ the feasible state set $\Xi(k|k) \subset K$ containing all the state values $\xi(k)$ that are compatible with the robot dynamics, the assumptions (3.4) on the disturbances, and the measurements collected up to time $k$.

The solution to the above problem is given by recursion (2.9)-(2.10), slightly modified to account for the linear model:
\[
\Xi(0|0) = \Xi(0) \tag{3.7}
\]
3.2. LOCALIZATION IN A SET-THEORETIC FRAMEWORK

\[
\Xi(k|k-1) = A(k-1)\Xi(k-1|k-1) + B(k-1)u(k-1) + G(k-1)\text{Diag}[^w_{e_{k-1}}|B_\infty] \tag{3.8}
\]

\[
\Xi(k|k) = \Xi(k|k-1) \cap \mathcal{M}(k) \tag{3.9}
\]

**Remark 3.1** The algorithm defined by recursion (3.7)-(3.9) enjoys two interesting properties:

- The states contained in the set \(\Xi(k|k)\) are compatible with all the available information: it is guaranteed that no admissible state can lie outside this set, and in the meanwhile, for each \(\bar{\xi}(k) \in \Xi(k|k)\) there is at least one admissible realization of the disturbances for which the true state value is exactly \(\bar{\xi}(k)\).

- As a set estimator, the algorithm can perform global localization. Ambiguous situations (occurring when \(\Xi(k|k)\) is the union of disjointed subsets) are naturally carried over in the time evolution - measurement update until the information provided by measurements removes the ambiguity. Notice that, in this situation, any choice of pointwise estimator would lead to weird results: a central estimator could provide an estimate outside the feasible set, while a projection estimator would necessarily choose a point inside one of the subsets, that could eventually disappear during a subsequent measurement update (the state estimation would consequently “jump”).

As stated in Section 2.5, the main issue with set-theoretic algorithms is their computational complexity: equation (3.9) generally implies the evaluation of the intersections of complex nonlinear sets in \(\mathbb{R}^6\). To tackle this problem, a common choice is to approximate the true sets with elements of a class of reduced complexity. Since errors are bounded in \(\ell_\infty\)-norm, two classes of regions turn out to be computationally appealing: orthotopes (see eq. (2.14)) and parallelotopes (see eq. (2.15)). When using these approximations, algorithm (3.7)-(3.9) becomes (see Section 2.5.2)

\[
\mathcal{R}(0|0) = \mathcal{R}\{\Xi(0)\} \tag{3.10}
\]

\[
\mathcal{R}(k|k-1) = \mathcal{R}\{A(k-1)\Xi(k-1|k-1) + B(k-1)u(k-1) + G(k-1)\text{Diag}[^w_{e_{k-1}}|B_\infty}\} \tag{3.11}
\]

\[
\mathcal{R}(k|k) = \mathcal{R}\{\mathcal{R}(k|k-1) \cap \mathcal{M}(k)\} \tag{3.12}
\]

To be able to apply recursion (3.10)-(3.12) one must be able to solve, the two following problems:

L1 Compute the minimum volume box (parallelotope) containing the vector sum of two parallelotopes (eq. 3.11);
L2 Compute the minimum volume box (parallelotope) containing the intersection of a box (parallelotope) with the measurement set \(M(k)\) (eq. 3.12).

When box approximations are considered, problem L1 can be solved exactly by exploiting Proposition 2.1. For parallelotopic approximations, a suboptimal algorithm for solving problem A1 has been presented in Section 2.5.2. Nevertheless, when the dimension of the system state is reduced (as in the case of slow robot dynamics, when \(\xi(k) \in \mathbb{R}^2\)), it is also reasonable to pursue the optimal solution.

First, the concept of tightness between sets must be defined.

**Definition 19 (Tightness)** Let \(K, P \subset \mathbb{R}^n\) be two sets satisfying \(K \supseteq P\). Then, \(K\) is tight with respect to \(P\) if there exist no \(\alpha \in (0, 1)\) and \(x \in \mathbb{R}^n\) such that \(\alpha K + x \supseteq P\).

The following result, on minimal parallelotopic outbounding of polytopic sets holds [87].

**Lemma 3.1** Consider the bounded convex polytope \(X \subset \mathbb{R}^n\), defined by the intersection of \(n+k\), \(k \geq 1\), tight strips \(S_i\), i.e. \(X = \cap_{i=1}^{n+k} S_i\). Then, the minimum volume parallelotope \(P\) containing \(X\) is given by

\[
\mathcal{P}(X) = \bigcap_{j=1}^{n} S_j^*
\]

where the set of strips \(S_i, i = i_1^*, \ldots, i_n^*\) includes \(n\) out of the \(n+k\) given strips.

Let us introduce the two parallelotopic sets

\[
\mathcal{P}(T_1, c_1) = A(k-1)R(k-1|k-1)
\]
\[
\mathcal{P}(T_2, c_2) = G(k-1)\text{Diag}[e_{i_k^*}]B_{\infty} + B(k-1)u(k-1)
\]

**Theorem 3.1** Let \(\mathcal{P}(T_1, c_1)\), and \(\mathcal{P}(T_2, c_2)\) be given. Set \([t_1 \ t_2 \ t_3 \ t_4] = [T_1 \ T_2] \in \mathbb{R}^{4 \times 2}\).

Then

\[
\mathcal{P}\{\mathcal{P}(T_1, c_1) + \mathcal{P}(T_2, c_2)\} = \mathcal{P}(\bar{T}, \bar{c}),
\]

where

\[
\bar{c} = c_1 + c_2,
\]
\[
\bar{T} = [\alpha_{i_j^*} t_{i_j^*} \alpha_{i_j^*} t_{j^*}] \frac{1}{\eta_{i_j^*}},
\]

and

\[
\alpha_i = \sum_{k=1 \atop k \neq i}^{4} \eta_{ij},
\]
3.2. LOCALIZATION IN A SET-THEORETIC FRAMEWORK

\[ \eta_{ij} = \eta_{ji} = |\det(T_1^{-1}[t_i t_j])|, \quad (3.16) \]

\[ (i^*, j^*) = \arg \min_{i,j=1,\ldots,4} \nu_{ij}, \quad (3.17) \]

\[ \nu_{ij} = \begin{cases} \frac{\alpha_i \alpha_j}{\eta_{ij}} & \text{if } \eta_{ij} \neq 0 \\ +\infty & \text{if } \eta_{ij} = 0 \end{cases}, \quad (3.18) \]

**Proof.** First notice that

\[ P(T_1, c_1) + P(T_2, c_2) = \tau + P(T_1, 0) + P(T_2, 0) = T([T_1 T_2], \tau) \]

which is a 4-dimensional parpolygon in \( \mathbb{R}^2 \), with center \( \tau \) given by (3.13). Hence, one has

\[ \overline{P} \{ T([T_1 T_2], \tau) \} = \tau + \overline{P} \{ P(T_1, 0) + P(T_2, 0) \}. \]

Moreover, from the properties of parpolygons

\[ P(T_1, 0) + P(T_2, 0) = T([T_1 T_2], 0) = \bigcap_{i=1}^{4} S(p_i, 0) \]

i.e., the 4-dimensional parpolygon in \( \mathbb{R}^2 \) is the intersection of 4 tight strips, for suitable vectors \( p_i, 1 \leq i \leq 4 \), satisfying the following conditions

1. orthogonality to the segments \( t_i \) of the parpolygon
   \[ \overline{p}_i t_i = 0, \quad 1 \leq i \leq 4; \quad (3.19) \]

2. tightness of the strips \( S(p_i, 0) \) to the parpolygon, i.e.,
   \[ \max_{q \in T([T_1 T_2], 0)} | \overline{p}_i q | = 1 \]

   which becomes, by substituting \( q = [T_1 T_2] \gamma, \| \gamma \|_{\infty} \leq 1, \)

   \[ \max_{\gamma \in \{-1,1\}} \left| \sum_{k=1}^{4} \overline{p}_i \gamma k t_k \right| = \sum_{k=1}^{4} |p_i t_k| = 1. \quad (3.20) \]

By applying Lemma 3.1, one has that the minimum area parallelopotope containing \( P(T_1, 0) + P(T_2, 0) \) is given by

\[ \overline{P}(P(T_1, 0) + P(T_2, 0)) = S_j \bigcap S_i = P(T, 0) \]

for some \( i, j, 1 \leq i, j \leq 4, i \neq j \), and a suitable matrix \( T \in \mathbb{R}^2 \).

It remains to show that \( T \) is given by (3.14)-(3.18). Now,

\[ S_j \bigcap S_i = \left\{ q : \left\| \begin{bmatrix} \overline{p}_j \\ \overline{p}_i \end{bmatrix} q \right\|_{\infty} \leq 1 \right\} = \left\{ q : q = T \xi, \| \xi \|_{\infty} \leq 1 \right\} \]
with
\[ \begin{bmatrix} \overline{p}_j \\ \overline{p}_i \end{bmatrix}^{-1} = [\beta_i \nu_i] \]

for suitable positive \( \beta_i, \beta_j \) satisfying
\[ |\overline{p}_j \beta_i \nu_i| = 1 \quad , \quad |\overline{p}_i \beta_j \nu_j| = 1 \]

and hence
\[ \beta_i = \frac{1}{|\overline{p}_j \nu_i|} \quad , \quad \beta_j = \frac{1}{|\overline{p}_i \nu_j|}. \]

Let us consider, for example, \( \beta_i \) and let \( v = \frac{\nu_i}{\nu_j} \). Exploiting (3.19) and (3.20), one has
\[ \beta_i = \frac{4}{\sum_{k=1}^{4} |\overline{p}_j \nu_k|} \quad , \quad \beta_j = \frac{1}{|\overline{p}_i \nu_j|} \]

thus proving (3.14), with \( \alpha_j \) and \( \eta_{ij} \) given by (3.15) and (3.16) respectively.

Finally, it has to be shown which pair \( i, j \) must be selected in order to minimize the area of the parallelotope \( \mathcal{P}(\overline{T}, \overline{\pi}) \). Since
\[ \text{area}\{\mathcal{P}(\overline{T}, \overline{\pi})\} = 4 \left| \det \overline{T} \right| \]

and, if \( \eta_{ij} \neq 0 \),
\[ \left| \det \overline{T} \right| = \beta_i \beta_j |\det[t_i \ t_j]| = \frac{\alpha_j \alpha_i}{\eta_{ij} \eta_{ij}} |\det T_i| |\eta_{ij}| = \nu_{ij} |\det T_i|, \]

then (3.17)-(3.18) hold and the proof is completed.

Concerning problem L2, the evaluation of the minimum volume box (parallelotope) containing the intersection of a box (parallelotope) and some nonlinear set, such as those defined by (3.6), requires strategies tailored to the specific problem.

A quite general and viable technique to obtain simpler intersections to deal with is state decomposition. When using this technique, sets are first projected onto suitable subspaces. Thus simpler sets are obtained. Set intersections are then performed in these subspaces, and afterwards the overall feasible set is approximated by the cartesian product of the evaluated subsets. This strategy introduces an approximation in the sense that information on the dependence between two state elements belonging to different projected subspaces is lost. Nevertheless, the set obtained by the final product always contains the real set. The following 2D linear example shows all the aforementioned properties.
3.2. LOCALIZATION IN A SET-THEORETIC FRAMEWORK

Figure 3.1: Example of the state decomposition technique.

**Example 3.1** Let us consider the following set $\Delta \subset \mathbb{R}^2$ defined by

$$\begin{align*}
|x_1| & \leq 1 \\
|x_2 - x_1| & \leq 1
\end{align*}$$

and depicted in grey in Fig. 3.1. By projecting the set onto the two subspaces described by the coordinate axes, one obtains the two monodimensional sets

$$\begin{align*}
\Xi_1 &= \{x_1 : -1 \leq x_1 \leq 1\}, \\
\Xi_2 &= \{x_2 : \min_{y \in [-1, 1]} |x_2 - y| \leq 1\} = \{x_2 : -2 \leq x_2 \leq 2\}.
\end{align*}$$

These two sets are reported as thick lines on the $x_1$ and $x_2$ axis, respectively. Performing the cartesian product of the two projected region, the set $\overline{\Delta} = \Xi_1 \otimes \Xi_2$, shown as a dashed box in Fig. 3.1, is obtained. It can be easily verified by inspection that this set has the following properties:

- $\overline{\Delta}$ contains the set $\Delta$.

- The use of $\overline{\Delta}$ in place of $\Delta$ introduces some approximation, due to the loss of information about the relationship between the two components of $(x_1, x_2)$.

- Since each component is considered separately, the set $\overline{\Delta}$ is also the minimum orthotope containing the set $\Delta$. 
In the next two sections, it will be shown how state decomposition and set approximation techniques can be employed to solve problem L2, for different types of measurements performed by a robot navigating in an environment described by a landmark-based map. In Chapter 4 these techniques are used to tackle the more challenging problem of simultaneous localization and map building.

### 3.3 Localization from Visual Angle Measurements

In this section, the localization problem, in the setup presented in [8, 81], is tackled. Let us consider a robot navigating in a 2D environment, and able to perform measurements on $m$ distinguishable landmarks, that have been previously identified and whose coordinates are known (i.e., a map of the surroundings is available to the robot). In particular, for each pair of landmarks $l_i, l_j$, the robot is able to measure the visual angle $\theta_{ij}$, i.e. the angle formed by the rays from the vehicle position to each landmark (see Fig. 3.2). This implies that, at each time instant $k$, up to $m(m - 1)/2$ measurements are collected. This kind of measurements are available if, e.g., the robot is equipped with a panoramic camera [8, 81]. Each measurement can be modeled as (3.3). Since these measurements do not depend on the robot orientation, it is not possible to retrieve any information on this state variable: the estimation of the sole robot position $(x, y)$ will be pursued.

Being some odometric measurements available, the robot dynamics can be described by the simple model (3.1), where $\xi(k) = p(k) = [x(k) \ y(k)]'$ $\in \mathbb{R}^2$ is the quantity to be estimated. Under the UBB noise assumption for the exteroceptive measurements ($|v_{ij}(k)| \leq \epsilon_{ij}(k), \ i,j = 1, \ldots, m, \ i < j$), a bounded set where the robot is constrained.
3.3. LOCALIZATION FROM VISUAL ANGLE MEASUREMENTS

Figure 3.3: An example of “thickened ring” $C_{ij}$.

Figure 3.4: An example of measurement set $M(k)$, when 4 landmarks are identified in the environment.

To lie is defined for each angle measurement $\theta_{ij}$. This set can be described as

$$C_{ij} = C(\theta_{ij}, \epsilon_{ij}) = \{ p \in \mathbb{R}^2 : \theta_{ij} - \epsilon_{ij} \leq \mu_{ij}(p) \leq \theta_{ij} + \epsilon_{ij} \}. \quad (3.21)$$

From a geometrical point of view, $C_{ij}$ is a “thickened ring”, i.e. the region between two circular arcs with the same extreme points, the landmarks $l_i, l_j$ [56, 81] (an example is reported in Fig. 3.3).

Therefore, at each time $k$, the measurement set will be the intersection of $m$ thickened rings

$$M(k) = \bigcap_{i,j=1}^{m} C_{ij}(k),$$

defining a nonlinear feasible set. An example of such a set, when 4 landmarks are identified, is reported in Fig. 3.4.
Chapter 3. Pose Estimation in Known Environments

Figure 3.5: a,b,c: Suboptimal algorithm steps for three landmarks. d: Optimal (light) and suboptimal (dark) estimated sets. Dashed boxes are the previous-step set estimates.

Being the structure of the measurement set $\mathcal{M}(k)$ known, it is possible to develop suitable strategies to solve problem (3.12).

The computational burden involved in the exact evaluation of the intersection of nonconvex sets is generally far too high; hence a suboptimal solution based on recursive minimum area approximations is pursued: for each pair $i,j$, the minimum area set in the class $\mathcal{R}$ containing the intersection of $\mathcal{C}_{ij}$ with the current approximating region is computed. Consequently, problem (3.12) boils down to computing $\overline{\mathcal{R}}\{\mathcal{R} \cap \mathcal{C}_{ij}\}$. An example of recursive set approximation according to the above strategy is shown in Fig. 3.5, in the case of 3 landmarks, when orthotopes are chosen as approximating sets. In Fig. 3.5d, the output of both the optimal and suboptimal strategies are reported.

In the following, we will separately derive the algorithm for the two classes of approximating sets.
3.3. LOCALIZATION FROM VISUAL ANGLE MEASUREMENTS

3.3.1 Approximation through boxes

When the suboptimal approach is adopted, for each angle measurement it is required to evaluate the set approximation \( B \cap C_{ij} \), where \( B \) is the current approximating orthotope. Let us introduce the following sets:

- \( \delta C_{ij} \): the boundary of each thickened ring \( C_{ij} \) in (3.21). Notice that \( \delta C_{ij} = \delta C_{ij}^+ \cup \delta C_{ij}^- \), where \( \delta C_{ij}^+ = C(\theta_{ij} + \epsilon_{ij}, 0) \) and \( \delta C_{ij}^- = C(\theta_{ij} - \epsilon_{ij}, 0) \).

- \( V(B) \): the set of the vertices of the box \( B \).

The following result allows one to compute the minimum area box containing \( B \cap C_{ij} \).

**Lemma 3.2** Let \( C_{ij} \) be given by (3.21), and \( B \) be an assigned box. Then

\[
\overline{B}(B \cap C_{ij}) = \overline{E}
\]

where \( E = E_1 \cup E_2 \cup E_3 \cup E_4 \), and

\[
\begin{align*}
E_1 &= V(B) \cap C_{ij} \\
E_2 &= \delta B \cap \delta C_{ij} \\
E_3 &= \{l_i, l_j\} \cap B \\
E_4 &= \delta \left( \overline{E}(\delta C_{ij}^-) \right) \cap \delta C_{ij}^- \cap B
\end{align*}
\]

**Proof.** See [41].

Notice that sets \( E_i, \ i = 1, \ldots, 4 \) contain a finite number of points, which are easily computed. For example, the points in \( E_2 \) are given by the intersection of a circle and a segment, while \( E_4 \) requires the computation of the tangency point between a line and a circle. As a consequence of Lemma 3.2, the minimum area box containing \( B \cap C_{ij} \) can be computed knowing only the finite set of points \( E \). This is stated in the next result.

**Theorem 3.2** Let \( B(b, c) \subset \mathbb{R}^2 \) be a given box and \( C_{ij}(k) \) be given by (3.21). Moreover, let \( E \) be defined as in Lemma 3.2. Then

\[
\overline{B}(B(b, c) \cap C_{ij}) = B(\overline{b}, \overline{c})
\]

where

\[
\begin{align*}
\overline{c}_1 &= \frac{\overline{x}_1 + \overline{x}_2}{2} & \overline{c}_2 &= \frac{\overline{y}_1 + \overline{y}_2}{2} \\
\overline{b}_1 &= \frac{\overline{x}_1 - \overline{x}_2}{2} & \overline{b}_2 &= \frac{\overline{y}_1 - \overline{y}_2}{2}
\end{align*}
\]

and

\[
\overline{x} = \max_{p \in E} x, \quad \overline{y} = \max_{p \in E} y, \quad \underline{x} = \min_{p \in E} x, \quad \underline{y} = \min_{p \in E} y.
\]
Using Theorem 3.2, the following recursive procedure can be employed to find a guaranteed outer approximation of the set $\mathcal{R}(k|k)$ in problem (3.12)

0. Set $\mathcal{B} = \mathcal{R}(k|k - 1)$;
1. For $i,j = 1, \ldots, m$, $i < j$, set
   \[ \mathcal{B} = \mathcal{B}(\vec{b}, \tau) \] as in Theorem 3.2.
2. Set $\mathcal{R}(k|k) = \mathcal{B}$.

**Remark 3.2** As a consequence of Theorem 3.2, the number of operations required to solve problem (3.12) is $O(m^2)$, where $m$ is the number of landmarks, if all possible landmark pairs are considered. Since the number $m$ is usually not very high, the method is suitable to be used also in real-time problems.

**Remark 3.3** The algorithm proposed supplies an approximate set-valued description of the quantities to be estimated. In many applications, a nominal estimation value is required. Considering the optimality features of the central algorithms (see Section 2.3) and being the computation of the center of a box an easy task, it is a reasonable choice to consider the center of the approximating box as the nominal estimate of the robot position at each time step.

### 3.3.2 Approximation through parallelotopes

Also for the choice of parallelotopes as approximating sets, it is possible to adopt a strategy similar to that proposed in the previous section. Since parallelotopes have more degrees of freedom than boxes, the recursive intersection with $C_{ij}$ and approximation procedure will usually shrink the parallelotope and change its shape (i.e., the orientation of its strips). If the two operation of shrinking (reducing the parallelotope area without changing the orientation of the strips) and reshaping the parallelotope are considered separately, it is possible to extend the results devised for boxes approximations.

Let us denote by $\mathcal{P}_T\{P\}$ the minimum parallelotope of fixed shape $T$ containing the set $P$, i.e. $\mathcal{P}_T(P) = \mathcal{P}(TD^*, c^*)$, where

\[
\begin{align*}
\{D^*, c^*\} &= \arg \min \det(TD) \\
\text{s.t.} & \\
\{D \text{ diagonal}, c \in \mathbb{R}^2 : \mathcal{P}(TD, c) \supseteq P\}.
\end{align*}
\]

It is then possible to extend Lemma 3.2 to the case of parallelotopes of fixed orientation.
Lemma 3.3 Let $C_{ij}$ be given by (3.21), and $P = P(T, c)$ be an assigned parallelotope. Then

$$P_T \{P \cap C_{ij}\} = \mathcal{F}(\mathcal{F})$$

where $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 \cup \mathcal{F}_4$, and

$$\begin{align*}
\mathcal{F}_1 &= \mathcal{V}(P) \cap C_{ij}, \\
\mathcal{F}_2 &= \delta P \cap \delta C_{ij}, \\
\mathcal{F}_3 &= \{l_i, l_j\} \cap P, \\
\mathcal{F}_4 &= \delta \left(\mathcal{F}\delta C_{ij}^{-}\right) \cap \delta C_{ij}^+ \cap P.
\end{align*}$$

As for Lemma 3.2, the sets $\mathcal{F}_i$, $i = 1, \ldots, 4$ contain a finite number of points that are very easy to compute. Using Lemma 3.3, it is possible to give an explicit characterization of the set $P_T \{P \cap C_{ij}\}$. This is illustrated by the following result.

Proposition 3.1 Let a parallelotope $P = P(T, c) \subset \mathbb{R}^2$ and a thickened ring $C_{ij}$ be given, and let $[p_1', p_2'] = T^{-1}$. Moreover, let $\mathcal{F}$ be defined as in Lemma 3.3. Then,

$$P_T \{P \cap C_{ij}\} = \bigcap_{i=1}^2 S(\overline{p}_i, \overline{s}_i)$$

where

$$\begin{align*}
\overline{p}_i &= \frac{2}{\gamma_i^+ - \gamma_i^-} p_i, \\
\overline{s}_i &= \frac{\gamma_i^+ + \gamma_i^-}{\gamma_i^+ - \gamma_i^-}, \\
\gamma_i^+ &= \max_{q \in \mathcal{F}} p_i q, \\
\gamma_i^- &= \min_{q \in \mathcal{F}} p_i q, \\
i &= 1, 2.
\end{align*}$$

Notice that the optimal solution $D^*, c^*$ of problem (3.22) can be easily obtained from $\overline{p}_i$, $\overline{s}_i$, $i = 1, 2$ in Proposition 3.1, using parallelootope definitions (2.15)-(2.16) and equation (2.18).

Now, let $\overline{P}_T$ denote the tightened parallelotope $\overline{P}_T \{P \cap C_{ij}\}$ and let us try to further reduce the volume of $\overline{P}_T$ by changing its shape, i.e. by substituting new strips to the $S(\overline{p}_i, \overline{s}_i)$, $i = 1, 2$, characterized by Proposition 3.1. Let us consider the set

$$\mathcal{F}_s = \mathcal{F}_3 \cup \{\delta C_{ij}^+ \cap \overline{P}_T\}.$$

Recall that $\mathcal{F}_3$ is the set of landmarks lying inside the initial parallelotope $P$ (and hence also inside $\overline{P}_T$), while $\delta C_{ij}^+$ is the inner circle bounding the thickened ring $C_{ij}$. It is clear that $\mathcal{F}_s$ is made of a finite set of points; more precisely, it can contain from zero up to eight points (the maximum number of intersections between a circle and a parallelotope in $\mathbb{R}^2$). However, the most common situations are the following:
1. \( \mathcal{F}_s \) is empty;

2. \( \mathcal{F}_s \) contains two points.

When \( \mathcal{F}_s \) is empty it is easy to show \([41]\) that reshaping cannot reduce the size of the approximating parallelotope, i.e.,

\[
\overline{\mathcal{P}}\left\{ \mathcal{P}(T, c) \cap \mathcal{C}_{ij} \right\} = \overline{\mathcal{P}}_T.
\]

Let us consider case 2 and set \( \mathcal{F}_s = \{q_1, q_2\} \). Let \( v' \) be a row vector orthogonal to \( q_1 - q_2 \) (i.e., satisfying \( v'(q_1 - q_2) = 0 \)), and denote by \( \mathcal{S}_s\{P\} \) the minimum width strip orthogonal to \( v' \) containing the set \( P \). From the definition of strip, this is given by \( \mathcal{S}(\alpha^* v', s^*) \), where

\[
\{\alpha^*, s^*\} = \arg \max_{\alpha, s} \quad \text{s.t.} \quad \{\alpha > 0, s \in \mathbb{R} : \mathcal{S}(\alpha v, s) \supset \mathcal{Z}\}.
\]

When \( P = \mathcal{C}_{ij} \), the solution of the above problem simply requires the computation of the tangency point between a line of direction \( q_1 - q_2 \) and the circle \( \delta \mathcal{C}_{ij} \) (the outer circle bounding \( \mathcal{C}_{ij} \)). The strip \( \mathcal{S}(\alpha^* v', s^*) \) is used as a candidate strip for replacing one of the two strips of \( \overline{\mathcal{P}}_T \), given by Proposition 3.1. The minimum volume parallelotope \( \mathcal{P}(\hat{T}, \hat{c}) \) containing the intersection of the three strips \( \mathcal{S}(\overline{p}_1, \overline{z}_1), \mathcal{S}(\overline{p}_2, \overline{z}_2) \) and \( \mathcal{S}(\alpha^* v', s^*) \), can be easily obtained by applying Theorem 2.6. Notice that by construction we have

\[
\mathcal{P}(\hat{T}, \hat{c}) = \overline{\mathcal{P}}\left\{ \mathcal{S}(\alpha^* v', s^*) \cap \mathcal{S}(\overline{p}_1, \overline{z}_1) \cap \mathcal{S}(\overline{p}_2, \overline{z}_2) \right\} \supset \left\{ \mathcal{P}(T, c) \cap \mathcal{C}_{ij} \right\}.
\]

**Remark 3.4** It has been observed that \( \mathcal{F}_s \) can contain more than two points, but this happens only in special situations which occur very rarely in the localization problem. In these cases, one can obviously keep the tightened parallelotope \( \overline{\mathcal{P}}_T \{\mathcal{P} \cap \mathcal{C}_{ij}\} \) given by Proposition 3.1 as approximating parallelotope containing \( \mathcal{P} \cap \mathcal{C}_{ij} \), without selecting any new strip.

Summing up, the following recursive procedure can be employed to find a guaranteed outer approximation of the set \( \mathcal{R}(k|k) \) in problem (3.12).

0. Set \( \mathcal{P} = \mathcal{R}(k|k - 1) \);

1. For \( i, j = 1, \ldots, m, i < j \), set

\[
\mathcal{P} = \begin{cases} 
\mathcal{P}(\hat{T}, \hat{c}) \text{ in (3.24),} & \text{if } \mathcal{F}_s \text{ contains two points,} \\
\overline{\mathcal{P}}_T \{\mathcal{P} \cap \mathcal{C}_{ij}\}, & \text{otherwise;}
\end{cases}
\]

2. Set \( \mathcal{R}(k|k) = \mathcal{P} \).
3.4 Pose Estimation from Relative Orientation Measurements

In this section, the previous problem will be extended to the estimation of both position and orientation (i.e., the robot pose) from exteroceptive measurements. We recall that the orientation of the robot is defined as the angle between the $X$-axis of the absolute reference system in which the robot moves ($X, Y$) and the $X$-axis of the robot centered reference system ($X^{(r)}, Y^{(r)}$). To achieve this goal, one needs some measurements to be functions of the robot position and orientation. These measurements can be obtained from a panoramic camera, by measuring the angular distance of each landmark from a fixed direction (see Fig. 3.6).

Similarly to the previous section, the robot is supposed to navigate in an environment that can be described by a 2D landmark map: $(x_{li}, y_{li}), i = 1, \ldots, m$ will be the coordinates of the $i$-th landmark $l_i$. Assuming the availability of odometric measurements and under the slow robot dynamics assumptions, the robot motion can be described by a simple linear model (3.1), where $\xi(k) = p(k) = [x(k), y(k), \theta(k)]'$ $\in \mathcal{Q} = \mathbb{R}^2 \times [-\pi, \pi]$ and $\theta(k)$ is the robot orientation. Therefore, the time update (3.11) in the localization procedure, can be
performed via the approximation algorithms presented in Section 2.5.2. As long as the measurement update (3.12) is concerned, for each landmark \( l_i \), the navigator is able to measure the angle formed by its orientation and the straight line connecting it to the landmark. Consequently, the measurement model is given by

\[
\mu_i(\xi(k)) = \operatorname{atan2}(\Delta y_i(k), \Delta x_i(k)) - \theta(k),
\]

where \( \Delta y_i(k) = y_i(k) - y(k) \), \( \Delta x_i(k) = x_i(k) - x(k) \) and \( \operatorname{atan2}(a, b) \) is the four quadrant inverse tangent. Equation (3.25) implies that the measurement set \( \mathcal{M}_i \) in (3.5) has the following expression

\[
\mathcal{M}_i = \{ p \in \mathcal{Q} : c_i - \epsilon_i^\nu \leq \operatorname{atan2}(\Delta y_i, \Delta x_i) - \theta \leq c_i + \epsilon_i^\nu \}.
\]

This set is a portion of \( \mathcal{Q} \) delimited by two helicoids: a part of this set, limited in the \((X, Y)\) plane, is shown in Fig. 3.7.

As a matter of fact, evaluating the intersections of these unbounded sets is far from being a trivial task. Hence, the state decomposition approach proposed in Section 3.2 will be employed. Unfortunately, being the sets \( \mathcal{M}_i \) unbounded, they are not suitable for projection on any subspace. Nevertheless, it can be easily verified that the intersection of any two measurement sets, related to distinct landmarks, is always bounded. Moreover, it turns out that its projection on the \((X, Y)\) plane is a thickened ring. In fact, evaluating the difference between two measurement equations (3.3), with \( \mu(\cdot) \) given by (3.25), one obtains

\[
c_i - c_j = \theta_{ij} + v_i - v_j
\]

Figure 3.7: Measurement set \( \mathcal{M}_i \) associated to a relative orientation measurement.
where \( \theta_{ij} \) is the visual angle between two distinct landmarks, as introduced in the previous section. The associated uncertainty region is the thickened ring

\[
C_{ij} = \{ [x \ y]' : |c_i - c_j - \arctan_2(\Delta y_i, \Delta x_i) + \arctan_2(\Delta y_j, \Delta x_j)| \leq \epsilon_i^v + \epsilon_j^v \}.
\]

The intersection between two different measurement sets \( \mathcal{M}_i \) and its projection on the \((X, Y)\) plane is reported in Fig. 3.8. As shown in the previous section, given \( m \) landmarks, one can produce up to \( m(m - 1)/2 \) different thickened rings, considering all possible landmark pairs. Since (3.27) does not depend on the robot orientation, it is possible to evaluate separately a 2D set \( \mathcal{R}_l(k|k) \) containing the feasible vehicle positions, and an interval \( \mathcal{R}_o(k|k) \) for the admissible robot orientations. Consequently, problem (3.12) boils down to solving the following two approximation problems:

**B1** Compute the minimum volume 2D box (parallelotope) \( \mathcal{R}_l(k|k) \) containing the intersection of a box (parallelotope) in \( \mathbb{R}^2 \) with the set \( \mathcal{C} = \bigcap_{i>j} C_{ij} \) (projection of (3.12) on the \((X, Y)\) plane);

**B2** Compute an interval \( \mathcal{R}_o(k|k) \) containing the orientations set of the vehicle (projection of (3.12) on the subspace spanned by \( \theta \)).

Notice that the cartesian product \( \mathcal{R}_l(k|k) \otimes \mathcal{R}_o(k|k) \) is the desired approximation of the feasible pose set. A suboptimal solution to problem B1 has been proposed in Section 3.3, so an algorithm to solve B2 must be devised.
3.4.1 Orientation set approximation

A technique for computing an interval approximation for the robot orientation (i.e., a suboptimal solution to problem B2), when boxes are chosen as approximating class for the robot position, is presented in this section. Let $c_i(k)$, $i = 1, \ldots, m$ be the angle measurements at time instant $k$ and $R_l(k|k)$ be a box containing all the vehicle positions compatible with the robot dynamics and the measurements up to time $k$, computed as described in the previous section.

From each angle measurement $c_i(k)$, relative to a landmark $l_i$ lying outside the box $R_l(k|k)$, it is possible to derive an interval containing the actual robot orientation in the following way. Let us denote by $(X,Y)$ the absolute reference system, and by $(X^{(r)}, Y^{(r)})$ the robot reference system. In Fig. 3.9, the landmark $l_i$ and the box $R_l(k|k)$ are shown. Lines $s_{i1}$ and $s_{i2}$ connect landmark $l_i$ and the box vertices $V$ and $\overline{V}$, chosen so that the angle $\varphi_i$ with the $X$ axis is respectively minimized ($\varphi_i$) and maximized ($\overline{\varphi}_i$). The cartesian equations describing these lines are known, being the landmark and the vertices coordinates known. The robot orientation $\theta$, i.e. the angle between $X^{(r)}$ and $X$, is unknown, and so are the angles $\alpha_i$, $\overline{\alpha}_i$ between $X^{(r)}$ and lines $s_{i1}$ and $s_{i2}$. From simple geometric considerations, it turns out that

$$\overline{\alpha}_i \leq \mu_i \leq \alpha_i \quad (3.28)$$

$$\varphi_i = \alpha_i + \theta \quad (3.29)$$

$$\overline{\varphi}_i = \overline{\alpha}_i + \theta \quad (3.30)$$
in which \( \varphi \) and \( \varphi_i \) are known. Substituting (3.29)-(3.30) into (3.28), one gets
\[
\varphi - \mu_i \leq \theta \leq \varphi_i - \mu_i. \tag{3.31}
\]

Since only a noisy measurement \( c_i \) of the relative orientation \( \mu_i \) is available, (3.31) has to be modified according to (3.3) and assumption (3.4) on error bounds, thus giving
\[
\varphi - c_i - \epsilon_i^v \leq \theta \leq \varphi_i - c_i + \epsilon_i^v
\]
which provides a restriction on the admissible orientation values. If \( M_o \triangleq [\varphi - c_i - \epsilon_i^v, \varphi_i - c_i + \epsilon_i^v] \) denotes the interval of feasible orientations corresponding to measurement \( c_i \), it is possible to obtain the desired outer approximation of the orientation set by computing the intersection
\[
M_o(k, R_l(k|k)) = \bigcap_{i: l_i \notin R_l(k|k)} M_o_i. \tag{3.32}
\]

Notice that (3.32) provides an approximation of the exact feasible orientation set, because \( R_l(k|k) \) is an approximation of the exact feasible position set. The overall localization procedure, based on state decomposition and set approximation, can be formalized in the following algorithm, where \( \Pi = [I_2 \ 0_{2 \times 1}] \), \( \Omega = [0 \ 0 \ 1] \).

0. Let \( R(0|0) = \overline{B}\{\Xi(0)\} \).

For \( k=1,2,\ldots \)

1. Find \( R(k|k-1) \) in (3.11) (computed as in Section 2.5.2);

2. Find \( R_l(k|k) \supset \overline{B}\{\Pi R(k|k-1) \cap C\} \) as in Section 3.3;

3. Let \( R_o(k|k) = \Omega R(k|k-1) \cap M_o(k, R_l(k|k)) \);

4. Let \( R(k|k) = R_l(k|k) \otimes R_o(k|k) \).

The set \( R(k|k) \) is the desired outer approximation of the admissible pose set, as it satisfies \( \Xi(k|k) \subset R(k|k) \) by construction.

### 3.4.2 Recursive refinement of the approximating set

Once that the interval \( R_o(k|k) \), containing all the feasible robot orientations, is computed, it is generally possible to use such information to refine the position estimate by further reducing the area of the box \( R_l(k|k) \) containing the feasible position set. In fact, if \( \theta \) is allowed to vary inside the interval \( R_o(k|k) \), then the robot position must lie in a sector of the \((X,Y)\) plane (the grey region in Fig. 3.10 is an example). Let
\[
\hat{\theta}(k) = \frac{\theta + \vartheta}{2}, \quad a(k) = \frac{\vartheta - \theta}{2},
\]
where \( \theta = \min \mathcal{R}_o(k|k) \) and \( \Theta = \max \mathcal{R}_o(k|k) \). It is possible to define, for each landmark \( l_i \), a set containing the vehicle position as

\[
\mathcal{M}_l(k, \mathcal{R}_o(k|k)) = \{ [x(k) y(k)]' \in \mathbb{R}^2 : |c_i + \hat{\theta}(k) - \tan_2(\Delta y_i, \Delta x_i)| \leq c_i^r(k) + a(k) \}.
\]

It is easy to verify that this is a sector delimited by two lines originating in \((x_{i}, y_{i})\) (denoted by \( s_i \) and \( s_i \) in Fig. 3.10).

Considering the intersection

\[
\mathcal{M}_l = \bigcap_{i=1}^{n} \mathcal{M}_l_i
\]

a new admissible position set is obtained. One can try to reduce the uncertainty on the robot position by evaluating the minimum area box containing the intersection of such set with \( \mathcal{R}_l(k|k) \) (see, e.g., Fig. 3.11)

\[
\mathcal{R}_l^{(i)}(k|k) = \overline{B} \left\{ \mathcal{R}_l^{(i-1)}(k|k) \cap \mathcal{M}_l(k) \right\}
\]

(3.33)

where the superscript \( (i) \) is the number of refinement iterations performed on the set \( \mathcal{R}_l(k|k) \).

Clearly, once the position uncertainty set \( \mathcal{R}_l(k|k) \) has been tightened, also the interval of admissible orientations \( \mathcal{R}_o(k|k) \) can be reduced. This can be done by repeating step 3 of the algorithm illustrated in the previous section, using the position box \( \mathcal{R}_l^{(i)}(k|k) \) computed in (3.33).
3.4. POSE ESTIMATION FROM RELATIVE ORIENTATION MEASUREMENTS

Iterating the computation of $\mathcal{R}_l(k|k)$ and $\mathcal{R}_o(k|k)$, it is possible to successively reduce the uncertainty affecting position and orientation estimates. This is done by choosing

$$
\mathcal{R}_l^{(0)} = \mathcal{B}\{\Pi \mathcal{R}(k|k-1) \cap \mathcal{C}\}
$$

$$
\mathcal{R}_o^{(0)} = \Omega \mathcal{R}(k|k-1) \cap \mathcal{M}_o(k, \mathcal{R}_l^{(0)}(k|k))
$$
as in Section 3.4.1, and then repeating for $i = 1, 2, \ldots$

$$
\mathcal{R}_l^{(i)} = \mathcal{B}\{\mathcal{R}_l^{(i-1)}(k|k) \cap \mathcal{M}_l(k, \mathcal{R}_o^{(i-1)})\}
$$

$$
\mathcal{R}_o^{(i)} = \mathcal{R}_0^{(i-1)}(k|k) \cap \mathcal{M}_o(k, \mathcal{R}_l^{(i)}(k|k)).
$$

Solution of (3.34) requires the computation of the minimum box containing the intersection of a box and $m$ sectors $\mathcal{M}_l$. This can be done by solving 4 linear programming problems with $2m + 4$ constraints

$$
\begin{align*}
\min & \quad c'p \\
\text{subject to} & \quad Ap \leq b
\end{align*}
$$

where $p = [x \ y]'$, and the cost $c$ depends on which vertex of the box must be computed, i.e.,

$$
c' = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}
$$

(3.37)
In order to obtain less computational demanding algorithms, suboptimal solutions based on recursive set approximation procedures, similar to those presented in Section 2.5.2, can be adopted [20]. By processing sequentially the intersection of each set $\mathcal{M}_l$, with the current approximating box, the computational burden reduces to $O(m)$ operations for each set approximation (3.34). Notice that (3.35) can be easily computed exactly, as it requires the intersection of at most $m + 1$ intervals (see (3.32)). Several heuristic criteria (e.g. based on the number of iterations and the relative area reduction) can be formulated to stop the above recursion. In practice, very few iterations are necessary to significantly reduce the size of the uncertainty boxes. Results of simulation experiments, aiming at testing the algorithms presented in this chapter for both static and dynamic setting, are presented in Section 7.1.
Chapter 4

Simultaneous Localization And Map Building

A step forward (with respect to localization in known environments) towards true autonomous navigation, the Simultaneous Localization and Map Building (SLAM) problem asks whether it is possible for an autonomous vehicle to start in an unknown location of an unknown environment, and then to incrementally build a map of this environment while simultaneously using this map to compute absolute vehicle location.

In many respects, the SLAM problem is “a Holy Grail of the autonomous vehicle research community” [28]. A solution to the SLAM problem would be really valuable in every application where absolute position and/or precise map information can not be obtained, such as autonomous planetary exploration, subsea navigation and autonomous airborne vehicles. Consequently, the problem has been subject of substantial research since the birth of the robotics research community and, before then, in areas such as manned vehicle navigation systems or geophysical surveying.

The chapter is organized as follows: Section 4.1 summarizes the possible approaches to the SLAM problem. Section 4.2 presents a formulation of the SLAM problem in a set-theoretic framework. The approximations needed to devise fast algorithms are studied in Sections 4.3 and 4.4, where algorithms are proposed for both orthotopic and parallelotopic approximations.

4.1 Approaches to Concurrent Mapping and Localization

The approaches to the SLAM problem can be roughly categorized into three main classes: (i) “grid-based”, (ii) “feature-based” and (iii) “topological”. The following sections sum-
marize the story and the state of the art for each of these approaches to robot navigation.

4.1.1 Grid-based approach

The grid-based approach to the SLAM problem is probably the simplest. It was first introduced by Moravec and Elfes [65] and extended and utilized by others [33, 78, 75, 92]. Also the “metric” approaches to SLAM [84] belong to this class of techniques. In grid-based techniques, the environment is divided into a grid of cells of some fixed physical size. Each cell is assigned a probability for the presence of an object at the cell location. A cell that is certainly filled with an object would be assigned the value 1, while a cell that is for sure empty would have the value 0. These algorithms, by pixelating the environment, generate a map of the environment. To localize itself, the robot produces a new local grid describing the environment. Performing a correlation analysis between the current global map and the new local map, the location of the robot that results in the highest correlation is defined as the new estimate of the robot location. Once that the vehicle position is determined, the measurements that define the local certainty grid are merged with the old certainty grid, thereby increasing its accuracy. The major advantages of this method are that it is intuitive, easy to be implemented and extended to higher dimensions [78]. On the other hand, its computational cost is high, as well as its storage requirements.

4.1.2 Feature-based approach

Feature-based approaches to SLAM use easily identifiable attributes in the environment (landmarks) and build a metrically accurate map of these landmarks. Usually, this map is updated using an estimation-theoretic methodology, based on the Kalman filter. The pioneering work in this field was done by Smith and Cheeseman [76] and Durrant-Whyte [31]: they established a statistical basis for describing relationships between landmarks and manipulating geometric uncertainty. At the same time, Ayache and Faugeras [2] and Mortalier and Chatila [67] were applying Kalman filter-type algorithms in visual navigation of mobile robots. These two lines of research had much in common as it resulted in the key paper by Smith, Self and Cheeseman [77]. This paper showed that as a mobile robot moves through an unknown environment taking relative observations of landmarks, the estimates of these landmarks are all necessarily correlated with each other because of the common error in the estimate of the vehicle location (this has been recently proven by Dissanayake et al. [28] for the linear problem). Other related papers developed a number of aspects of the SLAM problem (see, e.g., [16, 54]). The main conclusion of this amount of work can be resumed in the two following statements:
4.1. APPROACHES TO CONCURRENT MAPPING AND LOCALIZATION

- Accounting for correlation between landmarks is important if filter consistency is to be maintained.

- A full Kalman-filter based SMAL solution requires that a state vector consisting of all states in the vehicle motion model and the position of all landmarks is maintained and updated.

Consequently, the optimal algorithm that retains all correlations, shows high computational requirements (the so-called map scaling problem): as the number $m$ of landmarks grows, the computations required at each step increase as $m^3$, and the required map storage increases as $m^2$ [28]. Moreover, the failure of simple strategies which ignore the correlations has been proven in simulation by Uhlmann et al. [85] and in experiments by Castellanos et al. [17].

Currently, there are two approaches to the problem: the first uses bounded approximations to the estimation of correlation between landmarks; the second method exploits the structure of the SLAM problem to transform the map building process into a computationally simpler estimation problem.

Bounded approximation methods use algorithms which make worst-case assumptions about correlation between two estimates. The covariance intersection method proposed by Uhlmann et al. [85] result in SLAM algorithms which have constant time update rules (independent of the number of landmarks in the map) and which are statistically consistent. However, this method may not provide an effective solution because the error bound can be too conservative.

Transformation methods usually end up in using multiple local maps, in order to reduce the dimension of the estimation problem, by computing multiple partial solutions in parallel. Betgé-Brezetz et al. [7] use multiple local maps to isolate odometry errors. Chong and Kleeman [21] use multiple local maps to address the issue of map scaling: when the vehicle enters a new map region, relocation of the vehicle between maps is accomplished by building a new map and matching it to a previously stored sub-map, using a technique developed exclusively for the sensor employed. Leonard and Feder [55] propose the use of a set of globally-referenced submaps, in order to avoid the matching of the submaps when moving from one region to the other. Thus, their Decoupled Stochastic Mapping method achieves constant-time performance (i.e., the complexity of the algorithm at each time instant is $O(1)$).

Finally, it must be noted that some feature-based approaches do away with the statistical formalism (and Kalman filtering), adopting other measures to describe uncertainty. Such strategies include iconic landmark matching [93] and global map registration [24].
4.1.3 Topological approach

Topological approaches to the SLAM problem do not attempt to build metrically accurate maps of the environment. Instead, they employ more qualitative knowledge of the relative location of landmarks and vehicle to build maps and guide motion. This general philosophy has been developed by a number of different groups in different ways: the most important contributions are those of Brooks [15], Levitt and Lawton [56] and Mataric [58]. The main advantage of qualitative navigation is the limited need for accurate models and affordable computational requirements. Moreover, they have a significant “anthropomorphic appeal”: it is believed that humans and animals do not produce accurate metric maps of the environment they work in, but rather they set up some topological description. This can be easily described using graph-like structures, where nodes correspond to significant places (landmarks), and arcs connecting these nodes correspond to action, or action-sequences that connect neighboring places.

Some effort has also been spent to merge together topological and feature-based approaches: Kurz [51] proposes to partition the free space into situation areas which are defined as regions wherein a specific situation can be recognized. Using dead reckoning, these areas are attached to a graph structure which also includes metric information. A modified extended Kalman filter is then developed to compensate the dead reckoning drift.

In general, all approaches to simultaneous localization and map building exhibit computational complexity rapidly increasing with the size of the area to be mapped (or with features density in the map). In addition, the problem of data association, i.e. resolving ambiguities in the source of measurements, is still an open issue. Also the problem of identifying suitable landmarks or “significant places” to be used for localization and map building needs to be solved.

4.2 SM SLAM Solution From Distance and Orientation Measurements

In this section, the concurrent mapping and localization problem is tackled in the set-membership framework. A recursive algorithm of reduced complexity (requiring at most $O(m)$ computation for each time step) is presented. This algorithm provides approximated set estimation for the exploring vehicle and each landmark: each of the aforementioned elements is guaranteed to lie into its own estimated set. The case of flat landscape and static landmarks is considered: in the following, the problem is cast in the framework
previously presented in Section 2.5.

4.2. SET-MEMBERSHIP SLAM SOLUTION

4.2.1 Models and problem formulation

Let us consider a mobile robot navigating in an unknown environment, for which a planar representation is appropriate. Let \( p(k) = [x(k) \ y(k)]' \in \mathbb{R}^2 \) be the coordinates of the vehicle at time \( k \). Under the same hypotheses introduced in Section 3.2, the robot dynamics can be described by the linear discrete-time model (3.1).

Since the robot has no a priori description of the environment, an efficient way to map it has to be pursued: here it is assumed that the robot will select \( m \) static landmarks, such that their position \( l_i(k) = [x_i(k) \ y_i(k)]' \in \mathbb{R}^2 \), \( i = 1, \ldots, m \) satisfies

\[
l_i(k + 1) = l_i(k), \tag{4.1}
\]

the initial condition \( l_i(0) \) being unknown. Nevertheless the following treatment can be readily extended to the case of moving landmarks (for example, by using equations similar to (3.1) to describe landmark dynamics). This will allow us to apply this technique also to the problem of unknown environment exploration performed by a team of robots, as it will be shown in Chapter 5.

As stated in the previous section, when tackling the SLAM problem, both landmarks and robot positions must be estimated. This requires state estimation of a dynamic system whose dimension can be very large, since it depends on the number of remarkable features present in the environment. Indeed, when \( m \) landmarks are considered, the state vector of the global system is given by

\[
\xi(k) = [p'(k) \ l_1'(k) \ldots \ l_m'(k)]'. \tag{4.2}
\]

From equations (3.1) and (4.1), the state update equation is

\[
\xi(k + 1) = \xi(k) + E_2 u(k) + E_2 G(k) w(k), \tag{4.3}
\]

where \( E_2 = [I_2 \ 0 \ldots 0]' \in \mathbb{R}^{2(m+1) \times 2} \).

The measurement model depends on the sensors used. For the solution of the SLAM problem, sensors able to measure both distances and orientations are usually employed: among them, there are millimeter-wave radars [28], sonars [55, 84], laser range finders [43] and stereovision systems [29, 52]. Under the hypothesis of a relative flat landscape, the measurements provided by the exteroceptive sensors concern the relative distance and orientation between the vehicle and each visible landmark. The following model will therefore be used:

\[
\begin{align*}
D_i(k) &= d_i(\xi(k)) + v_{d_i}(k), \\
T_i(k) &= \theta_i(\xi(k)) + v_{\theta_i}(k), \tag{4.4}
\end{align*}
\]
where $D_i(k)$ and $T_i(k)$ are the actual readings provided by the sensors at time $k$; $v_{d_i}(k)$ and $v_{\theta_i}(k)$ are measurement noise affecting, respectively, the distance and the orientation measurements, and

$$d_i(\xi(k)) = d(p(k), l_i(k)) \triangleq \sqrt{(x(k) - x_{l_i}(k))^2 + (y(k) - y_{l_i}(k))^2},$$

$$\theta_i(\xi(k)) = \theta(p(k), l_i(k)) \triangleq \arctan_2(y_{l_i}(k) - y(k), x_{l_i}(k) - x(k)).$$

(4.5)

Hence, in the second equations of (4.4)-(4.5) the direction of the straight line connecting the robot with the selected landmark, with respect to a fixed direction (here chosen as the positive $X$-axis on the reference system, see Fig. 4.1), is measured. This is possible if the robot is equipped with a sensor able to measure the robot orientation, such as a compass. Notice that the measurements (4.4)-(4.5) provide relative information: they can be thought of as the displacement of the landmarks with respect to the robot or vice versa. Each measurement gives two (noisy) nonlinear relations among four different components of the state (the robot coordinates and the selected feature coordinates).

In a set-theoretic framework, all disturbances $w(k)$, $v_{d_i}(k)$ and $v_{\theta_i}(k)$ are assumed to be unknown but bounded. Using weighted $\ell_\infty$ bounds, as introduced in Section 3.2, it results

$$\|w(k)\|_{\ell_\infty}^{e_w} \leq 1$$

(4.6)

$$\|v_{d_i}(k)\|_{\ell_\infty}^{e_{d_i}} \leq 1$$

(4.7)

$$\|v_{\theta_i}(k)\|_{\ell_\infty}^{e_{\theta_i}} \leq 1$$

(4.8)

where $e_w^k$, $e_{d_i}^k$ and $e_{\theta_i}^k$ are column vectors containing the time-varying weights. Assumptions (4.7) and (4.8) allow one to define, for each measurement (4.4), a set where the
system state is constrained to lie, whose expression is

\[ \mathcal{M}_i(k) = \left\{ \xi \in \mathbb{R}^{2(m+1)} : |D_i(k) - d_i(\xi(k))| \leq e_i^{x_i}(k) \text{ and } |T_i(k) - \theta_i(\xi(k))| \leq e_i^{\theta_i}(k) \right\}. \]  

(4.9)

Notice that each set \( \mathcal{M}_i \) is unbounded, since it defines two constraints on only four state variables (the other \( 2m - 2 \) being free). Nevertheless, the measurement set

\[ \mathcal{M}(k) = \bigcap_{i=1}^{m} \mathcal{M}_i(k) \]  

(4.10)

is limited, since each \( \mathcal{M}_i \) involves two different state elements, the \((2i+1)\)-th and \((2i+2)\)-th, respectively.

According to the dynamics model (4.3), the measurement model (4.4) and the UBB disturbances assumptions (4.6)-(4.8), the Simultaneous Localization And Map building problem can be written in the form presented in Section 2.5.1.

**Set-Membership Simultaneous Localization and Mapping Problem (SM-SLAM):** Let \( \Xi(0) \subset \mathbb{R}^{2(m+1)} \) be a set containing the initial state position of the vehicles and the landmarks \( \xi(0) \). Given the dynamics model (4.3) and the measurement equations (4.4), find at each time \( k = 1, 2, \ldots \) the feasible state set \( \Xi(k|k) \subset \mathbb{R}^{2(m+1)} \) containing all the state values \( \xi(k) \) that are compatible with the robot dynamics, the assumptions (4.6)-(4.8) on the disturbances, and the measurements collected up to time \( k \).

The solution to the above problem is given by recursion (3.10)-(3.12), with equation (3.11) suitably modified to account for model (4.3).

\[
\begin{align*}
\Xi(0|0) &= \Xi(0) \quad (4.11) \\
\Xi(k|k-1) &= \Xi(k-1|k-1) + E_2 u(k-1) + G(k-1) \text{Diag}[e_{k-1}] \mathcal{B}_\infty \quad (4.12) \\
\Xi(k|k) &= \Xi(k|k-1) \cap \mathcal{M}(k) \quad (4.13)
\end{align*}
\]

Notice that, in an unknown environment, there is no available information on the initial position of any element of the problem (vehicle and landmarks), so the initial set estimate (4.11) should be chosen as \( \mathbb{R}^{2(m+1)} \). Nonetheless, being all measurements relative, one is allowed to choose an arbitrary reference system. Hence, without loss of generality, the origin of the reference system can be set in the initial position of the robot, whose initial coordinates will consequently be known without uncertainty.

As already stated in Section 3.2, to devise computationally affordable algorithms, some approximations are introduced at different stages of the SLAM recursion (4.11)-(4.13):
1. Decomposition of the state vector $\xi(k)$ into subset of state variables and set-membership estimation of each subset.

2. Guaranteed approximations of the true feasible set through classes of simple regions.

In the following each approximation is discussed separately.

### 4.3 State Decomposition

State vector $\xi(k)$ is decomposed into two different subsets of variables: robot position $p(k)$ and landmarks position $L(k) = [l'_1(k) \ldots l'_m(k)]'$. Some remarkable differences between the two subsets can be easily pointed out: i) under the hypothesis of static landmarks, robot position is the only part of the state which undergoes a dynamic evolution; as a consequence, only uncertainty affecting robot coordinates is enlarged by the time update step; ii) robot displacement (and consequently position) is measured through proprioceptive sensors (e.g.: wheel encoders, compass), while information on landmarks position is collected only through distance and angle measurements performed from the robot standpoint.

The two state subsets are considered separately in the measurement update step (4.13). First, all the measurements are processed in order to get a set of robot positions that are compatible with all the information about landmarks, the time update performed at the previous time-instant, and the measurements. During this step landmark positions are not updated. In the second step, the same measurements are reprocessed to (possibly) tighten the uncertainty set of each landmark. The guiding idea is similar to the Baum-Welch algorithm [84], where map and robot location are alternatively updated by maximization in the likelihood space. Nevertheless, in this case, no probabilistic setting is adopted to describe the measurement process.

Notice that this approach allows one to simplify the measurement update step, but it also introduces an approximation because information about the relative position between robot and landmark is lost. Let us analyze how splitting the state vector modifies the feasible set recursive updating outlined in (4.11)-(4.13). Let $\Xi_p$ denote the feasible robot position set, $\Xi_{l_i}$ denote the feasible $i$-th landmark position set, and let $\Xi_p(0), \Xi_{l_i}(0)$ be the corresponding initial sets (they can be easily obtained by projecting $\Xi(0)$ onto the subspaces defined by $p$ and $l_i$). Initialization (4.11) clearly splits into

$$
\Xi_p(0|0) = \Xi_p(0), \\
\Xi_{l_i}(0|0) = \Xi_{l_i}(0).
$$

(4.14) 
(4.15)
4.3. STATE DECOMPOSITION

The time update equation (4.12) readily boils down to

\[ \Xi_p(k|k-1) = \Xi_p(k-1|k-1) + m(k-1) + G(k-1)\text{Diag}\{\epsilon_{k-1}^w\}B_\infty, \]
\[ \Xi_{li}(k|k-1) = \Xi_{li}(k-1|k-1); \quad i = 1, \ldots, m. \]

As said above, the measurement update (4.13) is performed in two steps. First, let us consider robot position. Since each distance and orientation measurement is relative, it is clear that landmark \( l_i \) "sees" the robot under the angle \( \theta(p(k), l_i(k)) + \pi \). It turns out that, using the measurement taken with respect to the \( i \)-th landmark, the position of the vehicle can be written as

\[
\begin{align*}
  x(k) &= x_{li}(k) - d_i(\xi(k)) \cos \theta_i(\xi(k)) \\
  y(k) &= y_{li}(k) - d_i(\xi(k)) \sin \theta_i(\xi(k))
\end{align*}
\]

(4.18)

Since \((x_{li}, y_{li}) \in \Xi_{li}(k)\), and noises affecting measurements \( D_i(k) \) and \( T_i(k) \) of \( d_i(\xi(k)) \) and \( \theta_i(\xi(k)) \) are bounded, as assumed in (4.7)-(4.8), it turns out that each of the measurements provides a feasible set \( \mathcal{C}_{R_i}(k) \) where the robot position must lie. Moreover, since the sources of uncertainty in (4.18) are independent, the set of feasible robot positions generated by the \( i \)-th measurement at time \( k \) is given by

\[ \mathcal{C}_{R_i}(k) = \Xi_{li}(k|k-1) + \mathcal{M}_{R_i}(k), \]

(4.19)

where

\[ \mathcal{M}_{R_i}(k) = \{ p \in \mathbb{R}^2 : |D_i(k) - d(p, 0)| \leq \epsilon_{\theta}(k) \text{ and } |T_i(k) - \theta(p, 0)| \leq \epsilon_{\theta}(k) \} \]

(4.20)

is the robot uncertainty set relative to the \( i \)-th measurement, for a landmark placed at the origin of the reference system.

Since this process can be repeated for every (measured) landmark, the robot position is constrained to lie in the set

\[ \mathcal{C}_R(k) = \bigcap_{i=1}^m \mathcal{C}_{R_i}(k). \]

(4.21)

Consequently, it turns out that measurement update for robot position can be performed as

\[ \Xi_p(k|k) = \Xi_p(k|k-1) \bigcap \mathcal{C}_R(k). \]

(4.22)

As a second step, we reconsider each measurement performed at time \( k \), this time trying to refine our knowledge on the landmarks. Note that, since each measurement provides a relationship between sets, and no statistical assumption is made, it is possible to use the same measurements to update the uncertainty sets relative to landmarks position. Using
an approach similar to the one presented for the robot position, one can state that, due
to the measurement at time $k$, the $i$-th landmark will lie in the set defined by

$$C_i(k) = \Xi_p(k|k) + M_i(k),$$

(4.23)

where

$$M_i(k) = \{ l \in \mathbb{R}^2 : |D_i(k) - d(0, l)| \leq \epsilon_i^u(k) \; \text{and} \; |T_i(k) - \theta(0, l)| \leq \epsilon_i^v(k) \}$$

(4.24)
is the $i$-th landmark uncertainty set, for a robot placed at the origin of the reference system.
Consequently, measurement update of the $i$-th landmark position can be performed as

$$\Xi_i(k|k) = \Xi_i(k|k-1) \bigcap C_i(k).$$

(4.25)

Notice that the sets $M_R_i(k)$ and $M_l_i(k)$ are sectors of corona; some examples of $M_l_i(k)$
are shown in Fig. 4.2.

We point out that the double-step measurement update provided by eqns. (4.19)-(4.25)
is more conservative than the original set-membership algorithm, where only the state
vectors satisfying eqns. (4.9)-(4.10) are deemed feasible for $\Xi(k|k)$. On the other hand, it
is clear that if the sets $\Xi_p$ and $\Xi_l$ are computed exactly, they represent the maximum
amount of information available on robot and landmark locations, when constraints on the relative position between them are neglected (i.e., when information on \( p(k) \) and \( l_i(k) \) is processed separately).

4.4 Set Approximation

Computing exactly sums and intersections of nonconvex region bounded by nonlinear curves, as required by (4.16)-(4.17) and (4.19)-(4.25) is still computationally too demanding. Hence, the approximation scheme outlined in Section 2.5.2 is applied to recursion (4.14)-(4.17), (4.22) and (4.25), thus leading to

\[
\begin{align*}
R_p(0|0) &= \overline{R}\{\Xi_p(0)\}, \\
R_{l_i}(0|0) &= \overline{R}\{\Xi_{l_i}(0)\}; \quad i = 1, \ldots, m (4.26) \\
R_p(k|k-1) &= \overline{R}\{R_p(k-1|k-1) + m(k-1) + G(k-1)\text{Diag}[\epsilon_{k-1}^\infty]B_\infty\}, (4.27) \\
R_{l_i}(k|k-1) &= R_{l_i}(k-1|k-1), (4.28) \\
R_p(k|k) &= \overline{R}\{R_p(k|k-1)\bigcap C_R(k)\}, (4.29) \\
R_{l_i}(k|k) &= \overline{R}\{R_{l_i}(k|k-1)\bigcap C_{l_i}(k)\}. (4.30)
\end{align*}
\]

Two different approximation algorithms, using respectively boxes and parallelotopes, can be devised. According to the above recursion, one must solve the three set approximation problems:

S1 Compute the minimum area box (parallelotope) containing the vector sum of a box (parallelotope) and a parallelotope (see (4.28));

S2 Compute the minimum area box (parallelotope) containing the intersection of a box (parallelotope) and the robot measurement set \( C_R(k) \) (see (4.30));

S3 Compute the minimum area box (parallelotope) containing the intersection of a box (parallelotope) and the \( i \)-th landmark measurement set \( C_{l_i}(k) \) (see (4.31)).

Problem S1 can be solved using Proposition 2.1 for boxes, and Theorem 2.5 (faster, suboptimal algorithm) or Theorem 3.1 (optimal solution) for parallelotopes. Suboptimal solutions to problems S2 and S3 will be discussed in the following.

4.4.1 Approximation through boxes

In order to simplify the computations, problems S2 and S3 are solved by performing first the approximation of the sets \( C_R(k) \) and \( C_{l_i}(k) \), and then that of the sets defined by the
intersections in eqns. (4.30) and (4.31). In other words, equations (4.30) and (4.31) are replaced by the following ones

\[
\mathcal{R}_p(k|k) = \mathcal{R} \left\{ \mathcal{R}_p(k|k-1) \cap \mathcal{R} \{ \mathcal{C}_R(k) \} \right\}, \quad (4.32)
\]

\[
\mathcal{R}_l_i(k|k) = \mathcal{R} \left\{ \mathcal{R}_l_i(k|k-1) \cap \mathcal{R} \{ \mathcal{C}_l_i(k) \} \right\}. \quad (4.33)
\]

When axis-aligned boxes are used as approximating sets, the above equations take on the simpler form

\[
\tilde{\mathcal{R}}_p(k|k) = \mathcal{R}_p(k|k-1) \cap \mathcal{B} \{ \mathcal{C}_R(k) \}, \quad (4.34)
\]

\[
\tilde{\mathcal{R}}_l_i(k|k) = \mathcal{R}_l_i(k|k-1) \cap \mathcal{B} \{ \mathcal{C}_l_i(k) \}. \quad (4.35)
\]

It is important to point out that sets \( \tilde{\mathcal{R}}_p(k|k) \) and \( \tilde{\mathcal{R}}_l_i(k|k) \) always contain sets \( \mathcal{R}_p(k|k) \) and \( \mathcal{R}_l_i(k|k) \) respectively. Moreover, we note that the conservativeness introduced by using (4.34)-(4.35) in place of (4.30)-(4.31) depends on the shape and the direction of the sets \( \mathcal{C}_R(k) \) and \( \mathcal{C}_l_i(k) \). If these sets do not stretch along the diagonals of the \((X,Y)\) plane, the approximation introduced is quite reasonable.

Due to the above simplification, the aforementioned approximation problems to be solved are the following:

B2’) Compute the minimum area box containing the robot measurement set \( \mathcal{C}_R(k) \) (eqn. (4.21));

B3’) Compute the minimum area box containing the \( i \)-th landmark measurement set \( \mathcal{C}_l_i(k) \) (eqn. (4.23)).

Notice that problem B3’ is just a special case of problem B2’, in which the set \( \mathcal{C} \) is generated by only one measurement. Computing the intersection of \( m \) nonconvex regions bounded by nonlinear curves is generally computationally untractable. To solve this problem online, a suboptimal solution based on recursive approximation is pursued: for every \( i \), the minimum area box containing \( \mathcal{C}_R \) is computed, and then all the approximating boxes are intersected. In other words, we replace \( \mathcal{B} \{ \mathcal{C}_R(k) \} \) in (4.34) with \( \cap_{i=1}^m \mathcal{B} \{ \mathcal{C}_R(k) \} \). Hence, problem B2’ boils down to solving \( m \) problems of the same type of problem B3’. Thus, we concentrate on problem B3’, and observe that the set \( \mathcal{C}_l_i(k) \) arises from the sum of the current landmark position estimate set and the sector of corona \( \mathcal{M}_l_i(k) \) (see eqn. (4.23)).

Then, we exploit the following results [27].

**Proposition 4.1** Let \( \mathcal{B}_p(k|k) \) be a given box containing \( \Xi_p(k|k) \) and \( \mathcal{M}_l_i(k) \) be defined as in (4.24). Then,

\[
\mathcal{C}_l_i(k) \subseteq \mathcal{B} \{ \mathcal{B}_p(k|k) + \mathcal{M}_l_i(k) \} = \mathcal{B}_p(k|k) + \mathcal{B} \{ \mathcal{M}_l_i(k) \}.
\]
4.4. SET APPROXIMATION

Figure 4.3: The points in $\mathcal{V}(\mathcal{M}_l)$ and $\mathcal{Z}(\mathcal{M}_l)$ for three typical uncertainty set $\mathcal{M}_l$ ($\circ$: points in $\mathcal{V}(\mathcal{M}_l)$; $\ast$: points in $\mathcal{Z}(\mathcal{M}_l)$).

Proof. It immediately follows from (4.23) and $\overline{\mathcal{B}}(\mathcal{B} + \mathcal{P}) = \mathcal{B} + \overline{\mathcal{B}}(\mathcal{P})$, which holds for any generic set $\mathcal{P}$ and box $\mathcal{B}$.

Let us denote by $\mathcal{V}(\mathcal{M}_l)$ the set of vertices of $\mathcal{M}_l$. Moreover denote by $\mathcal{Z}(\mathcal{M}_l)$ the set of points of the outer arc bounding $\mathcal{M}_l$ (the one corresponding to the maximum distance $D_i + \epsilon_i^{\text{gr}}$), where the tangent to the arc is parallel to one of the coordinate axes (see points denoted by asterisks in Fig. 4.3).

The following result allows one to compute the minimum area box containing $\mathcal{M}_l(k)$.

**Proposition 4.2** The minimum area box containing $\mathcal{M}_l(k)$ is the one containing the set $\mathcal{N} \mathcal{Z} = \mathcal{V}(\mathcal{M}_l(k)) + \mathcal{Z}(\mathcal{M}_l(k))$, i.e.,

$$
\overline{\mathcal{B}}(\mathcal{M}_l(k)) = \mathcal{B}(\bar{c}, \bar{b}),
$$

where

$$
\begin{align*}
\bar{c}_1 &= \frac{x + x_2}{2}, & \bar{c}_2 &= \frac{y + y_2}{2}, \\
\bar{b}_1 &= \frac{x - x_2}{2}, & \bar{b}_2 &= \frac{y - y_2}{2},
\end{align*}
$$

and

$$
\overline{x} = \max_{p \in \mathcal{N} \mathcal{Z}} x, \quad x = \min_{p \in \mathcal{N} \mathcal{Z}} x, \quad \overline{y} = \max_{p \in \mathcal{N} \mathcal{Z}} y, \quad y = \min_{p \in \mathcal{N} \mathcal{Z}} y.
$$
CHAPTER 4. SIMULTANEOUS LOCALIZATION AND MAP BUILDING

Proof. Let \( \text{co}\{P\} \) denote the convex hull of the set \( P \). We have \( \text{co}\{\mathcal{M}_i(k)\} = \overline{\text{co}\{\mathcal{M}_i(k)\}} \).

The boundary of \( \text{co}\{\mathcal{M}_i(k)\} \) is made of three segments and one arc of circumference (the one corresponding to the maximum distance \( D_i(k) + \varepsilon_i^d(k) \)). Hence, it is clear that the only points that must be considered besides the vertices \( \mathcal{V}(\mathcal{M}_i(k)) \) are those (if any) in \( \mathcal{Z}(\mathcal{M}_i(k)) \).

Propositions 4.1 and 4.2 provide the exact solution of problem B3', and hence an approximate solution of problem B2', in the sense explained above. It is then possible to compute a suboptimal solution of problem S3 by simply performing the intersection of \( \mathcal{R}_i(k|1) \) with \( \overline{\text{co}\{\mathcal{C}_i(k)\}} \), as suggested by equation (4.35). Finally, a suboptimal solution of problem S2 can be obtained by computing \( \mathcal{C}_i = \mathcal{B}(\mathcal{B}_i(k|1) \cup \mathcal{M}_i(k)) \), as suggested by equation (4.34) and the previous discussion.

The \( k \)-th step of the overall recursive approximating procedure is summarized in Table 4.1.

Let \( \mathcal{B}_i(k|1) \) and \( \mathcal{B}_i(k|1) \), \( i = 1, \ldots, m \) be given.

- \( \mathcal{B}_i(k|1) = \mathcal{B}(\mathcal{B}_i(k|1) + \mathcal{M}_i(k)) \) \( \text{[Prop. 2.1]} \);
- \( \mathcal{B}_i(k|1) = \mathcal{B}_i(k|1) \), \( i = 1, \ldots, m \);
- \( \mathcal{B}_i = \mathcal{B}(\mathcal{B}_i(k|1) \cup \mathcal{M}_i(k)) = \mathcal{B}_i(k|1) \) \( \text{[Prop. 4.1 and 4.2]} \);
- \( \mathcal{B}_i = \bigcap_{i=1}^m \mathcal{C}_i \);
- \( \mathcal{B}_i(k|1) = \mathcal{B}_i(k|1) \cap \mathcal{B}_i \);
- \( \mathcal{B}_i = \mathcal{B}(\mathcal{B}_i(k|1) + \mathcal{M}_i(k)) = \mathcal{B}_i(k|1) \) \( \text{[Prop. 4.1 and 4.2]} \);
- \( \mathcal{B}_i(k|1) = \mathcal{B}_i(k|1) \cap \mathcal{C}_i \), \( i = 1, \ldots, m \).

Table 4.1: The \( k \)-th step of the orthotope-based recursive SLAM algorithm.

4.4.2 Approximation through parallelotopes

When parallelotopes are used as approximating sets, it is possible to adopt the same strategy presented in the previous section. The approach to problems S2 and S3 is slightly
modified, introducing some further conservativeness, in order to reduce the computational burden of the algorithm.

In particular, we make the following approximations.

- Equations (4.30) and (4.31) are replaced with equations (4.32)-(4.33), i.e. outer approximation of the sets $C_R(k)$ and $C_{l_i}(k)$ is performed first, then intersections defined in eqns. (4.30) and (4.31) are computed. Notice that the intersection of two parallelogotopes is not a parallelogotope, and hence the following outer approximating regions must be computed

$$\tilde{R}_p(k|k) = \overline{P} \left\{ R_p(k|k-1) \bigcap \overline{P} \{ C_R(k) \} \right\},$$

(4.36)

$$\tilde{R}_{l_i}(k|k) = \overline{P} \left\{ R_{l_i}(k|k-1) \bigcap \overline{P} \{ C_{l_i}(k) \} \right\}.$$  

(4.37)

- $\overline{P} \{ C_R(k) \}$ in (4.36) is replaced with $\cap_{i=1}^n \overline{P} \{ C_{R_i}(k) \}$. The latter intersection is performed recursively, i.e. for every $i$ the minimum parallelotope containing the intersection of $\overline{P} \{ C_{R_i}(k) \}$ and the current approximating parallelotope is computed.

- The computation of $\overline{P} \{ C_{R_i}(k) \}$ in (4.19) is replaced by that of

$$\overline{P} \{ P_{l_i}(k|k-1) + \overline{P} \{ M_{R_i}(k) \} \}$$

where $P_{l_i}(k|k-1)$ is the parallelotope containing $\Xi_{l_i}(k|k-1)$, provided by the previous time update (see (4.29)). Notice that

$$C_{R_i}(k) \subset \overline{P} \{ P_{l_i}(k|k-1) + M_{R_i}(k) \} \subset \overline{P} \{ P_{l_i}(k|k-1) + \overline{P} \{ M_{R_i}(k) \} \}$$

and the rightmost term can be computed according to Theorem 3.1. Similarly, we replace the computation of $\overline{P} \{ C_{l_i}(k) \}$ in (4.23) with that of

$$\overline{P} \{ P_p(k|k) + \overline{P} \{ M_{L_i}(k) \} \}$$

where $P_p(k|k)$ is the parallelotope containing $\Xi_p(k|k)$, provided by the robot position measurement update (see (4.30)).

According to the above discussion, approximate solutions of problems S2 and S3 can be obtained by solving the following problems:

P2') Compute the minimum area parallelotope containing the intersection of two parallelogotopes (see eqns. (4.36)-(4.37));

P3') Compute the minimum area parallelotope containing the set $M_{R_i}(k)$ in (4.20) (and analogously, $M_{L_i}(k)$ in (4.24)).
The solution of problem P2’ can be obtained using Theorem 2.5 (faster, suboptimal algorithm) or Theorem 3.1 (optimal solution). Let us consider problem P3’. One has to compute the minimum parallelotope containing $M_l_i(k)$. The following result holds (the following procedure can be similarly applied to $M_R_i(k)$).

**Proposition 4.3** The minimum area parallelotope containing $M_l_i(k)$ is a rectangle given by

$$\overline{P}(M_l_i(k)) = P(\bar{T}, \bar{c})$$

where

$$\bar{c} = \begin{bmatrix} \frac{\bar{d}}{2} \cos T_i \\ \frac{\bar{d}}{2} \sin T_i \end{bmatrix}, \quad \bar{T} = \begin{bmatrix} \frac{\bar{d}}{2} \cos T_i \\ -\bar{d} \sin \epsilon_{i}^{on} \sin T_i \\ \bar{d} \sin \epsilon_{i}^{on} \cos T_i \end{bmatrix}$$

(4.38)

and

$$\bar{d} = D_i + \epsilon_{i}^{on}, \quad \underline{d} = (D_i - \epsilon_{i}^{on}) \cos \epsilon_{i}^{on}. \quad (4.39)$$

**Proof.** Let us denote by $V(M_l_i)$ the set of vertices of $M_l_i$ and by $A_l_i$ the point of $M_l_i$ corresponding to the maximum distance $D_i + \epsilon_{i}^{on}$ and the nominal orientation $T_i$, i.e. $A_l_i = ((D_i + \epsilon_{i}^{on}) \cos T_i, (D_i + \epsilon_{i}^{on}) \sin T_i)$ (see Fig. 4.4). Since $M_l_i$ is symmetric with respect to the line containing the origin and $A_l_i$, also the minimum area parallelotope containing $M_l_i$ is symmetric, and therefore it is a rectangle. Moreover, its edges will be respectively parallel and orthogonal to the symmetry axis of $M_l_i$, and tangent to $M_l_i$ in the set of points $V_M = V(M_l_i) \cup \{A_l_i\}$. Hence, it is easy to see that this rectangle can be written as $P(\bar{T}, \bar{c})$, where (4.38) holds with

$$\bar{d} = \max_{p \in V_M} \sqrt{x^2 + y^2 \cos(\arctan(y, x) - T_i)}, \quad \underline{d} = \min_{p \in V_M} \sqrt{x^2 + y^2 \cos(\arctan(y, x) - T_i)}.$$

Finally, (4.39) can be obtained by direct substitution, observing that the maximum is achieved in $A_l_i$, while the minimum is achieved in the two vertices of $V(M_l_i)$ corresponding to the minimum distance $D_i - \epsilon_{i}^{on}$. \hfill \blacksquare

The $k$-th step of the parallelotope-based recursive approximating procedure is summarized in Table 4.2.

### 4.4.3 Initialization and computational complexity

The proposed set-membership algorithm based on box (parallelotopic) approximations, must be initialized by selecting suitable sets $R_p(0|0)$, $R_l_i(0|0)$, to start recursion (4.26)-(4.31). These sets represent the available knowledge on robot and landmark positions at the beginning of the SLAM experiment. As observed in Section 4.3, the origin of the
4.4. SET APPROXIMATION

The proposed set-membership SLAM strategy can be easily implemented in real-time problems, due to its low computational complexity. In fact, the most demanding tasks are the computation of $\mathcal{R}_p(k|k)$ and of the $m$ sets $\mathcal{R}_l(k|k)$, in (4.30) and (4.31), respectively. It turns out that each problem, when tackled via the proposed recursive suboptimal solution, requires $O(m)$ operations, and also the memory requirements to store all the information about the map are proportional to the number of landmarks. Since each landmark is treated separately, the computational requirements of the algorithm depend on the number of perceived landmarks. This can consistently lower the computational burden of the algorithm. Moreover, when stereovision is used to perform measurements on the environment, the nearest landmarks provide the most accurate data [90]. This indeed allows one to devise an algorithm with constant-time performances, by processing only the $\tilde{m}$ “nearest” measurements (where $\tilde{m}$ is a fixed number selected a priori).

For all the above reasons, the proposed approach can handle also quite wide (or feature-
Let $\mathcal{P}_p(k|k-1)$ and $\mathcal{P}_l(k|k-1)$, $i = 1, \ldots, m$ be given.

- $\mathcal{P}_p(k|k-1) = \overline{\mathcal{P}} \{ \mathcal{P}_p(k|k-1) + m(k-1) + G_d(k-1) \text{Diag}[\epsilon_{k-1}^{\omega}] \mathcal{B}_\infty \}$ [see Theorem 3.1];

- $\mathcal{P}_l_i(k|k-1) = \mathcal{P}_l_i(k|k-1)$, $i = 1, \ldots, m$;

- $\mathcal{P}_{\mathcal{C}_R_i} = \overline{\mathcal{P}} \{ \mathcal{P}_l_i(k|k-1) + \overline{\mathcal{P}} \{ \mathcal{M}_{R_i}(k) \} \}$, $i = 1, \ldots, m$ [Prop. 4.3 and Theorem 3.1];

- $\mathcal{P}_{\mathcal{C}_R} = \mathcal{P}_{\mathcal{C}_R_1}$;
  for $i = 2, \ldots, m$
  $\mathcal{P}_{\mathcal{C}_R} = \overline{\mathcal{P}} \{ \mathcal{P}_{\mathcal{C}_R} \cap \mathcal{P}_{\mathcal{C}_R_i} \}$ [see [19]];
  end;

- $\mathcal{P}_p(k|k) = \overline{\mathcal{P}} \{ \mathcal{P}_p(k|k-1) \cap \mathcal{P}_{\mathcal{C}_R} \}$ [19];

- $\mathcal{P}_{\mathcal{C}_l_i} = \overline{\mathcal{P}} \{ \mathcal{P}_p(k|k) + \overline{\mathcal{P}} \{ \mathcal{M}_{l_i}(k) \} \}$, $i = 1, \ldots, m$ [Prop. 4.3 and Theorem 3.1];

- $\mathcal{P}_l_i(k|k) = \overline{\mathcal{P}} \{ \mathcal{P}_l_i(k|k-1) \cap \mathcal{P}_{\mathcal{C}_l_i} \}$, $i = 1, \ldots, m$ [19].

| Table 4.2: The $k$-th step of the paralleloptope-based recursive SLAM algorithm. |

dense) environments. Notice that the above considerations hold for both orthotope-based and paralleloptope-based approximations. Nonetheless, the paralleloptopic approach is more computationally demanding, due to the more complex structures it has to handle (see for example the difference between the intersection of orthotopes and of paralleloptopes).

In Section 7.2, the above estimation techniques are tested on a group of experiments, in order to evaluate their performances.
Chapter 5

Cooperative Localization and Map Building

In recent years, the attention of the robotics research community has been attracted by the study of robot colonies [1, 3]. As a matter of fact, many robotic applications require that robots work in collaboration in order to perform a certain task [36]. Nevertheless, most existing localization approaches refer to the case of a single robot. Even when a group of $N$ robots is considered, the group localization problem is usually tackled by solving independently $N$ pose estimation problems. Each robot estimates its position based on its individual experience (proprioceptive and exteroceptive sensor measurements). This is a relatively simple approach, that may provide reasonable results in many environments. However, if robots can detect each other and share their information, the result of the localization and mapping algorithm can be drastically improved, for several reasons. Each robot plays the role of a (possibly moving) landmark for all other robots. Moreover, at each time instant, the same environment feature can be perceived by more than one robot. If the robots share mapping information, this can lead to a more accurate, faster converging global map. In the following, the set-membership approach to the SLAM problem presented in the previous chapter will be extended to the multi-robot case.

The chapter is organized as follows: Section 5.1 presents a brief review of the available works on the Cooperative Localization and Mapping (CLAM) problem. Section 5.2 presents the extension of the algorithms introduced in Chapter 4 to multi-robot systems. Finally, Section 5.3 considers the problem of optimal information fusion from several set-valued maps of the same environment.
5.1 Approaches to Cooperative Localization and Mapping

As already stated, even in the case of colonies of cooperative robots, there are few works tackling the issue of cooperative localization.

Concerning the localization problem, Kato et al. [48] present a static relative localization strategy based on angular measurements between all the robots. Relative positions are computed by applying geometrical constraints (the sum of the angles between three neighboring robots must be 180 degrees) to a least-squares estimation method. Kurazume et al. [50] propose a cooperative algorithm for relocalization in planned motion execution. They divide the robots into two teams. At each time step, one team is in motion, while the other remains stationary and acts as a group of landmarks. In the next phase the role of the teams are reversed and this process continues until both teams reach the target. Rekleitis et al. [72] propose an algorithm dealing with the problem of exploration of an unknown environment using two mobile robots. Also in this approach, only one robot is allowed to move at any point in time, while the other robot observes the moving one. The stationary robot tracks the position of the moving robot, thus providing position estimates more accurate than those provided by pure dead reckoning. A similar approach, based on trilateration, extended Kalman filtering and omnidirectional sonar sensors is presented by Grabowski et al. [42]. They consider a team of homogeneous robot: at each time step three of them remain stationary, while the others explore the area, using the still robots as beacon. Since the performance of the triangulation algorithm depends on the relative location of the landmarks, particular care is dedicated to the planning of the movement sequence.

Roumeliotis and Bekey [73] propose a centralized Kalman filter for the localization of \( N \) robots: the group is viewed as a single system, and the proposed algorithm fuses information provided by the sensors distributed on the individual robots. The centralized Kalman filter is then decomposed in \( N \) modified filters, each running on a separate robot.

Fox et al. [38] extend the Markov localization techniques to deal with the multi-robot localization issue. Each robot develops its own belief function, but when two robots detect each other, new probabilistic constraints are added, and the two belief functions are tied together. It is also shown, through simulation experiments, how the collective approach to localization can dramatically improve the quality of the results, also for heterogeneous robots: this allows for reduction of hardware costs, since it is no longer necessary to equip each robot with a sensor suit needed for global localization.

Concerning cooperative map building, solutions have been proposed [25, 47] for generating an augmented topological map through fusion of maps produced by a team of robots.
equipped with different sensors. In this situation, the robots localization issue is not considered, since these algorithms process the data off-line.

Few authors have investigated the problem of cooperative localization and mapping (i.e., employing multiple robots to solve the SLAM problem): Yamauchi [91] focuses on the problem of exploration (i.e., choosing how the robot moves to completely map the environment). He introduces the idea of frontier (i.e., the boundary between known and unknown regions): then he chooses how the single robot has to move to increase the speed with which the map is built. Consequently, the cooperation factor is quite limited in his work. Barth and Ishiguro [6] describe a multi-robot system in which each robot is equipped with a 360 degree panoramic camera. The robots are able to determine their relative position and can use this data to combine each robot local map to form a global map.

### 5.2 Set-Membership CLAM

In this section, the multi-robot cooperative localization and map building (CLAM) problem will be tackled in the set-theoretic estimation framework. The algorithm presented in the previous chapter will be extended to the case of $n$ robots, whose position at time $k$ will be denoted by $p_1(k), \ldots, p_n(k)$, moving in an environment containing $m$ static landmarks $l_1, \ldots, l_m$. All the hypotheses on robot dynamics and sensor measurements are those reported in the previous chapters.

When dealing with $n$ robots and $m$ landmarks, the state vector to be estimated becomes

$$\xi(k) = [p_1'(k) \ldots p_n'(k) l'_1 \ldots l'_m]'$$

Assuming robot dynamics $p_i(k+1) = p_i(k) + u_i(k) + G_i(k)w_i(k)$ for each agent, the state update equation turns out to be

$$\xi(k+1) = \xi(k) + E_nu(k) + E_nG(k)w(k)$$

where $E_n = [I_{2n} \ 0] \in \mathbb{R}^{2(n+m) \times 2n}$, $G(k)$ is the block diagonal matrix of blocks $G_i(k)$, $i = 1, \ldots, n$ (i.e., $G(k) = \text{Diag}[G_1(k), \ldots, G_n(k)]$), $u(k) = [u_1'(k) \ldots u_n'(k)]'$ and $w(k) = [w_1'(k) \ldots w_n'(k)]'$.

Since each agent performs measurements with respect to all visible features in the environment (the other robots and the static landmarks), the maximum amount of information provided by exteroceptive sensors is

$$D_i^j(k) = d_i^j(\xi(k)) + v_{d_i^j}(k)$$

$$T_i^j(k) = \theta_i^j(\xi(k)) + v_{\theta_i^j}(k)$$
for \( j = 1, \ldots, n \), \( i = 1, \ldots, n + m \), \( i \neq j \), where \( D_i^j(k) \) and \( T_i^j(k) \) are the actual readings of the \( j \)-th robot sensors towards the \( i \)-th feature in the environment (with the notation \( d_i^j(\xi) = d(p_j, p_i), i = 1, \ldots, n \) and \( d_{i+n}^j(\xi) = d(p_j, l_i), i = 1, \ldots, m \), where \( d(a, b) \) is defined in (4.5). The same notation is adopted for \( \theta_i^j \).

In the following we will denote by \( U_i^j(k) \) the pair of measurements \( D_i^j(k), T_i^j(k) \). Similarly to what has been done in the single-robot case, unmodeled dynamics errors and measurement noises are assumed to be unknown-but-bounded, as in (4.6)-(4.8). Then, the measurement set \( \mathcal{M}_i^j(k) \) associated to \( U_i^j(k) \), can be defined as in (4.9) and the exact feasible state set is obtained (at least in principle) through a recursive procedure similar to (4.11)-(4.13). Clearly, the exact computation of sets \( \Xi \) cannot be tackled in practice. Therefore, an efficient set approximation strategy exploiting the specific structure of the SM-CLAM problem is needed. The main idea is to decompose the approximation of the feasible set \( \Xi \) into \( n + m \) approximations of 2D feasible subsets for each feature in the environment.

Let \( R_{p_j} \) and \( R_{l_h} \) denote outer approximations of the feasible position sets of robot \( p_j \) and landmark \( l_h \) respectively, chosen in the set class \( \mathcal{R} \). The sets defined below will be useful in the following treatment:

- \( \mathcal{M}_{p_i}(U_j^i, R_{p_j}) \): set of positions of robot \( p_i \), compatible with uncertainty set \( R_{p_j} \) and measurements \( U_j^i \);
- \( \mathcal{M}_{p_i}(U_{h+n}^j, R_{l_h}) \): set of positions of robot \( p_i \), compatible with uncertainty set \( R_{l_h} \) and measurements \( U_{h+m}^i \);
- \( \mathcal{M}_{p_i}(U_i^j, R_{p_j}) \): set of positions of robot \( p_i \), compatible with uncertainty set \( R_{p_j} \) and measurements \( U_i^j \);
- \( \mathcal{M}_{l_h}(U_{h+n}^j, R_{p_j}) \): set of positions of landmark \( l_h \), compatible with uncertainty set \( R_{p_j} \) and measurements \( U_{h+m}^j \).

Notice that the first two sets show how the uncertainty on the measured features affects the uncertainty of the robot performing those measurements. On the contrary, the third and fourth sets state how the uncertainty on the robot performing the measurements affects that of the measured features. As it has been previously shown in Section 4.4, for the pair of measurements \( U_i^j \) defined in (5.3)-(5.4), the above sets are the sum of sectors of circular coronas and the sets \( R_{p_j} \) or \( R_{l_h} \) (see, e.g. Fig. 5.1).

In particular, if axis-aligned boxes are used as approximating regions it is easy to compute the minimum area box containing one of the above sets, using Proposition 4.1.
5.2. SET-MEMBERSHIP CLAM

Let $\mathcal{R}\{P\}$ denote, as usual, the minimum area set in the class $\mathcal{R}$, containing the set $P$. Hence, the following suboptimal recursive approximation strategy can be applied at each time $k$.

**Step 0.** Let $\mathcal{R}_{p_i}(k-1|k-1), \mathcal{R}_{l_j}(k-1|k-1)$ be 2D sets containing the projection of $\Xi(k-1|k-1)$ on the subspaces spanned respectively by $p_i$ and $l_j$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.

**Step 1.** Time update of robot uncertainty sets.

For $i = 1, \ldots, n$:

$$\mathcal{R}_{p_i}(k|k-1) = \mathcal{R}\left\{\mathcal{R}_{p_i}(k-1|k-1) + u_i(k-1) + G_i(k-1)\text{Diag}[\epsilon_{k-1}^{p_i}]\mathcal{B}_{\infty}\right\}.$$

**Step 2.** Robot measurement update (based on measurements performed by the same robot).

For $i = 1, \ldots, n$:

- let

$$A_{1,i} = \bigcap_{j=1}^{n} \mathcal{R}\left\{\mathcal{M}_{p_i}(U_j(k), \mathcal{R}_{p_j}(k|k-1))\right\},$$

$$A_{2} = \bigcap_{h=1}^{m} \mathcal{R}\left\{\mathcal{M}_{p_i}(U_{h+n}^i(k), \mathcal{R}_{l_h}(k-1|k-1))\right\};$$

- compute

$$\hat{\mathcal{R}}_{p_i}(k|k) = \mathcal{R}\left\{\mathcal{R}_{p_i}(k|k-1) \cap A_{1,i} \cap A_{2}\right\}.$$
The set $A_{1,i}$ contains the feasible set for agent $p_i$ based on other robots uncertainty and measurements performed on them by vehicle $p_i$. Set $A_2$ contains the feasible set for robot $p_i$ based on landmarks uncertainty and measurements performed by the same robot.

**Step 3.** Robot measurement update (based on measurements performed by other robots).

For $i = 1, \ldots, n$:
- let $A_{3,i} = \bigcap_{\substack{j=1\ j\neq i}}^{n} \mathcal{R}\left\{ M_{p_i}(U_{i}^j(k), \hat{R}_{p_j}(k|k)) \right\}$;
- compute $\mathcal{R}_{p_i}(k|k) = \mathcal{R}\left\{ \hat{R}_{p_i}(k|k) \cap A_{3,i} \right\}$.

Set $A_{3,i}$ contains the feasible set for robot $p_i$ depending on other agents current uncertainty (updated in step 2) and their measurements performed on robot $p_i$.

**Step 4.** Landmark measurement update.

For $h = 1, \ldots, m$:
- let $A_4 = \bigcap_{j=1}^{n} \mathcal{R}\left\{ M_{l_h}(U_{h+n}^j(k), \mathcal{R}_{p_j}(k|k)) \right\}$;
- compute $\mathcal{R}_{l_h}(k|k) = \mathcal{R}\left\{ \mathcal{R}_{l_h}(k-1|k-1) \cap A_4 \right\}$.

Set $A_4$ contains the feasible set for landmark $l_h$ based on robots current uncertainty (updated in step 3) and measurements performed by all robots on the same landmark $l_h$.

Clearly, in the above strategy there is a redundancy, due to the fact that the same measurements are processed two times: first in step 2, then in steps 3 or 4. This redundancy is useful because in each step several set approximations are involved and hence measurement reprocessing leads to approximation refinement.

Notice that when axis-aligned boxes are considered as approximating regions $\mathcal{R}$, the intersections in the sets $A$ are simple 2D box intersections, and also minimum box approximations $\mathcal{R}\{\cdot\}$ are very easy to compute.

It is worth remarking that, no matter which class of approximating sets $\mathcal{R}$ is chosen, the above strategy is conceived so that a guaranteed outer approximation of the true feasible state set $\Xi(k|k)$ is obtained at each $k$. 
5.3. MAP FUSION IN THE SET-MEMBERSHIP FRAMEWORK

In fact, as it has been pointed out in Section 3.2, it turns out that $\Xi(k|k) \subset R_{p_1}(k|k) \otimes \ldots \otimes R_{p_m}(k|k) \otimes R_{L_1}(k|k) \otimes \ldots \otimes R_{L_n}(k|k)$.

The only problem concerning the proposed algorithm is that of initializing the recursive procedure. This is a crucial step, because a rough initialization would cause the whole procedure to fail. Indeed, if the initial sets $R_{p_i}(0|0)$, $R_{L_j}(0|0)$ are too large (for example, including the whole area in which the robots are moving) steps 2-4 of the procedure will typically provide no uncertainty reduction. This is a consequence of the simplification introduced by state decomposition, in a context where only relative measurements are available.

A natural way to overcome this problem is to exploit the information that each robot is able to acquire by itself in the initialization step. In fact, every agent can solve a single-robot SM-SLAM problem in a preliminary phase and then provide the estimated map (in its own reference system) to a central unit. The unit has to merge all the maps to obtain a unique map in an “absolute” reference system, and then send it to all the agents that will use it to initialize the recursive procedure previously outlined.

The next section will address the problem of merging set-valued maps worked out by different agents.

5.3 Map Fusion in the Set-Membership Framework

As stated in Section 4.4.3, in the single robot SLAM problem, the initial position of the exploring agent can be arbitrarily fixed (in the robot own reference system) and all the environment features are then estimated with respect to that initial position. When operating with multiple agents, the starting position of each robot is not known and cannot be arbitrarily fixed.

Each robot is able to produce a self-centered initial map of the environment, exploiting the SM-SLAM strategy presented in Chapter 4. Clearly, all these descriptions are valid, suboptimal representations of the environment. The quality of the maps will generally vary from robot to robot. While it is possible to choose the “more precise” map as a global map for all the robots, this choice does not use all the available information. Exploiting the “feasibility” property (i.e., the fact that each set-valued map contains the correct environment representation), it is possible to obtain a global refined map, by finding a description which satisfies all the constraints of each map.

It is assumed for simplicity that all robots are equipped with a compass and consider the same reference direction for angle measurements. Therefore, only relative translation among the maps is unknown. If axis-aligned boxes are chosen as approximating sets,
it is possible to decouple the 2D map fusion into two different 1D map fusions, thus leading to further simplified computations. Therefore, let us consider $n$ robots and $m$ landmarks spread on a line. Each robot produces his own 1D self-centered map, containing $n + m$ intervals. Let us suppose that intervals in each 1D map are correctly related to the corresponding features.

Let us call $C_j^i$ the center of the set corresponding to the $j$-th feature, in the map provided by the $i$-th agent. Moreover, let us call $C_{j-}^i$ and $C_{j+}^i$ the minimum and maximum value of the interval with center $C_j^i$.

In terms of relative positions between sets, each ordered map $i$ provides a certain number of constraints, such as

$$m_j - M_k \geq C_{j-}^i - C_{k+}^i,$$

$$M_j - m_k \leq C_{j+}^i - C_{k-}^i.$$  

for $j > k$, $k \geq 1$, $j \leq n + m$, where $m_j$ and $M_j$ are respectively the minimum and maximum admissible value for the position of the $j$-th feature of the map. Note that (5.5) provides $(n + m)(n + m - 1)$ inequalities in $2(n + m)$ unknowns for each map $i$: as a general consequence, for $n + m \geq 3$, each map provides an overconstrained system, which results to be always feasible. If all the inequalities (5.5) provided by all maps are considered, one has a system of $n(n + m)(n + m - 1)$ inequalities. Since the SM approach guarantees that each set contains the true position of the corresponding feature, the latter will satisfy all the constraints in (5.5) for each $i$. To get from (5.5) the tighter constraints, one can choose as “minimum” distance between set $j$ and $k$ the maximum of all minimum distances (and similarly as “maximum” distance, the minimum of all maximum distances). This leads to

$$m_j - M_k = \max_{i=1,...,n}(C_{j-}^i - C_{k+}^i),$$

$$M_j - m_k = \min_{i=1,...,n}(C_{j+}^i - C_{k-}^i).$$  

(5.6)

This time, however, the constraints provided by (5.6) will not generally be compatible, thus giving an overconstrained, unsolvable linear system. In order to get a solution $(m_k, M_k)$, $k = 1, \ldots, n + m$, it is possible to relax system (5.6) into a Linear Programming problem, where each of the equality constraints is transformed into a suitable inequality, while the objective function can be chosen so that the global map uncertainty is minimized. This leads to the following LP problem

$$\min_{m_k, M_k} \sum_{k=1,...,n+m} M_k - m_k$$

for $k = 1, \ldots, m + n$.  

(5.7)
subject to the constraints

\[
\begin{align*}
    m_j - M_k & \leq \max_{i=1,\ldots,n} (C_{j-}^i - C_{k+}^i), \quad j > k; \\
    M_j - m_k & \geq \min_{i=1,\ldots,n} (C_{j+}^i - C_{k-}^i), \quad j > k; \\
    m_k - M_k & \leq 0, \quad k = 1, \ldots, n + m; \\
    m_1 & = 0.
\end{align*}
\]

The last equality constraint fixes a point of the map as a common reference for all robots. By solving problem (5.7), one obtains the maximum and minimum value for the uncertainty interval of each feature, and thus the map (central estimate plus interval width) merging all the available information. Notice that the solution of problem (5.7) provides the minimum uncertainty map (in the sense defined by the chosen cost function) satisfying all the compatible constraints of equations (5.6), the other constraints being relaxed of the minimum quantity needed to guarantee solvability of the system.

As stated before, the use of boxes permits to transform the generic 2D problem into two separate 1D problems, by considering separately the \(x\) and \(y\) coordinates. This means that 2D map fusion boils down to the solution of just two LP problems.

Simulation result for both a static and dynamic setting are reported in Section 7.3.
CHAPTER 5. COOPERATIVE LOCALIZATION AND MAP BUILDING
Chapter 6

Estimation of Time to Contact

In this chapter, another important problem in robot visual navigation is tackled in the set-membership framework: the determination of the time to contact (or time to collision), i.e. the computation of the time needed for the observer to reach a fixed object, in the hypothesis that the relative velocity along the optical axis is kept constant. In mobile robotics, the recovery of this parameter is of paramount importance in collision avoidance and in braking. The chapter is structured as follows: Section 6.1 provides a brief introduction to the approaches to the time-to-contact estimation problem available in the literature. Section 6.2 shows how the problem can be cast as a linear, recursive state estimation problem. Finally, Section 6.4 provides the theoretical development that allows one to evaluate guaranteed upper and lower bounds for the true time to collision.

6.1 Approaches to Time-to-Contact Estimation

The work of Nelson and Aloimonos [68] is one of the first papers in which vision-based spatio-temporal techniques enable the robotic device to avoid collisions by computing the time to contact (see [22] for a comprehensive bibliography on the topic). Within the world of dynamic vision, this specific problem appears as a typical “goal oriented” problem, in the sense that its solution does not require the complete 3D scene reconstruction, but it can be accomplished in an efficient way by using only a partial solution of the general structure-from-motion problem. Two main approaches have been proposed in the literature. The first one is based on differential invariants of the image velocity field [22]; the second one relies on the estimation of the optical flow [79, 59].

Differential invariants of the image velocity field (curl, divergence and shear) are used to characterize the changes in the shape of objects in the scene due to the relative motion
between the observer and the scene. Under the hypothesis of constant velocity along the optical axis, the time to contact turns out to be a function of the area enclosed by the object contour and its time derivative. One advantage of this method consists in the fact that the evaluation of the time to contact is performed without tracking point features in the image, i.e. without estimating the full image velocity field, which is an ill conditioned problem. In fact, since the estimates are based on surface integrals along the contours, the area based method is weakly sensitive to noise measurements. The main drawback of this technique consists in its sensitivity to partial occlusions of the object.

The second approach is based on the estimation of the optical flow [79, 59]. As such, this method shows good ability in following fast changes in the time to contact, but since it involves estimation of derivatives of the optical flow, it is quite sensitive to measurement noise.

In the following, the differential invariant approach will be adopted and a set-theoretic estimation technique will be employed to compute guaranteed bounds on the time to contact.

6.2 Contour Tracking

Linear parameterization of image contours is largely used in computer vision. B–spline interpolation shows attractive features for fitting both open and closed contours (see e.g. [9], [10]). In fact, it is well known that this parameterization leads to numerically well behaved solutions and to efficient algorithms.

A closed B–spline curve of degree $h$ is defined as

$$S(u) = \sum_{i=0}^{m-1} V_i B_{ih}(u), \quad u \in [u_{\text{min}}, u_{\text{max}}],$$

where $B_{ih}(u)$ are the B–spline base functions of degree $h$ and the coefficients $V_i = (X_i, Y_i)$ are the control points of the curve. The base functions satisfy a number of constraints at the knots of the curve, ensuring a given level of regularity of the interpolating curve [5].

The contour tracking problem consists in computing an estimate $\hat{S}(u; t)$ of the object contour as a function of the control points $V_i(t)$, evolving dynamically with time. The problem is frequently approached through Kalman filtering theory. This is done by constructing a state space model of the rigid object contour motion. State variables are defined as the coordinates $X_i(t), Y_i(t)$ of the control points and their respective time derivatives

$$X = [X_0, \ldots, X_{m-1}]', \quad Y = [Y_0, \ldots, Y_{m-1}]',$$

$$\dot{X} = [\dot{X}_0, \ldots, \dot{X}_{m-1}]', \quad \dot{Y} = [\dot{Y}_0, \ldots, \dot{Y}_{m-1}]'.$$  \hspace{1cm} (6.1)
A typical example of the contour dynamics equations is given by the random walk model

\[
\frac{d}{dt} \begin{bmatrix} X \\ \dot{X} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w_x ,
\]

(6.2)

where \( w \) represents the process noise accounting for randomly varying forces acting on the object. An equation similar to (6.2) holds for the state variables \( Y \) and \( \dot{Y} \). Equation (6.2) is commonly used when no a priori information on the contour dynamics is available (like, for example, in dead reckoning problems).

The visual measurement process consists in collecting measurements of the contour position, by searching along normal rays to the estimated curve and within a search window, whose size is function of the positional variance of the current estimate. Contour measurements can be expressed in terms of the state vectors as

\[
z_x(u,t) = \begin{bmatrix} B(u) & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \end{bmatrix} + v_x(u,t) ,
\]

(6.3)

where \( v \) is the measurement noise and \( B(u) \) is a vector whose components are the B–spline base functions \( B_i(u), i = 0, \ldots, m-1 \). A similar equation holds for measurements \( z_y(u,t) \) of the second coordinate of the control points.

### 6.2.1 Planar affine transformation of contours

If the image shape is complex, the parameterization through B–splines requires a large number of control points. As a consequence, the dimension of the state space vector built by grouping control points in (6.1) dramatically increases, leading image tracking algorithms to instability [9].

For many purposes, like for example the computation of the time to contact, the objects to be tracked are usually planar rigid bodies, as for instance the area of the rear windscreen of a car in automatic parking applications. In such cases, the contour parameterization can be simplified. In fact, it is known that the projection onto the image plane of a planar rigid object, moving with respect to the camera, can be approximated by just six affine degrees of freedom, under the hypothesis of weak perspective, i.e. if the depth of the object is small with respect to its distance from the camera, or when the planar object is orthogonal to the optical axis (a quite common situation in time-to-contact evaluation, see [86, 11] for more details). The six degrees of freedom can be represented by a vector \( Q \) defined as

\[
Q = \begin{bmatrix} t_x & t_y & sr_{11} & sr_{22} & sr_{21} & sr_{12} \end{bmatrix}'
\]

(6.4)
where $t_x$ ($t_y$) represents the coefficients of the contour translation in the image plane along
the $x$ ($y$) direction, $s$ denotes the scaling factor and $r_{ij}$ is the $(i, j)$ entry of the rotation
matrix $R$ ($\det(R) = 1$).

The $Q$–parameterization describes the contour as a function of the template [34], i.e., the
initial reference estimated contour denoted by control points $(\hat{X}, \hat{Y})$. The relationship
between the $Q$ vector and the control points of a single captured image is given by

$$
\begin{bmatrix}
X \\
Y
\end{bmatrix} = W Q; \quad \text{(6.5)}
$$

where

$$W =
\begin{bmatrix}
c_x 1 & 0 & \hat{X} & 0 & 0 \\
0 & c_y 1 & 0 & \hat{Y} & \hat{X} & 0
\end{bmatrix}, \quad \text{(6.6)}
$$

$\mathbf{1} = [1 \ 1 \ldots \ 1]'$, and $(c_x, c_y)$ is a reference point in the image (usually, the rotation center
for the planar rigid motion of the image). Clearly, the initial contour is represented by the
vector $Q_0 = [0 \ 0 \ 1 \ 0 \ 0]'$. From (6.5) it is easy to check that

$$Q = M \begin{bmatrix}
X \\
Y
\end{bmatrix}, \quad \text{(6.7)}
$$

where $M$ is the $H$–weighted pseudoinverse, $M = (W'HW)^{-1}W'H$. Matrix $H$ is a metric
matrix arising from the normal equations for the problem of least–squares approximation

The 12–dimensional vector of state variables is here defined as $[Q' \ 0]'$, $\dot{Q}$ being the time
derivative of $Q$

$$\dot{Q} = M \begin{bmatrix}
\dot{X} \\
\dot{Y}
\end{bmatrix}. \quad \text{(6.8)}
$$

The contour dynamics in the affine state space is easily obtained from the corresponding
equations in the control points state space. For example, the random walk model (6.2)
becomes

$$
\frac{d}{dt}
\begin{bmatrix}
Q \\
\dot{Q}
\end{bmatrix} = \begin{bmatrix}
0 & I \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
Q \\
\dot{Q}
\end{bmatrix} + \begin{bmatrix}
0 \\
I
\end{bmatrix} M \begin{bmatrix}
w_x \\
w_y
\end{bmatrix}. \quad \text{(6.8)}
$$

As regards the visual measurements process, equation (6.3) and its $y$ counterpart turn into

$$
\begin{bmatrix}
z_x(u, t) \\
z_y(u, t)
\end{bmatrix} = \begin{bmatrix}
B(u) & 0 \\
0 & B(u)
\end{bmatrix}
W \begin{bmatrix}
0 \\
0
\end{bmatrix}
\begin{bmatrix}
Q \\
\dot{Q}
\end{bmatrix} + \begin{bmatrix}
v_x(u, t) \\
v_y(u, t)
\end{bmatrix}. \quad \text{(6.9)}
$$
6.2. CONTOUR TRACKING

6.2.2 Discrete time dynamics of image contour

In order to derive a tracking algorithm for the image contour, the discrete-time version of the contour dynamics and the measurement equation should be considered. For both the control points parameterization and the affine transformation parameterization, the discrete-time equations assume the following general form

\[
\begin{align*}
\xi(k+1) &= F(k)\xi(k) + G(k)w(k), \\
y(k) &= H\xi(k) + v(k)
\end{align*}
\]  

where \( \xi(k) \) represents the state vector at time \( k \) and \( w(k) \) (\( v(k) \)) is the process (measurement) noise.

When the control points parameterization approach is used, the dynamics of the \( X \) and \( Y \) coordinates can be analyzed separately. For the \( X \)-coordinate of the image contour, the state vector at time \( k \) is \( \xi(k) = [X_k^0 \ X_k^1]^T \), while \( w(k) \) is the process noise at time \( k \).

The set of visual measurements available at time \( k \), \( y(k) = [z_x(u_1,k) \ ... \ z_x(u_l,k)]^T \), is a vector containing \( l \) measurements of the contour position, \( v(k) = [v_x(u_1,k) \ ... \ v_x(u_l,k)]^T \) is the measurement noise vector, and the output matrix \( H \) in (6.10) can be written as

\[
H = \begin{bmatrix} \Pi & 0 \end{bmatrix}
\]

being \( \Pi \) a matrix whose components \( \Pi_{ij} \) are the B-spline base functions \( B_{jh}(u_i) \), \( i = 1, \ldots, l; j = 0, \ldots, m - 1 \).

When the affine transformation parameterization approach is considered, the state vector at time \( k \) is \( \xi(k)^T = [Q^T \ 0^T]^T \), while process noise vector at time \( k \) becomes \( w(k) = [w_x^T(k) \ w_y^T(k)] \). The \( 2l \) visual measurements available at time \( k \) are grouped into the vector

\[
y(k) = [z_x(u_1,k) \ z_y(u_1,k) \ ... \ z_x(u_l,k) \ z_y(u_l,k)]^T
\]

and the output matrix \( H \) can be written as in (6.11), with

\[
\tilde{H} = \begin{bmatrix} B(u_1) & 0 \\ 0 & B(u_1) \\ \vdots \\ B(u_l) & 0 \\ 0 & B(u_l) \end{bmatrix} W
\]

Moreover, in the affine parameterization approach the measurement noise vector assumes the following form

\[
v(k) = [v_x(u_1,k), v_y(u_1,k) \ ... \ v_x(u_l,k), v_y(u_l,k)]^T.
\]
CHAPTER 6. ESTIMATION OF TIME TO CONTACT

As long as the contour dynamics is considered, the choice of matrices $F(k)$ and $G(k)$ is a key point. Indeed, these matrices reflect the a priori knowledge on the contour dynamics. In particular, $F(k)$ represents the deterministic part of the dynamics and accounts for information about different motion components in the image plane, such as translation, rotation and scaling. On the other hand, $G(k)$ explains how random acceleration terms affect the contour dynamics. Two possible choices of the contour dynamics model are shown in the following examples.

**Example 6.1 (Random walk model).** The discrete-time version of the contour dynamics for the affine parameterization random-walk model is simply obtained via discretization of equations (6.8). The dynamics and disturbance matrices of the first equation in (6.10) are given by

$$F(k) = \begin{bmatrix} I_6 & \delta I_6 \\ 0_6 & I_6 \end{bmatrix}, \quad G(k) = \begin{bmatrix} \frac{\delta^2}{2} I_6 \\ \delta I_6 \end{bmatrix} M$$

(6.12)

with $\delta$ denoting the sampling period and $I_m$ standing for the $m$-dimensional identity matrix.

**Example 6.2 (Uniform motion model).** Let us consider the case of uniform motion towards the geometric center of the contour of a fixed planar rigid object. The projected image will undergo a simple expansion, with no translation and rotation. In the affine transformation parameterization introduced in Section 6.2.1, this contour motion is described by the state vector $Q(t) = [1 - s(t) \ 1 - s(t) \ s(t) \ s(t) \ 0 \ 0]'$, where the scale factor $s(t) = d_0/(d_0 - \nu t)$ is evaluated as the ratio between the distance from the object at time $t = 0$ ($d_0$) and that at time $t$, being $\nu$ the (constant) velocity of the uniform motion. The presence of unknown acceleration terms can be accounted by the continuous-time model

$$\ddot{s}(t) = 2\frac{\nu}{d_0} \dot{s}(t)s(t) + w(t).$$

(6.13)

Therefore, matrices $F(k)$ and $G(k)$ can be obtained by linearizing model (6.13) in $s(t)$ (where $t = k\delta$ and $\delta$ is the chosen sampling time) and then discretizing the linearized model. Notice that as $d_0$ and/or $\nu$ are usually unknown, the obtained linear time-varying model is not a priori known. A possible choice is to replace $s(k\delta)$ with an estimate $\hat{s}(k\delta)$, previously computed on the basis of the collected visual measurements up to time $(k-1)\delta$.

6.2.3 Contour tracking via set-membership filtering

Following the approach presented in Chapter 2, noises $w(k)$ and $v(k)$ will be assumed to satisfy the Unknown-But-Bounded hypothesis, thus satisfying equations (2.8). Since no
6.3. **COMPUTATION OF THE TIME TO CONTACT**

Statistical assumptions are made on \( w(k) \) and \( v(k) \), biased and/or nonstationary disturbances, for which a rich *a priori* knowledge is required, can be easily included. Actually, in many practical situations, it is considerably easier to get knowledge on error magnitude bounds.

Differently from Chapters 3-5, both the state and the output equations (6.10) are linear: as a consequence, the parallelotopic approximation techniques presented in Section 2.5.2 can be straightforwardly applied to this problem. Consequently, at time \( k \), the set-theoretic parallelotopic state filter based on Theorems 2.5 and 2.6 provides the set estimate

\[
P(k|k) = \{ \xi(k|k) : \xi(k|k) = \hat{\xi}(k|k) + T(k|k)\alpha, \|\alpha\|_\infty \leq 1 \} \tag{6.14}
\]

where \( \hat{\xi}(k|k) \) is the central estimate of the state vector and \( T(k|k) \) defines the parallelotopic uncertainty region associated to the central estimate. The uncertainty intervals relative to each control point coordinate, and corresponding time derivative, can be obtained by computing the minimum volume axis-aligned box containing the parallelotope \( P(k|k) \).

**Lemma 6.1** Consider the parallelotope \( P(k|k) = P(T(k|k), \xi(k|k)) \) in (6.14), and let \( T(k|k) = \{ t_{ij} \} \). Then, the minimum volume axis-aligned box containing \( P(k|k) \) is given by

\[
B(k|k) = \{ \xi(k|k) : \xi(k|k) = \hat{\xi}(k|k) + D(k|k)\alpha, \|\alpha\|_\infty \leq 1 \} \tag{6.15}
\]

where \( D(k|k) = \text{diag}[d_1, d_2, \ldots, d_n] \) and \( d_i = \sum_j |t_{ij}| \).

### 6.3 Computation of the time to contact

The time to contact \( \tau \) is defined as the time interval between the present instant and the instant when the observing sensor and the point on the object along the optical axis come to collision, under the hypothesis of uniform relative motion.

Denoting by \( A(t) \) the area enclosed by the contour estimated at time \( t \), and by \( \dot{A}(t) \) its time derivative, it can be shown that the time to contact \( \tau \) is given by [22]

\[
\tau = \frac{2A(t)}{\dot{A}(t)} . \tag{6.16}
\]

In the case of contour parameterization through B-splines control points, from Green’s theorem in the plane it is easy to show that the area enclosed by a curve with parameterization \( X(u) \) and \( Y(u) \) is given by:

\[
A(t) = \int_{u_{\text{min}}}^{u_{\text{max}}} X(u; t) \frac{d}{du} Y(u; t) du .
\]
For a B-spline curve it can be shown that the area is given by:

\[
A(t) = \int_{u_{\text{min}}}^{u_{\text{max}}} \sum_i \sum_j (X_i Y_j) B_{ik} \frac{d}{du} B_{jk} du \\
= \sum_i \sum_j (X_i Y_j) \int_{u_{\text{min}}}^{u_{\text{max}}} B_{ik} \frac{d}{du} B_{jk} du.
\]

In matrix notation one gets,

\[
A(t) = X'(t)SY(t)
\]
and its time derivative:

\[
\dot{A}(t) = \dot{X}'(t)SY(t) + X'(t)S\dot{Y}(t)
\]

where \( S = \{s_{ij}\} \), \( s_{ij} = \int_{u_{\text{min}}}^{u_{\text{max}}} B_{ik} \frac{d}{du} B_{jk} du \). Observe that both \( A(t) \) and \( \dot{A}(t) \) are multilinear function of the control points and their derivatives.

Note that \( A(t) \) and \( \dot{A}(t) \) can be estimated quite inexpensively at any iteration of the set-membership filter recursion described in Section 6.2.3. In fact, \( X_i, Y_i, \dot{X}_i, \dot{Y}_i \) are exactly the state variables of the contour dynamics model. Moreover, since the entries of matrix \( S \) are integral parameters which do not depend on time \( t \), they can be computed off-line, thus not affecting the computational burden of the tracking algorithm. Finally, notice that the integral nature of the parameters used to compute \( \tau \) provides a natural noise filter, ensuring a reliable degree of stability of the recursive estimates of the time to contact.

The feasibility region \( P(k|k) \) associated to the estimate provided by the set-membership filter can be employed to compute upper and lower bounds on the time-to-contact estimate, as it will be illustrated in Section 6.4.

For the \( Q \)–parameterization the above reasoning can be repeated. Let us partition equation (6.5) as

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} = \begin{bmatrix}
W_x \\
W_y
\end{bmatrix} Q.
\]

The area contained in the image contour and its derivative become

\[
A(t) = Q'W_x'SW_yQ
\]
\[
\dot{A}(t) = Q'(W_x'SW_y + W_y'SW_x)\dot{Q}.
\]

Due to the particular structure of matrices \( W_x \) and \( W_y \), the diagonal elements of matrix \( W_x'SW_y \) are zero. Hence also for the \( Q \)–parameterization case, \( A(t) \) and \( \dot{A}(t) \) are multilinear functions of the state vector \([Q' \dot{Q}]\) components.

Multilinearity will play a key role in computing the bounds on the time-to-contact estimate (see Section 6.4).
6.4 Exact Error Bounds on the Time-to-Contact Estimate

Let $\xi = [X', Y', X', Y']'$ or $\xi = [Q', \dot{Q}']'$ and let $B$ be the estimated uncertainty box for the control points coordinates or for the affine transformation variables and their respective time derivatives. Following Lemma 6.1, this set can be written as

$$B = \{\xi \in \mathbb{R}^M : \hat{\xi}_i - d_i \leq \xi_i \leq \hat{\xi}_i + d_i, i = 1, \ldots, M\}$$

where $M = 4m$ for control points and $M = 12$ for affine variables.

According to (6.16), the time to contact $\tau(\xi)$ is given by

$$\tau(\xi) = \frac{n[\xi(t)]}{d[\xi(t)]} = \frac{2A(t)}{A(t)} \quad (6.22)$$

From (6.17)-(6.18) and (6.20)-(6.21), it ensues that, for both the control points and the affine parameterization, the time to contact (6.22) is given by the ratio of two multilinear functions in the respective state space variables $\xi_i$, $i = 1, \ldots, M$.

Now, the uncertainty interval $[\tau_{\min}, \tau_{\max}]$ on the time to contact, induced by uncertainty in the parameter vector $\xi$ must be computed. In other words, the following constrained optimization problems must be solved

$$\tau_{\min} = \min_{\xi \in B} \tau(\xi), \quad \tau_{\max} = \max_{\xi \in B} \tau(\xi) \quad (6.23)$$

Notice that problems (6.23) are nonconvex, being the objective function the ratio of two multilinear functions of the independent variables $\xi$. In the following it is shown that both problems (6.23) can be solved in a finite number of steps.

First, a well-known result which will be used later is provided. Let us introduce a special extremal subset of $B$, containing the vertices of the hyperrectangle $B$

$$B_v = \{\xi \in B : \xi_i = \hat{\xi}_i + \alpha_i d_i, \ \alpha_i \in \{-1, 1\}, \ i = 1, \ldots, M\}.$$

**Lemma 6.2** Let $f(\xi) : \mathbb{R}^M \to \mathbb{R}$ be a multilinear function of $\xi$. Then

$$\max_{\xi \in B} f(\xi) = \max_{\xi \in B_v} f(\xi), \quad (6.24)$$

$$\min_{\xi \in B} f(\xi) = \min_{\xi \in B_v} f(\xi). \quad (6.25)$$

Now, the result concerning interval computation can be stated and proven.

**Theorem 6.1** Let the time to contact $\tau$ be defined as in (6.22), via either (6.17)-(6.18) or (6.20)-(6.21). Moreover, let $d(\xi) \neq 0$, $\forall \xi \in B$. Then

$$\tau_{\min} = \min_{\xi \in B_v} \tau(\xi), \quad \tau_{\max} = \max_{\xi \in B_v} \tau(\xi) \quad (6.26)$$
CHAPTER 6. ESTIMATION OF TIME TO CONTACT

Proof.
With reference to the definition of \( \tau \) in (6.22), let us introduce the function

\[
F(\tau; \xi) = \tau d(\xi) - n(\xi).
\]

(6.27)

For a fixed \( \tau \), define the value set

\[
V(\tau) = \{ x \in \mathbb{R} : x = F(\tau; \xi), \text{ for some } \xi \in \mathcal{B} \}. \tag{6.28}
\]

The set \( V(\tau) \) is a segment of the real axis, whose left and right endpoints are

\[
E_l(\tau) = \min_{\xi \in \mathcal{B}} F(\tau; \xi), \quad E_r(\tau) = \max_{\xi \in \mathcal{B}} F(\tau; \xi). \tag{6.29}
\]

From the definition of \( \tau \) in (6.22), it is easy to verify that \( 0 \in V(\tau) \), if and only if \( \tau_{min} \leq \tau \leq \tau_{max} \). By a continuity argument, the minimum and maximum values of \( \tau \) for which the equation

\[
F(\tau; \xi) = 0 \tag{6.30}
\]

admits a solution for some \( \bar{\xi} \in \mathcal{B} \), are values of \( \tau \) such that one of the endpoints of the value set \( V(\tau) \) coincides with the origin of the real axis, i.e.

\[
E_r(\tau_{min}) = 0 \quad \text{or} \quad E_l(\tau_{min}) = 0;
\]

\[
E_r(\tau_{max}) = 0 \quad \text{or} \quad E_l(\tau_{max}) = 0.
\]

Since the function \( F(\tau; \xi) \) is multilinear in \( \xi \), by Lemma 6.2 it is concluded that for any fixed \( \tau \), and hence also for \( \tau = \tau_{min} \) and \( \tau = \tau_{max} \), there exist \( \xi^m, \xi^M \in \mathcal{B} \) such that

\[
\xi^m = \arg \{ \min_{\xi \in \mathcal{B}} F(\tau; \xi) \}, \quad \xi^M = \arg \{ \max_{\xi \in \mathcal{B}} F(\tau; \xi) \}. \tag{6.31}
\]

This completes the proof.

Remark 6.1 Theorem 6.1 states that the evaluation of the bounds on \( \tau \) can be made by simply considering a very special extremal subset of the entire parameter uncertainty set. In principle, this result reduces a nonconvex optimization problem to a finite time search algorithm.

Remark 6.2 The main difference between using the control points and the affine parameterization is that in the first approach, the number of vertices may be high, depending
on the number of control points used to parameterize the contour of the considered image area. However, since for the time-to-contact problem the reference contour on the object is chosen by the operator, in practical applications it is always possible to use a limited number of control points.

In the second approach the dimension of the state vector does not depend on the number of control points and is always equal to 12. Hence, this approach is less computationally demanding than the previous one, when more than 3 control points are used. Under the hypothesis of uniform relative motion along the optical axis, translational parameters in (6.4) are identically zero and the dimension of the state vector $\xi$ can be reduced to 8, thus increasing the computational efficiency of the affine parameterization approach. However, it should be noted that the affine parameterization is less general than the parameterization by control points, as it provides an approximate description of the motion on the image plane, for planar rigid objects.

In Section 7.4, results obtained by applying the above estimation technique to simulated image sequences are presented and discussed.
Chapter 7

Simulation Experiments

In this chapter, the algorithms proposed in Chapters 3, 4, 5 and 6 are tested by means of some simulation experiments. Section 7.1 presents results for algorithms dealing with localization and pose estimation in known environments. Section 7.2 deals with the algorithms proposed for the simultaneous localization and map building problem, while Section 7.3 presents some results on maps fusion and cooperative localization and mapping. Finally, results on some tests on the time-to-contact estimation algorithm are presented in Section 7.4. For the latter problem it is also provided a comparison between standard Kalman filter techniques and the proposed set-membership approach.

7.1 Pose Estimation in Known Environments

In this section some tests on the algorithms presented in Chapter 3 are reported. Experiments have been performed to verify the quality of the estimates.

7.1.1 Localization from visual angle measurements

First, the set-membership localization algorithms presented in Section 3.3 have been tested. Both algorithms, providing respectively box and parallelotopic approximations of the feasible position set, have been employed.

Static setting

Initially, it has been considered the situation in which the robot is still and has to locate itself with respect to known landmarks. The aim of these experiments is to determine how the performances of the proposed algorithm depend on the error bounds and the available information.
At a fixed time \( k \), the navigating vehicle is supposed to be located at the center of a square room, of 20 meters side. This means that \( \mathcal{R}(k|k-1) = B([10, 10], [0, 0]) \), in a reference system centered at the vehicle position. The sets \( \mathcal{R}(k|k) \) defined in (3.12) are computed according to the recursive strategies outlined in Section 3.3.

In a first group of experiments, it is assumed that 5 landmarks are identified in the scene. The visual angle measurements (see Fig. 3.2) are corrupted by additive noise \( v_{ij} \), generated as an i.u.d. signal satisfying the constraint \( |v_{ij}| \leq \epsilon^\ast, \forall i, j \). The centers of boxes and parallelotopes provided by the algorithms have been considered as nominal estimates \( \hat{p} \) of the vehicle position. The localization errors \( \|\hat{p} - p\| \) (in meters), obtained for different values of \( \epsilon^\ast \) ranging from 0.5 to 5 degrees, are reported in Fig. 7.1a. Results are averaged over 1000 different landmark configurations (randomly generated) in the square room. In Fig. 7.1b, the area of the approximating boxes and parallelotopes are compared with that of the minimum area box containing the exact position set \( \mathcal{R}(k|k) \), i.e. \( B\{\mathcal{R}(k|k)\} = B\{\mathcal{R}(k|k-1) \cap \mathcal{M}(k)\} \). Notice that the parallelotopic uncertainty sets are smaller than the corresponding boxes, and that their area is quite close to that of the minimum outer box \( B\{\mathcal{R}(k|k)\} \) (whose computation has been performed via a 2 dimension gridding and hence it involves a much higher computational burden, compared to the proposed approximation algorithms).

The same experiments are repeated for the case of 10 landmarks, in the same setting presented above. Results concerning position errors and uncertainty set areas are reported.
7.1. POSE ESTIMATION IN KNOWN ENVIRONMENTS

Figure 7.2: Localization with 10 landmarks: a) average position errors; b) average uncertainty set areas for different angle noise bounds (*: boxes; +: parallelotopes; o: minimum box containing the exact feasible position set $\mathcal{R}(k|k-1) \cap \mathcal{M}(k)$).

in Fig. 7.2. It can be noticed that nominal position errors and area of each uncertainty set have been remarkably reduced.

Another group of experiments has been performed, assuming relative visual angle error bound. In these simulations, $\epsilon_{ij}$ is set to the 20% of the current visual angle measurement $\theta_{ij}$. The estimated position for 100 different settings including 5 landmarks are reported in Fig. 7.3. The average position error is 0.49m for parallelotopes and 0.51m for boxes. Average uncertainty sets are 2.32m$^2$ and 5.42m$^2$, respectively.

The same experiment is repeated placing the vehicle almost at one corner of the square room. In this case, the visual angles range only form 0 to 90 degrees, thus reducing the maximum size of measurement noise. The results are depicted in Fig. 7.4, where average position errors are 0.45m for parallelotopes and 0.51m for boxes. Average uncertainty areas are reduced to 1.77m$^2$ and 2.07m$^2$, respectively. The results obtained in all the performed experiments compare favorably with typical results presented in recent literature on mobile robot localization (see, e.g., [8, 81]).

**Dynamic setting**

In the following group of experiments, the vehicle follows a given trajectory inside a square room. The dynamic model $p(k+1) = p(k) + u(k) + w(k)$ is considered for the position of the vehicle, with $k = 0, \ldots, 15$. The driving input $u(k)$ describes the required trajectory, while the disturbance $w(k)$ is assumed to satisfy (3.4), with constant bound
Figure 7.3: Position estimates \((p = [0 0]')\) for 100 different groups of 5 landmarks, and relative error bounds (20%).

Figure 7.4: Position estimates \((p = [-9.5 -9.5]')\) for 100 different groups of 100 landmarks, and relative error bounds (20%).
7.1. POSE ESTIMATION IN KNOWN ENVIRONMENTS

Figure 7.5: Dynamic localization with 5 landmarks and absolute noise bound $\epsilon_v(k) = 5$ degrees: a) boxes; b) parallelotopes ($\times$: landmarks; solid line: robot trajectory; dashed box: initial feasible position set $R(0)$).

$\epsilon_v^i(k) = 0.2, i = 1, 2, \forall k$. A first group of experiments has been performed with 5 landmarks in the room and visual angles corrupted by bounded noise, with absolute noise bound equal to 5 degrees. A typical run employing boxes or parallelotopes as approximating regions is reported in Fig. 7.5.

Figure 7.6 shows the time variation of uncertainty set areas $R(k|k)$ and position errors $p(k) - \hat{p}(k)$ (where $\hat{p}(k)$ is the center of the approximating box or parallelotope). Here the results are averaged over 100 different noise realizations, for the same trajectory and landmarks configuration in the room.

Another group of experiments is performed for the case of 20% relative noise bound (i.e., $\epsilon_v^i(k) = 0.2\theta_{ij}(k)$ for each visual angle measurement, at each time instant $k$). This time, 8 landmarks in the room are used for dynamic localization. A typical run is reported in Fig. 7.7, while average uncertainty set areas and position errors are shown in Fig. 7.8.

Figures 7.5-7.8 show that the set-membership localization algorithms are able to remarkably reduce localization error and related uncertainty, also in the presence of disturbances in the vehicle dynamics. It can be observed that the parallelotopes generally give better approximations of the feasible set (especially during transients) with respect to boxes, while the two algorithms show similar performance in terms of nominal position error. The large reduction of uncertainty achieved during the first half of the trajectory in Fig. 7.7 is due to the fact that the robot reaches a location where most of the visual angles are
Figure 7.6: Average uncertainty set areas and position error for dynamic localization with 5 landmarks and absolute noise bound $\epsilon^v(k) = 5$ degrees (solid line: parallelopodes; dashed line: boxes).

Figure 7.7: Dynamic localization with 8 landmarks and relative noise bound $\epsilon^v_{ij}(k) = 0.2\theta_{ij}(k)$: a) boxes; b) parallelopotes ($\times$: landmarks; solid line: robot trajectory; dashed box: initial feasible position set $\mathcal{R}(0)$).
7.1. POSE ESTIMATION IN KNOWN ENVIRONMENTS

small, and therefore small relative errors affect the visual angle measurements.
It is worth noting that, due to the approximation introduced by recursive outbounding, the
size of uncertainty sets can be further reduced by reprocessing the same measurements
several times. In other words, the quality of set approximation can trade-off with the
required computational power in order to obtain the maximum uncertainty reduction
allowed by the available bandwidth.

7.1.2 Pose estimation from relative orientation measurements

Similarly to what has been presented in the previous section, a group of experiments have
been performed to test the pose estimation algorithm presented in Section 3.4.

Static setting

The robot, at time instant $k$, is located at the center of a room, of 20 meters side.
Since this time also orientation must be estimated, this implies that $\mathcal{R}(k|k-1) =
\mathcal{B}([10 10 \pi \pi', [0 0 0 0])$, in a reference system centered at the vehicle position. As point-
wise estimates of the vehicle position and orientation, the center of the box $\mathcal{R}_v(k|k)$
and the center of the interval $\mathcal{R}_o(k|k)$ are considered. For each identified landmark in the

Figure 7.8: Average uncertainty set areas and position error for dynamic localization with
8 landmarks and relative noise bound $\epsilon_{ij}^r(k) = 0.2\theta_{ij}(k)$ (solid line: paralleloptopes; dashed
line:boxes).
environment, the robot performs a measurement of relative orientation.

In a first test, it is assumed that 5 landmarks are identified in the scene. Angle measurements between the landmarks and the robot orientation (see Fig. 3.6) are corrupted by additive noise \( v_i(k) \), generated as an i.u.d. signal satisfying the constraint \( |v_i(k)| \leq \epsilon^v \), \( \forall i, k \). Data are processed according to the algorithm presented in Section 3.4, including the iterative refinement strategy. Position errors, orientation errors (in degrees) and the associate uncertainties, for different values of \( \epsilon^v \) ranging from 0.5 to 2.5 degrees, are reported in Table 7.1. Results are averaged over 1000 different landmark configurations in the square room. A second set of experiments is performed assuming that 10 landmarks are identified in the scene. In this case, the pose uncertainty and the average estimation error are remarkably reduced, as shown in Table 7.2.

As stated in Section 3.4.2, it is possible to use the set-valued estimate on the robot orientation to further improve the quality of the position estimate. Some tests have been performed to verify the quality of this refinement. Figure 7.9 shows the improvements on the localization error and uncertainty area sets due to the recursive refinement (3.34)-(3.35), for \( \epsilon^v \) ranging from 0.25 to 2.5 degrees, and 5 landmarks identified in the room.

<table>
<thead>
<tr>
<th>Error bound</th>
<th>Position error (m)</th>
<th>Box area (m²)</th>
<th>Orientation error (deg)</th>
<th>Orientation width (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.07</td>
<td>0.21</td>
<td>0.25</td>
<td>3.07</td>
</tr>
<tr>
<td>1.00</td>
<td>0.14</td>
<td>0.48</td>
<td>0.48</td>
<td>5.33</td>
</tr>
<tr>
<td>1.50</td>
<td>0.19</td>
<td>1.13</td>
<td>0.72</td>
<td>8.12</td>
</tr>
<tr>
<td>2.00</td>
<td>0.26</td>
<td>1.76</td>
<td>0.94</td>
<td>10.36</td>
</tr>
<tr>
<td>2.50</td>
<td>0.31</td>
<td>2.25</td>
<td>1.08</td>
<td>12.50</td>
</tr>
</tbody>
</table>

Table 7.1: Simulation results with 5 landmarks, after iterative refinement.

<table>
<thead>
<tr>
<th>Error bound</th>
<th>Position error (m)</th>
<th>Box area (m²)</th>
<th>Orientation error (deg)</th>
<th>Orientation width (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>0.02</td>
<td>0.0094</td>
<td>0.09</td>
<td>1.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.04</td>
<td>0.039</td>
<td>0.18</td>
<td>1.97</td>
</tr>
<tr>
<td>1.50</td>
<td>0.07</td>
<td>0.097</td>
<td>0.29</td>
<td>3.08</td>
</tr>
<tr>
<td>2.00</td>
<td>0.08</td>
<td>0.16</td>
<td>0.40</td>
<td>4.17</td>
</tr>
<tr>
<td>2.50</td>
<td>0.11</td>
<td>0.22</td>
<td>0.46</td>
<td>4.89</td>
</tr>
</tbody>
</table>

Table 7.2: Simulation results with 10 landmarks, after iterative refinement.
7.2 Simultaneous Localization and Map Building

This section illustrates some tests performed on the set-membership SLAM algorithms devised in Section 4.4. Both box and parallelotopic approximation algorithms have been tested over the same data sets.
Figure 7.10: Dynamic pose estimation: landmarks (*), position uncertainty (boxes), true trajectory (solid line) and estimated trajectory (dashed line).

Figure 7.11: Dynamic pose estimation: orientation uncertainty intervals and true orientation (solid line).
Figure 7.12: Examples of the results provided by the algorithm: a) Landmarks map realized by the algorithm using boxes as set approximations (real landmark position: \(\times\); estimates after the first iteration: dashed box; final estimates: solid box); b) Landmarks map realized by the algorithm using paralleloptopes as set approximations; c) Robot position estimate provided by the box approximations (initial position: \(*\); uncertainty box at each time: solid box); d) Robot position estimate provided by the paralleloptopic approximations. Results are relative to the first rough circle run covered by the robot.
In all the simulations the robot moves in three rough circles, covering a distance of about 100 meters, in a square area of 20 meters side. In this area 10 landmarks are spread randomly. The vehicle moves and activates the SLAM algorithm once every meter covered. The simulations reported do not consider any problem of landmark visibility, and assume that at each iteration a scan of the whole horizon is available, i.e. at each time instant, all the landmarks present in the scene are perceived. It is assumed that the robot is equipped with a compass and wheel odometers, and it performs measurements on the environment through a stereo couple, which takes panoramic scans each time the algorithm is triggered.

To model measurements from a stereo couple, distance and orientation measurements have been corrupted by additive noises $v_{d_i}(k)$ and $v_{\theta_i}(k)$, according to (4.4). Noises have been generated as i.u.d. signals satisfying eqns. (4.7) and (4.8), with constant bounds for the latter ($\epsilon^v_{i}(k) = \epsilon^v_{i}, \forall i, k$), while the bounds on the former depend quadratically on the distance measured ($\kappa_{d_i}(k) = \kappa_{d_i}(X(k))$). This is a standard measurement model for robots using stereovision data to extract landmark information [90]. In order to take into account the error accumulation occurring in odometric integration, error $w_{r_i}(k)$ in eq. (4.3) is generated as a nonstationary i.u.d. signal, with both range and mean value proportional to the distance covered during the last robot move (the second choice is done to take into account the wheel slippage, which is relevant on irregular, outdoors environment. This effect causes odometers to provide measurements that are on average greater than the true values).

Figure 7.12 reports the results provided by the algorithm during the first of the three circles covered by the robot. During this experiment, the odometry error bound $\epsilon^w$ is set to 5% of the distance covered, uncertainty bound on orientation measurement is $\epsilon^v_{\theta_i} = 3^\circ$, while the constant for the bound on distance measurements is $\kappa_d = 0.002$ (these are typical values for a commercial stereo couple).

Figure 7.13 shows some relevant quantities for a typical run: in both cases $R_{Li}(0|0)$ coincides with the whole cartesian plane $\forall i$. Notice that the parallelotopic approximation shows a faster transient for the mapping task, while the two methods are almost equivalent in terms of final average landmark uncertainty, robot position estimation error, and average robot position uncertainty.

To provide a more general evaluation of the results given by the algorithm, three different quantities have been considered: the average area of the landmark uncertainty sets, the average area of the robot uncertainty set during the whole run and the average robot position estimation error. Averages are computed over 100 different landmark configurations and robot paths. Figure 7.14 shows how the aforementioned quantities depend on distance measurement errors. In particular, figure 7.14a reports the average area of the
Figure 7.13: For the same experiment shown in Fig. 7.12: a) Average area of landmark boxes (□, solid line) and parallelograms (○, dashed line) at each step; b) Area of uncertainty box (□, solid) and parallelogram (○, dashed) for the robot position at each step; c) Robot position estimation error at each step for the box (□, solid) and parallelogram (○, dashed) set approximation.
Figure 7.14: a) Average area of landmark boxes (□, solid line) and parallelotopes (○, dashed line) for $\epsilon^\mu_l(k) = 0.1$, $\epsilon^\theta_i = 4^\circ$ and $\kappa_d \in [0.0002, 0.01]$; b) Area of uncertainty box (□, solid) and parallelotope (○, dashed) for the robot position averaged on the whole run (same error bounds); c) Average robot position estimation for the box (□, solid) and parallelotopic (○, dashed) set approximation (same error bounds).
landmarks uncertainty sets at the final step of the algorithm, for different values of the constant \( \kappa_d \) \((\kappa_d \in [0.0002, 0.01])\), when the other error bounds are fixed \((\epsilon^w = 4^\circ, \text{mean value of } w_i(k) \text{ equal to } 5\% \text{ of the covered distance } \|m(k)\|, \text{while } \epsilon^w = 0.1|m(k)|)\). The same information is provided for the average robot position uncertainty over the whole run (Fig. 7.14b), and for the average robot position estimation error (Fig. 7.14c).

Figure 7.15 reports how the same quantities depend on odometric measurement errors: we note that the parallelotopic approach turns out to be less sensitive to the increase of the odometric error bounds. As shown in Fig. 7.15, when \( \epsilon_i^w(k) \) increases, the relative performance of the two set approximating algorithms shifts in favor of the parallelotopic algorithm.

Finally, observe that the relative performance of the two algorithms depends also on the values of the exteroceptive measurement noise bounds: this is due to the fact that the shape of the measurement set \( M_l_i \) depends on the values of the distance and orientation error bounds: as a matter of fact, there are values of \( \kappa_d \) and \( \epsilon^w \) for which this set turns out to be definitely wider in one direction, thus having a rough rectangular shape. In this case, the parallelotopic approach provides better performances, because the area of the approximating boxes is on average larger than that of the approximating parallelotopes approximation. On the other hand, there are values of the noise bounds for which the shape of the sets is almost a square; in that case the box algorithm works slightly better than the parallelotopic one.

### 7.3 Cooperative Localization and Map Building

In this section, the algorithms presented in Chapter 5 have been tested via numerical simulations. Both static map fusion and dynamic collaborative localization and mapping have been considered.

**Map fusion**

A first group of experiments has been carried out to evaluate the performance of the static set-valued map fusion algorithm proposed in Section 5.3. A typical run of the algorithm is reported in Fig. 7.16, where Fig. 7.16a, b, c, d are the four self referenced maps provided by each robot (notice how each agent assume to perfectly know its own position), while Fig. 7.16e is the output of the proposed algorithm.

In order to quantify the improvement provided by map fusion over single-robot set-valued maps, several simulations with increasing number of robots \((n \geq 2)\) and non-sensing features \((m \geq 0)\), randomly spread over a fixed area, have been performed. Results are
Figure 7.15: a) Average area of landmark boxes (□, solid line) and parallelotopes (○, dashed line) for $\epsilon_i^m (k) \in [0.02, 0.1]$, $\epsilon_i^\theta = 4^\circ$ and $k_d$ equal to 0.0002 (left) and 0.002 (right); b) Area of uncertainty box (□, solid) and parallelotope (○, dashed) for the robot position averaged on the whole run (same error bounds); c) Average robot position estimation for the box (□, solid) and parallelotopic (○, dashed) set approximation (same error bounds). Note how the parallelotopic approach is generally less sensitive to the increase of odometric uncertainty.
7.3. COOPERATIVE LOCALIZATION AND MAP BUILDING

Figure 7.16: a,b,c,d: Self referenced set-valued maps produced by each robot. e: map produced by the proposed fusion algorithm. All the coordinate systems have been translated in order to have maps overlapping.
### Table 7.3: Percentage of total uncertainty reduction due to the fusion algorithm, with respect to the average total uncertainty over all initial maps.

<table>
<thead>
<tr>
<th>$n \setminus m$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>72.8</td>
<td>63.6</td>
<td>52.8</td>
<td>46</td>
<td>45.4</td>
<td>43.6</td>
</tr>
<tr>
<td>3</td>
<td>74.3</td>
<td>71.2</td>
<td>67.4</td>
<td>63.8</td>
<td>64.2</td>
<td>57.7</td>
</tr>
<tr>
<td>4</td>
<td>77.3</td>
<td>74.9</td>
<td>72.4</td>
<td>70.4</td>
<td>69.2</td>
<td>67.6</td>
</tr>
<tr>
<td>5</td>
<td>79.1</td>
<td>77.8</td>
<td>76.4</td>
<td>74.8</td>
<td>72.8</td>
<td>72.2</td>
</tr>
<tr>
<td>6</td>
<td>80.8</td>
<td>80</td>
<td>78.9</td>
<td>77.6</td>
<td>76.9</td>
<td>75.8</td>
</tr>
</tbody>
</table>

### Table 7.4: Percentage of total uncertainty reduction due to the fusion algorithm, with respect to the uncertainty of the best initial map.

<table>
<thead>
<tr>
<th>$n \setminus m$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>71.9</td>
<td>44.3</td>
<td>31.6</td>
<td>25</td>
<td>21.5</td>
<td>18.7</td>
</tr>
<tr>
<td>3</td>
<td>54.6</td>
<td>48.6</td>
<td>43.8</td>
<td>36.9</td>
<td>35.1</td>
<td>31.7</td>
</tr>
<tr>
<td>4</td>
<td>54.3</td>
<td>50.8</td>
<td>47.4</td>
<td>42.4</td>
<td>43.2</td>
<td>39.3</td>
</tr>
<tr>
<td>5</td>
<td>57.1</td>
<td>52.7</td>
<td>51.5</td>
<td>48.9</td>
<td>46.8</td>
<td>45</td>
</tr>
<tr>
<td>6</td>
<td>57.3</td>
<td>57.1</td>
<td>55.3</td>
<td>53.2</td>
<td>51.4</td>
<td>48.8</td>
</tr>
</tbody>
</table>

shown in Tables 7.3 and 7.4, where it is reported the improvement with respect to the mean map uncertainty and the minimum map uncertainty, respectively (where by “map uncertainty” it is meant the total area of all uncertainty sets in a map). Percentages are evaluated over 100 experiments.

### Dynamic setting

The dynamic SM-CLAM algorithm described in Section 5.2 has been tested in a setting where 4 moving agents and 10 static landmarks are randomly spread over a square region of about 8000 $m^2$. For each robot the slow-dynamic model (3.1) has been employed. Each robot performs distance and angle measurements, corrupted by additive noises, generated as i.u.d. signals satisfying (4.7) and (4.8), with constant bounds for the latter ($\epsilon_i^a(k) = \epsilon^a$, $\forall i, k$), while the bounds on the former depend quadratically on the distance measured ($\epsilon_i^d(k) = \kappa_d d_i^2(X(k))$). Disturbance $w_i(k)$ is generated as a nonstationary i.u.d. signal, with mean value proportional to the distance covered during the last robot move. Robots cover rough circles, with different radii. Only one map fusion is performed after the first exteroceptive measurements, then the algorithm presented in Section 5.2 is run. During this experiment, the odometry error bound is set to 10% of the distance covered,
7.3. COOPERATIVE LOCALIZATION AND MAP BUILDING

Figure 7.17: Comparison of single-robot (dashed lines) and multi-robot (solid line) SM-SLAM: total robot uncertainty.

Figure 7.18: Comparison of single-robot (dashed lines) and multi-robot (solid line) SM-SLAM: total landmark uncertainty.
uncertainty bound on angle measurements is $3^\circ$, while the constant for the bound on distance measurements is $\kappa_d = 0.002$. Figures 7.17 and 7.18 compare the results provided by the CLAM approach to those of the single-robot algorithm. Figure 7.17 shows the total area of uncertainty boxes for the 4 robots, computed by each agent via the single-robot SM-SLAM (dashed lines) and by applying the proposed CLAM algorithm (solid line). In Fig. 7.18 the same comparison is depicted for the total area of uncertainty boxes for the 10 landmarks.

All simulations demonstrate that the improvement of multi-robot SLAM with respect to the single-robot case is remarkable, in both static map fusion and dynamic localization and map updating.

### 7.4 Time-to-Contact Estimation

This section reports the results of several experiments performed in order to test the algorithm presented in Section 6.4. The affine parameterization introduced in Section 6.2.1 has been always used in the performed experiments. The center of the paralleloptope has been chosen as the nominal estimate of the variables involved in the evaluation of the time to contact, and Theorem 6.1 has been exploited to provide upper and lower bounds on that quantity. A first experiment has been performed by using a set of simulated images obtained moving a virtual camera at constant speed, along the optical axis and towards
7.4. TIME-TO-CONTACT ESTIMATION

Figure 7.20: Set-membership time-to-contact estimate in the case of exact a priori knowledge $\hat{\tau}_0 = \tau(0)$. a) True time to contact (dashed); nominal estimate (solid central); error bounds (solid outer); b) Volume of the estimated uncertainty paralleloptope.

the geometric center of the contour of a synthetic object (see Fig. 7.19a).

Several trials have shown that the dead reckoning model (6.12) is not suitable for our purposes. In fact, while it provides good estimates for the affine parameter $Q(t)$, the estimate of the derivative $\dot{Q}(t)$ is affected by a divergent error. This is due to two different features of the dynamical system (6.10)-(6.12). First, measurements are directly related only to $Q(t)$. Second, the random walk model is a very rough description of uniform motion (see eq. (6.13)), and the approximation error is greater on those state components that have no direct influence on the measured quantities. As a matter of fact, following either the set-membership or the Kalman filtering approach, although model (6.12) is appropriate for tracking image contours (see, e.g. [9], [11]), it cannot be used for time-to-contact estimation.

A discrete-time linearization of eq. (6.13) has been adopted as dynamic motion model, with an initial estimate $\hat{\tau}_0$ of the time to contact, generally different from the true value $\tau_0 = \tau(0) = \frac{d}{\rho}$. Obviously, the closer the initial estimate to the true time to contact, the better the model prediction. Beside $\hat{\tau}_0$, the other tuning parameters of the set-membership algorithm are the bounds $\epsilon^w$ and $\epsilon^v$ on $w_k$ and $v_k$, respectively. These bounds represent the confidence level in the dynamic model and the measurement accuracy, respectively.

Figure 7.20 shows the results of the set-membership filter for the case when it is assumed that $\hat{\tau}_0 = \tau(0)$ (exact initial condition). In this case, the model is well tuned, and con-
sequently $\epsilon^w$ can be chosen small with respect to $\epsilon^v$. The algorithm provides both good nominal estimates for $\tau$ and tight bounds on that value (up to less than 1/5 of the sampling time $\delta$, here set to 0.5 sec.).

Figure 7.21 shows the results for the case when the value of the parameter $\hat{\tau}_0$ is different from the true value of the time to contact at $t = 0$. Figure 7.21a (7.21b) reports simulation results for $\hat{\tau}_0 = 9.5$ ($\hat{\tau}_0 = 15.5$), while the true value is 12.5. In these cases, the model will underestimate (overestimate) the true time to contact. Due to the strong divergent behavior of the solution of eq. (6.13), the model prediction dominates on the measurement update, no matter how large (i.e., unreliable) is the value of the model error $\epsilon^w$. As a consequence, there exists a time instant when the feasible set provided by the algorithm becomes empty, and this will cause a break in the recursive procedure. It is interesting to note that the set-membership filter can be tuned in order to stop when the true value is outside the bounds provided by the algorithm (see Fig. 7.21). This feature can be used to verify the reliability of the model itself, and surely is one of its most appealing features (see Section 7.4.1). The algorithm has been tested also on a sequence of synthetic images for which the optical axis does not cross the geometric center of the object shape (see the experimental setup in Fig. 7.19b). In this case model (6.13) is not able to describe the translational motion added to the optical axis aligned motion. As a consequence, there exists a drift introducing an error on both measurement and model updates.
In Fig. 7.22, results are shown for $\hat{\tau}_0 = \tau(0)$ and for an experiment with motion along an axis not intersecting the center of the image. Apart from the first four steps, needed to catch the drifting component, the estimates exhibit the same behavior as in the first experiment (Fig. 7.20). Note, however, that the volumes of the paralleloptopes are definitely larger (up to $10^6$ times) than those of Fig. 7.20. This is due to the increased values of $\epsilon^w$ and $\epsilon^v$, necessary in order to keep the feasible set nonempty during the transient.

Also in this case, when $\hat{\tau}_0 \neq \tau_0$ the algorithm shows a behavior similar to that reported in Fig. 7.21.

### 7.4.1 Comparison with Kalman filter approach

As noted in Section 6.2.3, the problem of estimating the time to contact boils down to estimating the state of a linear system. In this case, it makes sense to compare the performance of the set-membership algorithm to that of the classical Kalman filter. It is relevant to point out that the two approaches are quite different. set-membership estimation provides hard bounds on the state estimates using suitable approximating regions; Kalman filtering provides confidence ellipsoids for the estimates, under the assumption that all noises are Gaussian.

In order to fairly compare the two methods, the covariance matrices of the Kalman filter has been chosen so that it is possible to establish a relationship between the noise
CHAPTER 7. SIMULATION EXPERIMENTS

Figure 7.23: Bounds provided by set-membership (solid) and Kalman filtering (dashed) in the ideal case ($\hat{\tau}_0 = \tau(0)$); true time to contact (dotted).

bounding boxes defined for the set-membership filter, and the 99% confidence ellipsoid for Gaussian noises in the Kalman filter approach. In particular, two different choices have been considered for the confidence ellipsoids: (i) the maximum volume ellipsoids completely contained in the noise boxes; (ii) ellipsoids with the same ratio between semiaxes and the same volume of the noise boxes.

First, the ideal case reported at the beginning of Section 7.4, under the hypothesis $\hat{\tau}_0 = \tau(0)$ has been considered. In this case, the Kalman filter provides better performances than the set-membership algorithm: the former shows faster convergence and gives tighter bounds. Figure 7.23 reports the upper and lower bounds for the time-to-contact estimate provided by the two algorithms. Note that for both approaches lower and upper bounds on the time to contact are evaluated on the minimum volume box containing the parallelotope (set-membership) and the confidence ellipsoid (Kalman), according to Theorem 6.1.

The specific features of the set-membership algorithm become more evident when the experimental setting is closer to a realistic setup. This is shown in Fig. 7.24, where it is assumed that the initial estimate $\hat{\tau}_0$ is larger than the true value $\tau(0)$. Consider the case when inner bounding confidence ellipsoids are used. It is easily verified that while the set membership filter always gives guaranteed bounds and stops when the true time to contact is no longer inside the bounds, the Kalman filtering approach provides tighter but incorrect bounds. Figure 7.24.a reports a case in which from time step 11 on, the true time to collision is outside the Kalman bounds.

The same situation arises when one considers larger confidence ellipsoids. Fig. 7.24b reports the results for the case when paralleloptopes and ellipsoids with equal volume are
Figure 7.24: Bounds provided by set-membership (solid) and Kalman (dashed) algorithms (true time to contact is dash–dotted) in the case $\hat{\tau}_0 > \tau(0)$. a) Inner bounding ellipsoids for the parallelotopes; b) Equivolume confidence ellipsoids.

Figure 7.25: Bounds provided by set-membership (solid) and Kalman (dashed) algorithms (true time to contact is dash–dotted) in the case $\hat{\tau}_0 < \tau(0)$. a) Inner bounding ellipsoids; b) Equivolume confidence ellipsoids.
considered. In this case, the bounds provided by the two algorithms are almost identical. Nevertheless, while the set-membership algorithm is able to detect when the true time to contact exceeds the estimated error bounds, the Kalman filter cannot do the same (see time instant 17 in Fig 7.24b).

Figure 7.25 presents similar results for the case when \( \hat{\tau}_0 < \tau(0) \). In this case the model underestimates \( \tau \), but the same observations as in the previous example hold.
Chapter 8

Conclusions

In this thesis, a new approach to estimation problems in the mobile robotics research field has been presented: set-theoretic strategies have been applied to several nonlinear estimation problems relevant to autonomous navigation.

A fundamental issue in the navigation problem consists of localizing an exploring agent in a possibly unknown environment, using information provided by proprioceptive and exteroceptive sensors that perform measurements on the vehicle itself and on the environment.

This results in an estimation problem that demands the development of nonlinear filters. In the presence of unknown-but-bounded process and measurement errors, the proposed approach provides guaranteed estimates, while maintaining a computational burden comparable to that of classic statistical approaches.

Some tools for developing fast set-theoretic algorithms based on set approximations have been provided:

- A general state decomposition technique has been introduced. This technique allows one to consider set intersections on subspaces of reduced dimensionality, thus allowing for simpler algorithms.

- Considering the shape of the uncertainty sets associated with the sensor measurements (visual angle between landmarks, relative orientation and distance), set approximation algorithms, using orthotopes and parallelotopes as approximating regions, have been developed for dynamic position estimation in known environments. The computational burden of the proposed algorithms is linear in the number of measurements collected at each time step. Simulation tests on these algorithms (see Section 7.1.1) provide encouraging results both in static and dynamic settings. Using
the state decomposition technique, also the pose (position and orientation) estimation problem has been tackled. The algorithm is based on recursive refinements of the estimated sets (see Section 3.4.2), thus allowing for the desired trade-off between quality of the estimates and computational burden required to obtain them. Simulation experiments confirm that a significant improvement can be achieved both on position errors and position uncertainty (see Section 7.1.2). Also in this case, experiments in dynamic setting provide interesting results.

- Simultaneous localization and map building (SLAM) problem has been tackled in Chapter 4. In this case the robot explores an unknown environment, and has to perform both self-localization and mapping at the same time. Both problems can be cast in a nonlinear state estimation problem, which is characterized by high dimensionality of the state vector. Statistical approaches, generally based on EKFs, are usually employed to tackle this problem. The main issue with these techniques is their high computational requirements. When the SLAM problem is tackled in the set-membership framework, and both set approximation and state decomposition techniques are employed to reduce the computational complexity, it is possible to devise algorithms whose storage space requirement is linear in the number of landmarks in the scene, while the computational burden is linear in the number of features perceived at each time instant. Consequently, the proposed algorithm is a good candidate whenever wide (or feature dense) environments must be explored. Performances of the proposed algorithms have been tested with group of experiments, reported in Section 7.2: box and parallelogones provide similar results, usually depending on the magnitude of the noise bounds. Nonetheless, the parallelogonal approach is usually less sensitive to the increase of odometry error bounds.

- The aforementioned approach can be easily adapted to tackle the cooperative localization and map building (CLAM) problem, where a team of robots is supposed to explore the same environment. Chapter 5 illustrates the extension to the case when \( n \) robots explore a planar environment where \( m \) static distinguishable landmarks can be identified. A centralized approach to the problem is proposed: an algorithm fuses all the available information from each robot. The main difference with the single-robot SLAM algorithm concerns its initialization: since several agents must be located on the same map, it is not possible to choose arbitrarily the origin of the reference system. A set-valued map fusion algorithm has been developed. This algorithm provides an efficient solution to the problem of finding the minimum-area map compatible with the information contained in the description of the environ-
ment provided by each robot. Exploiting the structure of box-based set-valued maps, the map-fusion task can be performed by solving two LP problems. Simulation experiments show that remarkable improvements can be achieved with this method (about 50% of uncertainty reduction with respect to the best single robot map, see Section 7.3).

- The recovery of the time to contact (time needed from a moving observer to reach a fixed target, while moving at constant speed along the optical axis) has been studied in a set-membership framework. The problem of contour tracking, whose solution is needed to dynamically evaluate the time to contact, can be cast as a linear state estimation. Therefore, the set-theoretic recursive state estimation algorithms presented in Chapter 2 can be applied to the problem. Parallelotopes have been chosen as approximating sets. Given the peculiar dependence of the time to collision on the estimated state, upper and lower bounds on the aforementioned quantity can be easily evaluated by maximizing and minimizing a multilinear function on a finite set, the vertices of a box outbounding the approximating parallelotope. Results of experiments on simulated image sequences, and comparison with an equivalent Kalman filter provide meaningful information: while the proposed set-membership approach tends to provide more conservative results in the ideal case, it confirms its robustness in the more realistic case of not perfectly tuned motion model (see Section 7.4.1).

Some potential advantages and some challenges of the proposed method have been outlined throughout the work. Nonetheless there are still many unanswered questions and a lot of research to be done, in order to devise efficient algorithms working in real-world situations. Below, some suggestions for future research are provided.

- **Feature extraction**
  Any feature-based approach to navigation depends heavily on correct and efficient feature identification. This is an extremely important area of research. The use of additional information (such as colors or textures) along with classical measurements (such as distance and relative orientation) may lead to easier identification. Also adaptive strategies in moving and sensing could improve feature detection and extraction.

- **Data association**
  Associating measurements with features in cluttered environments is another crucial point for any estimation problem that faces data association ambiguities. The use of
sensors providing several kinds of information on features can reduce such ambiguities. Moreover, the proposed set-membership approach can be useful to determine ambiguous measurements. Let us consider the $i$-th measurement at time $k$. This measurement can be associated to the $j$-th feature only if the intersection between the uncertainty region associated to that feature (a single point if the map is perfectly known) and the uncertainty set associated with the $i$-th measurement (sum of the uncertainty associated with the robot and that associated with the sensor itself) is not empty. Consequently, the group of ambiguous measurements can be easily determined. When tackling the localization problem (i.e. a map of the environment is available), the proposed strategies can also help to detect wrong measurements (false detection, outliers): if any measurement uncertainty set has empty intersection with each feature uncertainty set, then that measurement is wrong.

The problem of data association gets harder when the SLAM problem is considered. In this case the location of features is not known \textit{a priori}: special efforts have to be spent to distinguish between new features detections and wrong measurements. Moreover, data association can become tricky, particularly when moving features are considered. In addition, one must notice that the computational burden required by sets intersection is generally high, so the problem should be carefully investigated.

A special case of data association is landmark matching on different maps, needed to successfully perform map fusion in CLAM tasks. As a matter of fact, even a single wrong associations on the 1D maps may lead to completely wrong 2D maps. An efficient method for associating features on different maps would allow for more decentralized algorithms, where each robot could operate more independently, activating the map fusion algorithms only at some determined time instants.

- \textbf{More complex estimation problems}
  Estimation of robot orientation has been considered only for the case of perfectly known environments. When considering simultaneous localization and map building, it is easy to verify that the uncertainty on robot orientation can considerably affect the dimension of the uncertainty sets of far features, since the observed features belong to a set roughly resembling a sector of circular corona: its radii are determined by the measurement, while its angular width is determined by the robot orientation uncertainty. Special strategies should be developed to deal with this problem.

- \textbf{Robust path planning}
  The proposed estimation method provides sets that are guaranteed to contain the
true values. Then, it may be appropriate to develop some path planning techniques based on such sets, in order to provide robustly secure paths. This implies that all the possible paths, when the robot is affected by disturbances satisfying the UBB hypothesis, would be collision free. Research work on this topic is currently carried out.
Bibliography


[84] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping


identification with set membership uncertainty. *IEEE Transactions on Automatic

[88] E. Walter and L. Pronzato. Characterizing sets defined by inequalities. In *Proc. of
IFAC Symposium on System Identification*, pages 15–26, Copenhagen, Denmark, July
1994.


