POLITECNICO DI MILANO
Corso di Laurea in Ingegneria Informatica
Dipartimento di Elettronica e Informazione

Scan matching covariance estimation and SLAM: models and solutions for the scanSLAM algorithm

AI & R Lab
Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

Advisor: Eng. Matteo Matteucci
Co-Advisor: Prof. Basilio Bona
Co-Advisor: Eng. Davide Migliore
Co-Advisor: Eng. Simone Ceriani

Tesi di Laurea di:
Mauro Brenna, student id 720542

Academic Year 2008-2009

To my parents, Roberto and Laura

Sommario

Questa tesi si colloca nel contesto della robotica mobile, più precisamente affronta il problema denominato Simultaneous Localization and Mapping (SLAM), ovvero quello di stimare contemporaneamente la posizione del robot, (x, y, θ), e la mappa dell'ambiente utilizzando i dati rumorosi forniti da sensori, quali encoder, laser e sistemi di visione.

Questo problema si rivela essere di fondamentale importanza per la creazione di agenti fisici realmente autonomi. Il risultato degli algoritmi potrà essere infatti utilizzato come base per algoritmi decisionali di navigazione quali path planning e obstacle avoidance.

Il lavoro si concentra su un particolare algoritmo di SLAM, scanSLAM basato sulla tecnica di filtraggio denominata Extended Kalman Filter e sull'utilizzo di algoritmi di scan matching.

Il modello di scanSLAM è stato scelto poiché si rivela essere una interessante alternativa a metodi come il Rao-Blackwellized Particle Filter, per la creazione di mappe basate su dati provenienti da scansioni laser. Questo tipo di sensore, comune in molte applicazioni robotiche, è anche uno dei principali sensori utilizzati dell'ambito del progetto LURCH, riguardante lo sviluppo di una carrozzina autonoma. Essendo ancora sprovvista di sistemi di navigazione SLAM, la carrozzina è stata considerata come principale target per questo lavoro di tesi.

Lo scopo della tesi è quello di proporre estensioni al modello scanSLAM, nuovi o tratti da differenti algoritmi di SLAM, con particolare interesse alle tematiche di self calibration e dell'utilizzo di laser multipli, nonché la loro implementazione e testing. Questo lavoro può essere considerato il primo caso di testing dell'algoritmo con dataset reali, ottenuti dai progetti Rawseeds e LURCH, e per il quale sono analizzate sperimentalmente le tecniche per la stima della covarianza relativa all'errore della famiglia di scan matcher denominata Iterative Closest Points, ICP.

La scelta di estendere il modello al problema della self calibration deriva dall'esigenza di stimare in modo accurato i parametri di rototraslazione tra

il sistema di riferimento laser e il centro odometrico della carrozzina del progetto LURCH e di aggiornare on-line tali parametri per assicurare la robustezza dell'algoritmo di SLAM. Pochi articoli scientifici hanno trattato la calibrazione on-line di questi parametri estrinseci stimando simultaneamente la mappa dell'ambiente.

L'interesse verso un approccio multi sensore deriva dalla opportunità di impiegare più laser range finder di fascia medio bassa al fine di ottenere una mappa più accurata dell'ambiente e limitare i problemi derivanti dall'ambiguità delle scansioni laser.

Con l'obiettivo di implementare la versione più corretta possibile per il modello di scanSLAM, è stata condotta un'analisi dei più promettenti metodi di scan matching (ICP) e degli algoritmi che permettono di stimare la covarianza dell'errore di minimizzazione. Una approfondita serie di test ha permesso di individuare la soluzione migliore.

Si è infine realizzata una libreria generica e riusabile per la manipolazione dei dati acquisiti da laser range finders, per algoritmi di EKF e scanSLAM che potrà essere utilizzata nel progetto LURCH per la progettazione futura di alcune varianti proposte in seguito.

Benché nella fase di testing i risultati ottenuti in relazione alla self calibration non siano ottimali e malgrado l'algoritmo non converga alla soluzione esatta, le possibili fonti di errore sono state indagate e sono state proposte alternative al fine di migliorare l'algoritmo.

Il lavoro di tesi, corredato dall'analisi dei risultati di testing, rappresenta il primo vero approfondimento delle tematiche legate al modello scanSLAM che si rivela costituirsi una interessante alternativa alla soluzione del problema di SLAM mediante scansioni laser.

Acknowledgements

There are people without this master's thesis would have not been possible. To whom I give all my thankfulness and acknowledgements. In particular, I would like to thank my advisor, Eng. Matteo Matteucci, and the co-advisors Eng. Davide Migliore, Eng. Simone Ceriani and Prof. Basilio Bona.

I will remember with pleasure our meetings at AIRLab in Milan, and the VoIP meetings with Davide.

A special acknowledgement to Elisa Maira, for patiently checking the English draft of this work and for supporting me during these years of univesity.

To all the people who study or work in AIRLab laboratory, for the days spent together and the pieces of advice they gave to me every day: Davide Rizzi, Matteo Rossi, Marco Assini, Martino Migliavacca, Paolo Villa, "I Pesci" (Dario Simontacchi, Marco Perri and Andrea Parolina), Mattia Pirotti and all the others. To all my university friends of Politecnico di Milano and Politecnico di Torino, Andrea Scotti and all the ones who I met during my studies.

Thanks also to Andrea Ratti and Federico Dal Castello for their moral and technical support. To them I will send a special, *humoungous* and never *trivial*: "Go!".

Contents

Chapter 1

Introduction

> *"Everything is theoretically impossible, until it is done. One could write a history of science in reverse by assembling the solemn pronouncements of highest authority about what could not be done and could never happen."*
>
> Heinlein, Waldo & Magic, Inc. (1950)

This final dissertation deals with a specific area of the mobile robotics named probabilistic robotics. Mobile robotics is a specific field of robotics which concerns the problems associated with the movement of an autonomous physical agent, namely a robot, which interacts with the environment. Probabilistic robotics employs robust techniques derived from statistics to solve the problems of localization, mapping, path planning and more in general navigation and exploration of an environment.

Among these topics, a very fundamental one is named Simultaneous Localization and Mapping (SLAM) which concerns the problem of correctly estimating the location of the robot (the pose of the robot) and the map of the environment given the noisy data acquired by robot sensors, e.g., wheel encoders, laser range finders and digital cameras.

This work is focused on a particular SLAM algorithm, named scanSLAM, which uses raw laser sensor scans and scan matching algorithms in an Extended Kalman Filter (EKF) framework. This model is chosen because it constitutes an interesting alternative to common approaches for laser-based SLAM such as Rao-Blackwellized Particle Filter.

The laser range finder is a common sensor employed in contexts of autonomous robotics, and it is also one of the main sensors in use in the LURCH project, which regards the development of an autonomous wheelchair. This wheelchair, lacking of a proper software for SLAM, configures itself as the ideal target application for this work.

The aim of this thesis is to extend the scanSLAM model, using novel methods and techniques already employed in similar contexts, with particular interest to the themes of self calibration and multiple lasers. To the modelling phase follows the implementation and testing of the proposed models. In order to estimate the covariances needed for the Gaussian models of the EKF, a family of scan matching algorithms, which purpose is to align a pair of laser scans, is considered. Moreover, an extensive research and analysis of the state of the art algorithm in ICP covariance estimation is conducted.

The results of the testing phase shows that the scanSLAM algorithm, implemented with the proposed modifications, gives optimal results both using very accurate laser range finders such as SICK, and with lower accurate ones (e.g., Hokuyo). From visual inspection it can be observed that the maps of the environment are consistent and that can be employed, in principle, as input to algorithm for path planning. Moreover, an analysis of the outcomes is performed to investigate the sources of errors for the self calibration extension proposed, which does not retrieve the optimal solution.

ScanSLAM is first conceptualized as an example of Pose Snapshot Kalman Filter (PSKF) by Nieto et al. [59, 60]. Other authors such as Diosi and Kleeman [25] and Bosse [14] proposed the same model and extended it.To the author's knowledge, no researcher has extensively tested the standard scanSLAM algorithm with a real dataset. Moreover, the scientific paper by Bosse [14] is based on an extension and uses it for building local maps only.

One of the main components of the scanSLAM filter, beside EKF itself, is the scan matcher. This kind of algorithm retrieves the optimal geometric alignment between two laser scans (or between one scan and a reference shape), which are defined as a cloud of 2D or 3D points. Among the different scan matching techniques, the Iterative Closest Points family is the most used for its robustness and speed. In particular, three ICP algorithms are taken into consideration: the Classic ICP, the Metric Based ICP and the Point-to-Line ICP. The first method is derived from the works by Lu and Milios [45], Eggert [28] and Rusinkiewicz et al.[63], and it is based on the simplest distance metric, the Euclidean distance between points of the two scans. The second method is based on the scientific paper by Minguez et al. [52, 53] and considers a particular distance norm which takes into account the rotational components between the two scans. The last method is based on Censi's PLICP [19] and it is interesting for the speed of convergence and the minimization procedure in use.

As mentioned before, the estimation of the uncertainty associated to scan matching is needed to correctly implement the Extended Kalman Filter. In

the literature, few methods have been proposed. The most representative ones are: [44] by Lu and Milios, [5, 6] by Bengtsson and [19] by Censi. In particular, a good covariance estimation is needed also in peculiar under-constrained conditions such as corridors where the scan matching algorithm is very uncertain along one specific direction.

The three scan matchers have been realized and implemented in C++ together with the main algorithms for the estimation of the ICP covariance to create a generic software library for the manipulation of laser scans . A detailed and extensive analysis is performed to test the best combination of scan matcher and covariance estimation algorithm. The resulting best couple has been used to implement a generalized scanSLAM algorithm employing a novel generic, template-based, Extended Kalman Filter for SLAM. The results obtained are encouraging and promising. The scanSLAM algorithm is tested both with high cost, precise, laser range finders (SICK) and with low cost, low-ranged ones (Hokuyo). The generalized algorithm is then extended to deal with the issue of self calibration. The problem is to estimate the correct rototranslation between the laser and the robot reference frames. This problem is particularly interesting in relation to LURCH project for which the extrinsic calibration is manually performed through ad hoc methods. In this case, the scanSLAM algorithm could improve the robustness of the resulting map thanks to the on-line estimation of the extrinsic parameters. In this case, the results are less promising but an analysis is conducted to find the possible sources of error. This can be considered as the first work in the literature which regards to extensively test the scanSLAM and which shows the results obtained with a real dataset using the recent advancements in the estimation of ICP covariance.

Many improvements can be the subject for future research and developments. In particular, the implementation of the extension described in this work to deal with multiple lasers can successfully developed for enhancing the scanSLAM performance in relation to the autonomous wheelchair built by the AIRLab, Artificial Intelligence and Robotics Laboratory, of Politecnico di Milano under the LURCH project also. The procedure of self calibration can be improved by testing different probabilistic robot motion models and combine the information gathered by other localization algorithm in the EKF filter. Lastly, an algorithm to collapse the dimension of the EKF state variables by merging multiple landmarks constituted by scans is interesting to deal with very large maps, such as Rawseeds datasets [23].

The thesis has the following structure:

- In Chapter 2, the state of the art is reviewed in order to make the

reader familiar with the best results in scan matching, filtering, SLAM and self calibration procedures

- In Chapter 3, some implementations of the scan matching algorithms are described. The algorithms for estimating the ICP covariance are also shown together with the algorithm in pseudocode and the procedure for scan preprocessing and outlier rejection
- In Chapter 4, the Extended Kalman Filter for SLAM is reviewed and the motion model used for the implementation is described as well
- In Chapter 5, the main generalizations and extensions, novel or derived from other SLAM models, of the scanSLAM algorithm are explained. In particular, the model is generalized to take into account the uncertainty in the rototranslation between the laser and the robot reference frames. A self calibration method based on the EKF state augmentation is derived together with an extension to deal with multiple lasers and the policies for improving the step of data association
- In Chapter 6, the two dataset used and the related projects, Rawseeds and LURCH, are described. The two laser range finders used for the testing, Hokuyo and SICK, are also presented
- In Chapter 7, the software library is described from an architectural point of view. An analysis of the ICP covariance is performed, followed by the description of the results for scanSLAM and the extension of self calibration
- Chapter 8 is dedicated to the conclusions and suggestions for future research
- Four appendix are included in this work. Appendix A shows the common mathematical notation used in this work.Appendix B describes how to compute numerically the Hessian matrix associated with the ICP error, which can be used to estimate the covariance. Appendix C shows a simple implementation of the Kalman Filter, which is used to validate the generic software implementation. Appendix D defines the Mahalanobis distance and shows a simple example of usage.

Chapter 2

State of the Art

"There's no limit to what he can do. He could destroy the earth... If anything should happen to me you must go to Gort, you must say these words, "Klaatu barada nikto", please repeat that."

The Day the Earth Stood Still (1951)

The present thesis deals with different themes that are pervasive in mobile robotics, in particular it refers to the topics of scan matching, SLAM and autocalibration. In this chapter, the scientific progresses in the various fields are shown with the aim of making the reader aware of the most recent techniques and explaining how this work could be placed within the accademic literature.

2.1 Scan Matching

2.1.1 Iterative Closest Points

The term *scan matching* refers to a family of algorithms that solve the problem of misalignment of two lasers scans, or of a scan and its parametric model. The *scan matcher* uses the two scans as inputs and returns the rototranslation allowing to superimpose the second scan to the first. The present dissertation is focused on particular algorithms of this family named ICP, Iterative Closest/Corresponding Points.

Besl and Mc Kay [7], in 1992, created the ICP algorithm for the registration of 3D images, demonstrating the monotonic convergence to the nearest local minimum. In the article, they examined the case of a 3D cloud of points with respect to a parametric model which is given as polylines, set of triangles or curves.

In 1997, an article by Lu and Milios [45], showed one of the first applications of the ICP method for mobile robotics. The authors applied a 2D adapted method to solve the problem of localization with laser range scans. In fact,it is difficult to use landmarks techniques, e.g., detection of corners, for laser scans because the algorithm would need to solve an object recognition problem.

Two frequently cited algorithms were shown in that article. The first one exploited the tangent directions and it is regarded to be fast, while the second, slower but more stable, is the classic ICP point-to-point solved by least square technique. Both the methods could be employed in sequence and constituted the core of IDC, the Iterative Dual Correspondences.

Rusinkiewicz et al. [63] classified the most common variants of ICP algorithm, verifying the difference in the speed of convergence. They concluded that the point-to-plane metric, together with a projection-based corresponding points research, assured the most reliable result and a faster convergence than the classic point-to-point ICP metric. Furthermore, Eggert et al. [28] explained how it is possible to compute the rototranslation that alignes two set of corresponding points thanks to closed-form methods. Among the others, the method based on singular value decomposition (SVD) achieved the best overall results in terms of stability and accuracy.

Because of the intrinsic noise in the measures and robot motion, the laser scans could not be completely superimposed and, for this reasons, particular techniques are needed in order to eliminate outliers in the set of correspondences and so reduce the bias in the rototranslation. To this extent, two articles can be cited: the one by Chetverikov et al. [24] who proposed a robust version of ICP named Trimmed ICP and the one by Zhang [81], who showed an adaptive method for outlier rejection.

Since its first appearance, many variants of ICP algorithm have been proposed in the literature, with the aim to improve its robustness and speed. Censi (PLICP) [19], demonstrated that it is possible to obtain a method based on a point-to-line metric, which uses the information provided by the normal to the surface, which converged quadratically in a finite number of steps with a closed-form minimization. Minguez et al. (Metric Based ICP) [52, 53] considered a minimization metric - different from the common euclidean norm - which permits to take into consideration both the translational and rotational components ensuring higher robustness. Conversely, Diosi and Kleeman [25, 26], tried to increase the speed of convergence minimizing the time spent in the search of correspondences. To obtain this result,

the authors used the polar coordinates[1] proposing the Polar Scan Matching (PSM) algorithm, which resulted faster in comparison to the classic ICP and ensured a larger basin of attraction.

Other approaches have been successfully proposed in the literature: Histogram Matching (Bosse et al. [13]), which exploited two histograms, one for the orientation and one for the translational component; the scan matching methods, often named correlation-based, based on the use of one or more grids (look-up tables) as in Konolige et al. [42] and Olson [61]. Other authors have investigated evolutive techniques, such as genetic algorithms: Lenac et al. [43] (GLASM) used a binary look-up table and the Gray code to compute the chromosome distance, while Ze-Su et al. [78] (GPSM) exploited the genetic algorithms to refine the solution obtained with Polar Scan Matching. It is worth to highlight the existance of methods such as Yoshitaka et al.'s [77], which exploits other information acquired by the laser sensors such as the intensity of laser reflection with the aim of improving the result of scan matchers based on to geometric properties only.

Finally, particular attention must be drawn to the efforts in formulating the scan matching algorithm under the probabilistic framework carried on by Biber et al. [8], Montesano et al. [54], Censi [16],Nieto et al. [60] , Burguera et al. [15], Segal et al. [66]. In these papers the authors approximated the scans using sum of gaussians (SoG) or other probability density functions or, in some cases, they modelled probabilistically a step of ICP algorithm.

2.1.2 Error Covariance Estimation

In this thesis, the information obtained by the scan matching algorithm is used to create a consistent map of the environment and determine the correct pose of the robot. For this reason, an important step in order to include the scan matcher in a probabilistic filtering framework is constituted by the correct estimation of the error made by the algorithm. In particular, the estimate of the covariance matrix is needed in the SLAM (Simultaneous Localization and Mapping) algorithm to fuse the estimate with the other sensor ones, to obtain a proposal distribution for particle filters, or to weight constraints in a pose-graph [17]. Apparently, few researcher have dealt with this issue since the introduction of the ICP algorithm.

Bengtsson et al. [5, 6] examined the seminal works by Lu [44] and proposed two new methods to estimate the covariance matrix related to the position estimate (the examples are based on the IDC algorithm). Similarly to Bengtsson, Bosse et al. [13] derived another form of ICP covariance for

[1] see also Censi's contribution in [19] appendix

their own scan matching algorithm (that has the same error metric of Censi's one [19]).

Censi studied in detail the problem of the achievable accuracy for range-finder localization and pose tracking [18, 20] and the limits of ICP covariance estimation methods. In [17], he employed a general formula already used in statistics and computer vision to estimate the covariance matrix.

All these methods are considered in Chapter 3, while and sperimental analysis is conducted in Chapter 7.

It should be noted that in every scientific paper holds the assumption that the ICP algorithm is not trapped in a local minimum. This means that if the scan matcher converges to a wrong relative position, the covariance matrix could be very optimistic.

2.2 Simultaneous Localization and Mapping

The expression *Simultaneous Localization and Mapping* refers to the collection of techniques and methods that can be successfully employed to solve the problem of using the information acquired by the robot, odometry and sensor readings, to estimate consistently both the pose[2] of the robot and the map of the environment. These techniques combine two fundamental problems of autonomous robotics: localization and mapping. Namely, the first problem deals with the estimation of the robot pose given an almost perfect map, while the second problem is dual to the first one and consists in retrieving the map of the environment taking the exact robot pose for granted at every time step.

2.2.1 Incremental Mapping

In incremental mapping the local maps, e.g., the laser range finders readings, are incorporated into the global map using a greedy approach. When the laser scan is acquired, the algorithm computes an expectation maximization step retrieving the most probable pose. After this step, the current scan is included permanently in the map using that estimate. Once in the map, the scans are indistinguishable and no other probabilistic mechanism can be used to move or delete the scan if something wrong occurrs.

Thrun et al. [69] presented a probabilistic method for mapping combining an Expectation Maximization (EM) approach with the Maximum

[2]The term pose in contrast with the term position comprises the whole set of variables that define the location of the robot in the environment. For a 2D autonomous robot, this has 3 degree of freedom (x, y, θ)

Likelihood estimators of incremental mapping. In order to cope with the possible inconsistence they used also a backward correction procedure.

Gutmann et al. [35] took another approach to incremental mapping: the last n-scans were correlated to check the consistency of the map in two phases. At the beginning, the latest scans are aligned, then, after few scan acquisitions, a loop detection procedure is used to align the local maps.

Hähnel et al. in [37], and extensively in [36], elaborated a probabilistic algorithm to deal with populated environment. The approach is based on the determination of the most likely robot pose by the use of scan matching technique together with a filter to detect and track people in the vicinity.

2.2.2 Constraint Network Optimization - GraphSLAM

A different approach towards the SLAM problem is given by relation-based SLAM. In this family of algorithms the pose and map are internally represented as a graph where the edges are built from particular constraints between each vertex.

The work by Biber et al. [9] can be easily classified under this framework. The authors proposed a technique to perform multiple scan matches simultaneously and how that method could be employed in a relation-based SLAM similarly to Lu et al. [44]. The scan matching procedure is based on the minimization of an energy function and the resulting pose estimate is employed in a graph where the nodes are the laser scans and the edges are the pairwise relations between them.

Bosse et al. [11, 12] developed Atlas, a hybrid metrical/topological approach to SLAM designed for large-scale environments. In this case, the graph was defined as following: the nodes represented the local frames and the edges the transformations between them. Within each frame a local map was attached and a map matching algorithm was implemented in order to close the loops. The framework could be used both with local maps based on features or laser scan matching.

Iser el al. [39] described another scan matching application to topological mapping. The scan matcher used employed a sampling approach followed by an ICP refinement step. While the covariance was computed thanks to the resulting sampling distribution. Moreover, a RANSAC approach for loop closure was implemented as well.

Grisetti et al. [33, 34] developed TORO (Tree-based netwORk Optimizer), an efficient on-line optimization algorithm that fixes the issues in graphSLAM by means of a stochastic gradient descent and by the use of a tree-based parametrization of the network. Since it is assumed that the

constraints to the graph are given, it can be effectively used in couple with the Atlas framework for obtaining such data associations. Extensions to this methods also permitted the solution of the problem of 3D SLAM as well.

2.2.3 EKF-SLAM and ScanSLAM

EKF-SLAM is an acronym for Extended Kalman Filter SLAM, an approach to the SLAM problem that uses the well established mathematical framework given by Kalman filtering techniques. Kalman filter [75], was invented by Swerling and Kalman [41] as a technique for filtering and prediction in linear Gaussian systems [70]. The methods employed by Kalman were extended to non linear system using different techniques such as Taylor linearization or unscented transform and adapted for autonomous robots. The seminal work by Smith et al. [67] in 1990, gave input to further pubblications on this subject. For a good introduction to the state of the art in SLAM with particular emphasis to EKF-SLAM and particle filtering see Durrant-Whyte and Bailey's reviews [27, 4]. Another good introduction is given by Sasiadek et al. [65, 64], while a detailed example of 3D feature-based EKF SLAM algorithm is described by Blanco [10].

ScanSLAM is a generalization of EKF SLAM that deals directly with laser scans, without relying on geometric models (Feature based EKF). This approach can be seen as an example of Pose Snapshot Kalman Filter, PSKF [14]. Nieto et al. [59, 60] proposed under the name of scanSLAM the general algorithm in which landmarks (features) are directly defined by the raw sensed data and the robot pose when these data were acquired whereas, the result of the scan matching algorithm was used as a measurement in EKF. The authors showed also a sampling technique to compute the ICP covariance matrix and a scan segmentation procedure to augment the landmark templates. In the paper published in 2005, [59], the scan matcher was based on a probabilistic representation of the scans as Sum of Gaussian (SoG) which was explained in details in Bailey's PHD dissertation [3].

In the same period also Diosi and Kleeman worked on the same subject [25]. In their technical report, they explained the scanSLAM approach for their Polar Scan Matcher drawing inspiration from the Atlas work by Bosse et al.[12]. The implementation details are essentially the same as in Nieto et al.'s work where a landmark was updated only if the predicted robot pose was sufficiently close to it. Extension to multiple laser scanSLAM was also presented for simulated environment fusing the informations of the lasers before incorporating the scan in the landmark [71]. In this case, the rototraslation between the reference frames of the robot and the laser sensor

was not considered in the filtering phase.

The authors do not show a complete example of the their results using real datasets. Thus, estensive tests are performed in Chapter 7 for the standard scanSLAM algorithm, putting in evidence the strenghts and the limitations associated to the use of this approach. Moreover, the works written by Nieto [59], and Diosi et al. [25] do not consider the case of the generic rototranslation between laser and robot reference frames, which is examinated in this work in order to develop a model for the self calibration of the extrinsic parameters.

Recently, Bosse and Zlot in 2008 [14], presented an extension to plain scanSLAM algorithm. In their implementation, local maps are computed using EKF scanSLAM while at a global scale they exploited a fast map matching algorithm which was able to align overlapping local maps without an initial guess.

2.3 Self Calibration

The calibration process is a necessary step in order to correctly identify and tune the different parameters that affect the motion of the robot and the correspondences between different reference frames. In the literature it is possible to distinguish two types of calibration: intrinsic calibration, which regards the odometry motion models and some sensor parameters, and the extrinsic one, which refers to the transformations between different reference frames. In particular, "the odometry calibration consists of the identification of a set of kinematic parameters that allow reconstructing the vehicle's absolute position and orientation starting from wheels' encoder measurements"[2]. A procedure for on-line tuning of the parameters can also be called autocalibration while when it is performed together with a localization or SLAM algorithm it is defined also as Simultaneous Localization and Auto Calibration (SLAC).

Eliazar et al. [30] defined a generic motion model that takes into account a lateral translation term. In this way, they were able to model shifts in the orthogonal direction to the major axis, while the parameters are estimated using a particle filtering SLAM implementation, namely DP-SLAM[3]. Antonelli et al. in [2], described a systematic method for odometry calibration of differential drive robots. The unknown parameters in the kinematic equation (the wheelbase that connects the two lateral wheels and the two radii of the wheels) were linear and, in this case, a least-squares method could be

[3]http://www.cs.duke.edu/ parr/dpslam/

applied. The main advantage of the formulation is the possibility to assign a quality measure to test trajectories.

Martinelli et al. [50] investigated the theoretical results that can be achieved by on-line odometry self-calibration and tried to define the best trajectory that maximizes calibration accuracy. In [49, 51], they proposed the use of two coupled Kalman Filters in order to perform SLAM and simultaneously estimate the odometry parameters. The first Augmented Kalman Filter accomplished the estimation of the systematic component of odometry error while a second KF estimated the non-systematic component. In [48, 47], Martinelli considered the particular problem of self calibration by using a single point of feature, determining the extrinsic parameter of a bearing sensor and of the odometry.

Another approach to autocalibration is studied by Censi et al. in [21]. In the paper, they described a method based on the solution of a constraint least-square problem to simultaneously calibrate the odometry parameters (wheelbase and radii) and the extrinsic parameters that described the roto-translation between a laser sensor with respect to the robot frame.

A more specific method is described in [80, 79] where Zhang et al. described a procedure to calibrate together a digital camera and a laser sensor in order to determine the transformation between two sensor frames. The solution is given by a non linear minimization problem applied on an epipolar constraint between the two sensors. The experiments are carried on using both an apriori known environment (e.g., a chessboard) and an indoor unknown one.

Chapter 3

Scan Matching

"I visualize a time when we will be to robots what dogs are to humans. And I am rooting for the machines."

Claude Elwood Shannon, Omni Magazine 1987

In this chapter, the *Scan Matching* algorithms, which have been implemented in the software library, are thoroughly explained in order to describe their main characteristics and weaknesses. As mentioned in the previous chapter, the problem of scan matching consists in finding the optimal geometric alignment between two laser scans (or one scan and a reference shape), which are defined as a cloud of 2D or 3D points. As a necessary condition, the two scans must be *matchable*, that is, they must partially represent a common shape.

A lot of scan matchers have been described in the scientific literature: many are *grid-based*, that is based on discrete maps, while others, *ICP, Iterative Closest Point*, are based on the iterative alignment of corresponding points. In this chapter, ICP algorithms will be taken into consideration because they are commonly preferred for their efficiency, precision and large use. In particular, the ICP scan matchers implemented in the software library are: Classic Iterative Closest Point [63, 28], Metric-Based ICP [52, 53], Point-to-Line ICP [19].

3.1 General Model

Despite the apparent diversity, every scan matching algorithm can be represented in a uniform way since it solves the same problem, see Figure 3.1, 3.2. The black box model can be thought as follows:

$$[RT, Cov] = scanMatch(referenceScan, currentScan, RT_{guess}) \quad (3.1)$$

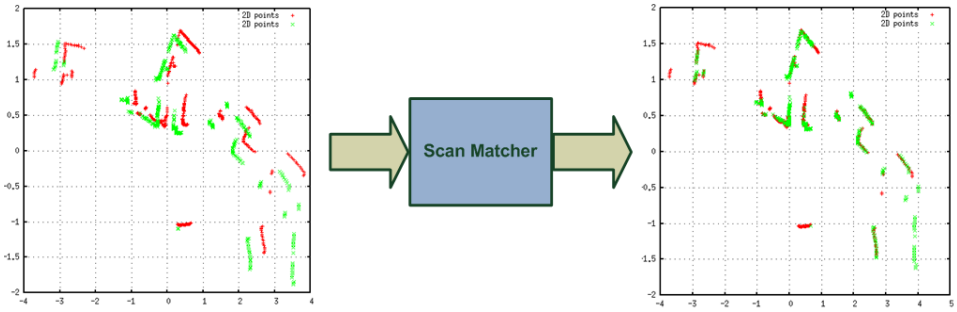


Figure 3.1: Black box model of the scan matching algorithm

Where:

referenceScan, currentScan are the laser scans obtained by a laser. The points can be represented using Cartesian, $\{x_i, y_i\}$, or polar, $\{range, angle\}_i$, coordinates. referenceScan is used as a fix reference scan while currentScan will be rototranslated to be aligned to the first one.

RT is the transformation, rototranslation, such that applied to currentScan aligns it to referenceScan. It can be expressed in matrix or vector form (see appendix A). In the terminology used in this document the transformation is referred to as $RT^{referenceScan}_{currentScan}$. RT_{guess} is the transformation used as the initial guess by the scan matcher.

Cov is the covariance matrix associated to the error in the minimization procedure.

In order to optimally solve the problem, the algorithm minimizes a figure of merit that models the alignment error:

$$E_{dist}(RT) = argmin \sum_{i=1}^{l} distance(p_i, RTq_{c_i})^2 \tag{3.2}$$

Where:

$\mathbf{p}_i$ $\in$ referenceScan;

$\mathbf{q}_k$ $\in$ currentScan;

(i,c$_i$) represents the indexes of corresponding points between the two scans;

distance is a mathematical distance, typically the Euclidean one ($norm_2$).

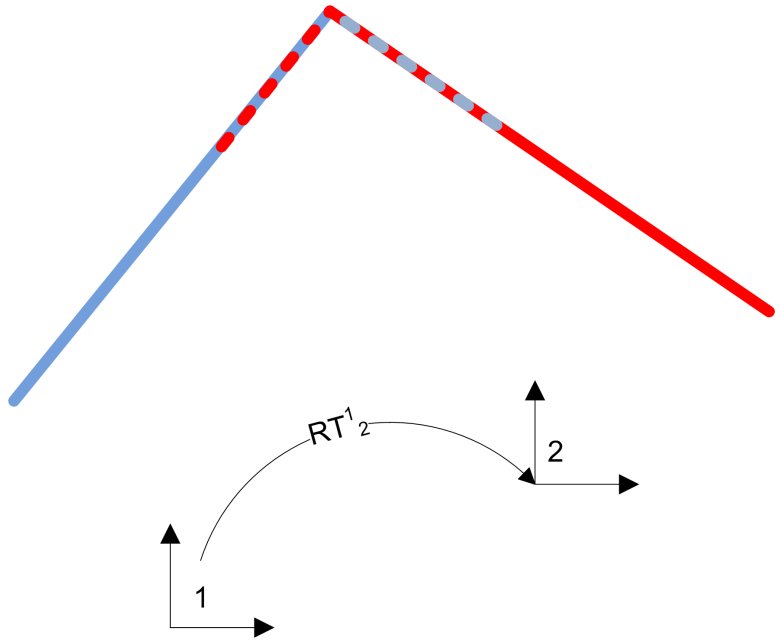


Figure 3.2: Geometric transformation involved in the scan matching algorithm

The fundamental role of (i, c_i) in the minimization procedure explains the reason why ICP is also called Iterative Corresponding Points. in 2001, Rusinkiewicz [63] built an interesting schema that frames every kind of ICP algorithm. Each iteration includes the following six steps:

Selection of interesting points in both the scans (meshes). The strategies range from the selection of all the points to sampling procedures in order to increase the significance of the chosen clouds of points.

Match the points in both the scans looking for corresponding pairs. This could be done using a closest point technique or by projecting the points on the segments constituted by points ot the reference scan.

Weight corresponding pairs. Strategies are used for error prevention since very distant corresponding pairs can provoke a bias in the estimation of the final transformation.

Reject certain pairs in order to minimize/eliminate outliers. The strategies used in the implementation are explained later in this chapter.

Algorithm 3.1.1 ICP General Model

```
[RT, Cov] = scanMatch(referenceScan, currentScan, RT_guess)
RT = RT_guess
begin (Repeat till convergence)
  tempScan = transform currentScan using RT
  Find correspondences between referenceScan and tempScan
  Delete otuliers from the set of correspondences
  RT = E_dist(RT) = argmin sum_{i=1}^{l} distance(p_i, RTq_{c_i})^2
end
Compute the covariance Cov
```

Assign an error metric. A distance must be defined between corresponding points or for the whole scan (see scan matchers in the probabilistic frameworks, Chapter 2).

Minimize the error metric using a closed-form or an iterative procedure (e.g., conjugate gradient).

Algorithm 3.1.1 shows a simplified general model of the ICP algorithm which describes the structure of all the algorithms presented in this chapter. In particular, the rejection of outliers is defined as a single step and the pairs of corresponding points are evenly weighted. The selection step is performed both in the acquisition phase, filtering the laser error codes and the out-of-range, and in the outlier-rejection phase.

3.2 Scan Matching Algorithms

3.2.1 Classic ICP

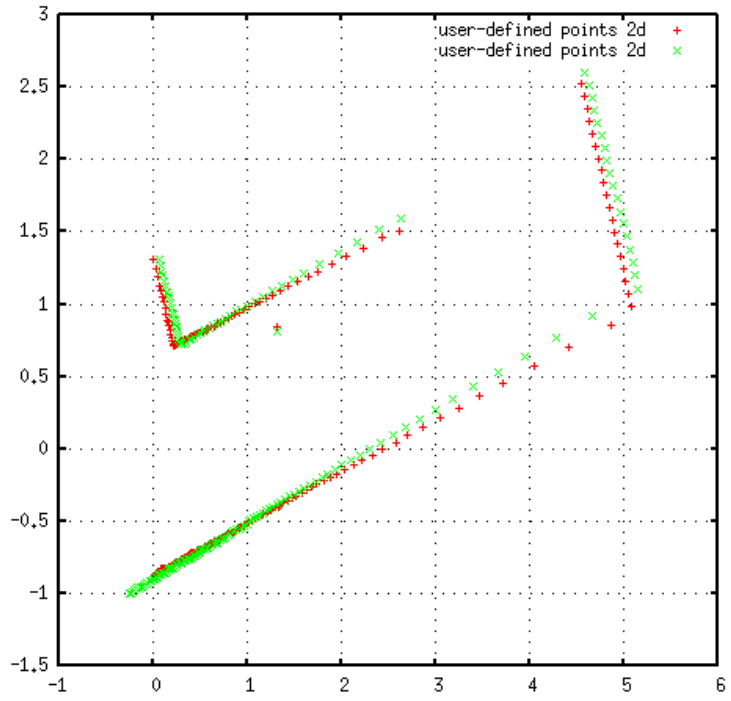
The first algorithm implemented in the software library is the Classic Iterative Closest Point. The version of the algorithm proposed in this work is the standard version, mutuated from different scientific papers [45, 28, 63], with some tweaks for improving stability and convergence. Using the scheme showed in Algorithm 3.1.1, this section and the following describe only the distinctive features of each implementation. The Classic ICP shares the most intuitive figure of merit:

$$E_{dist}(RT) = argmin \sum_{i=1}^{N} ||p_i + RT * q_{c_i}||^2 \tag{3.3}$$

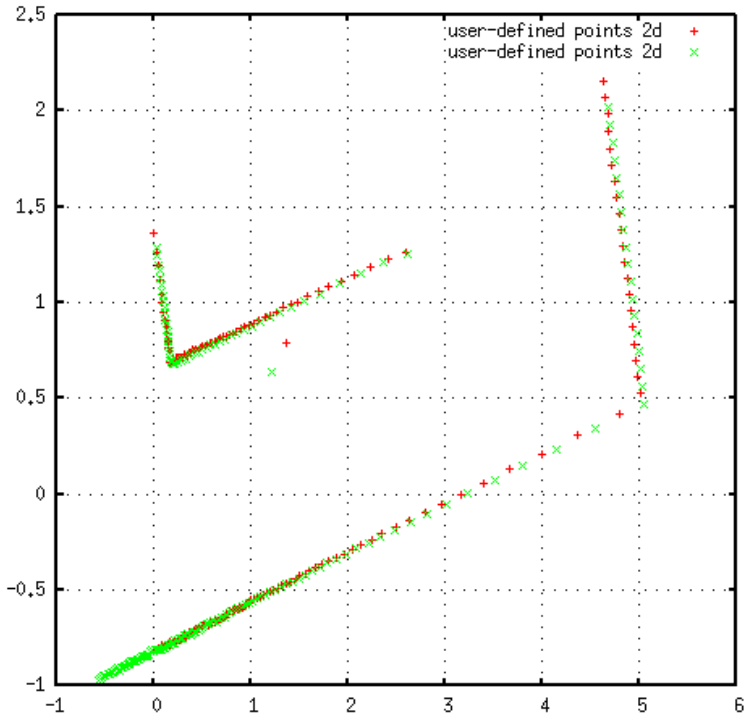
Algorithm 3.2.1 Classic ICP: 2DRototranslationBySVD

[RT] = 2DRototranslationBySVD(Pscan, Qscan)
$\bar{p} = average(Pscan)$
$\bar{q} = average(Qscan)$
$Pscan' = Pscan - \bar{p}$
$Qscan' = Qscan - \bar{q}$
$H_{2x2} = \sum_{i=1}^{N} p'_i * q'^T_i$
$H = U * S * V^T$
$R = V * U^T$
if (det(R) is -1)
  $V' = [v_1, -v_2]$, where $v_i \in M_{2x1}$
  $R = V' * U^T$
endif
$T = \bar{q} - R * \bar{p}$
$RT = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$

This function simply computes the Euclidean distance between each couple of corresponding points. A common improvement to this error metric is to consider the distance between each point in the current scan and the projection of the same point on the surface built from the reference scan using a polyline approach. This choice should reduce the bias in the estimate of the rototranslation RT (see Figure 3.3). In order to compute the 2D rigid transformation between the two set of points, the Classic ICP relies on an SVD solution described by Eggert et al. in [28] and showed in Algorithm 3.2.1 where the two set of points are considered already ordered (p_i corresponds to q_i where $q, p \in \mathbb{R}^2$).



(a) Metric Based ICP



(b) Plane-to-Line ICP

Figure 3.3: These images are the result of two different scan matchers. The laser sensor used is a SICK and the scale of the plot is in metres. Censi's Point-to-Line method converges to the right minimum because it is based on the computation of normals to segment

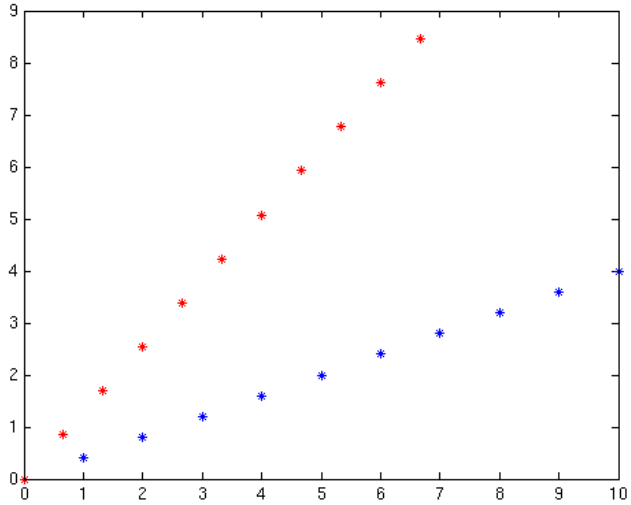


Figure 3.4: The red points, representing the reference surface, are obtained by applying a rototranslation ($30°deg$) centred in the origin of coordinate axes to the blue ones. The Euclidean distance for a pair of corresponding points varies between 0 and 5.5751 making difficult the process of association. In the distance metric proposed by Minguez, all the points shares the same distances to the corresponding ones given by the formula $dist(p_i, p_j) = \sqrt{L^2 * \theta^2}$, with $\theta = \frac{30\pi}{180}$

3.2.2 Metric Based ICP

The second algorithm implemented is the one described by Minguez et al. [52, 53]. The scan matcher proposed in the paper employs a peculiar distance function that explicitly considers the rotation components (see Figure 3.4).

Given a 2D rototranslation RT expressed in vector form (see Appendix A), the authors defined:

$$\begin{aligned} &||RT|| = \sqrt{x^2 + y^2 + L^2 * \theta^2} \\ &\text{where L is a constant, typically } L = 3 \\ &distance(p_1, p_2) = min\{||RT|| s.t. RT \oplus p_1 = p_2\} \end{aligned} \tag{3.4}$$

The expression of the distance has been linearised and the error metric can be defined as:

$$\begin{aligned} &E_{dist}(RT) = argmin \sum_{i=1} (\delta_{ix}^2 + \delta_{iy}^2 - \frac{(\delta_{ix} * p_{iy} - \delta_{iy} * p_{ix})^2}{p_{iy}^2 + p_{ix}^2 + L^2}) \\ &\delta_{ix} = c_{ix} - c_{iy} * \theta + x - p_i x \\ &\delta_{iy} = c_{ix} * \theta - c_{iy} + y - p_i y \end{aligned} \tag{3.5}$$

The expression of $E_{dist}(RT)$ is quadratic with respect to RT and the minimization could be done in closed-form $RT_{min} = -A_{3x3}^{-1} * b_{3x1}$ as explained in

[52, 53]. The adoption of this metric should improve the convergence when the rotation error component is preponderant.

3.2.3 Point-to-Line ICP

The Point-to-Line ICP is an ICP model proposed by Censi in [19]. The distinctive feature of PLICP is the improvement in speed given by the close form minimization of the point to line metric in 2D. This distance metric permits a quadratic convergency rate whereas the classic ICP has a linear one. The explicit formula is given by:

$$\begin{aligned} &E_{dist}(RT) = argmin \sum_{i=1}(n_i^T * (RT * q_{c_i} - p_{i1}))^2 \\ &(p_{i1}, p_{i2}) = \text{segment in reference scan that corresponds to } q_{c_i} \\ &n_i = \text{normal of the segment } (p_{i1}, p_{i2}) \end{aligned} \tag{3.6}$$

The closed-form solution of the error metric is explained by Censi in the appendix of [19] and is based on the solution of the following generalized metric through the use of Lagrange's multipliers.

$$\begin{aligned} &min_{RT} \sum_i ||RT * pi - q_{corresp_i}||^2_{C_i} \\ &\text{where } ||a||^2_C = a^T * C * a \\ &\text{and } C_i = w_i * I_{2x2} \text{ or } C_i = w_i * n_i * n_i^T \\ &\text{with } w_i \text{ weight for point-to-point and point-to-line respectively} \end{aligned} \tag{3.7}$$

In order to get a rototraslation estimation as unbiased as possible, careful attention must be paid to the computation of the normals to the reference surface. In particular, two simple methods could be employed:

- n_i is the normal to the segment formed by (p_{i1}, p_{i2}), the two closest points to q_{c_i}: $n_i = (p_{yi1} - p_{yi2}, p_{xi2} - p_{xi1},)^T$ [19];
- n_i is found averaging the two normals to three sequential points, one of which is the nearest point to q_{c_i} [14]. In this case, the scans should be given with points indexed in the order of polar coordinates.

3.3 Iterative Closest Point Covariance

The problem of correct estimation of the ICP covariance is critical in order to successfully use the scan matchers into a probabilistic filtering framework such as the Kalman Filter. The methods presented in the literature often work well only with a particular scan matching error metric or under particular circumstances. In this section, the most common procedures are reviewed. As already mentioned, the reader should keep in mind that every

Algorithm 3.3.1 Hessian Method

Assuming $Y = MX + w$ with $X = RT$ in vector form
w white noise
$E(\hat{X}) = (Y - M\hat{X})^T(Y - M\hat{X})$
then $Cov(\hat{X}) = (M^T M)^{-1}\sigma^2$
$s^2 = \frac{E_{min}(\hat{X})}{n-k}$
$n =$ number of correspondences
$k = 3$ number of paramters to be estimate
with s^2 estimator of σ^2
$H = \frac{dE^2((\hat{X})}{d\hat{X}^2} = 2M^T M$
then $Cov(\hat{X}) = (\frac{1}{2}H)^{-1}\sigma^2$

algorithm was created with the assumption that the scan matching algorithm selects the global minimum, finds and retrieves the correct transformation within an unknown error. This implies that if the scan matcher function returns the incorrect transformation the relative covariance estimation could be optimistic.

3.3.1 Bengtsson el al.'s Covariance: Hessian Method

Bengtsson el al. in [6] proposed the Hessian method hypothesizing that, if the error function could be linearised, the linear regression theory could be employed to solve this problem. The results reached by the authors are showed in Algorithm 3.3.1. In their paper the Hessian matrix, H, is numerically computed using a sampling technique: after the scan matcher finds the minimum, the scan matching algorithm is repeated giving as initial guess $RT_{result} \pm \delta$ and storing the resulting error. The computation of the Hessian matrix given the error is shown in Appendix B. Lu and Milios in [45] used a similar method estimating directly the matrix M by linearising the $E(\hat{X})$ for small angles. This method appears to give optimistic results and, for this reason, should not be employed in probabilistic filtering frameworks, as Bengtsson explained in [6].

3.3.2 Bosse et al.'s Covariance

Bosse et al. in [14] used an approach similar to Bengtsson's one. In the paper, they hypothesized to have a quadratic error minimization metric. In this case, it is possible to computed the Jacobian of its square root and used this result as the M matrix in Algorithm 3.3.1. The method differs from

Algorithm 3.3.2 Bosse Method

Given $E(\hat{X}) = h(\hat{X})^T h(\hat{X})$
$H = \frac{dh(\hat{X})}{d\hat{X}}$
where $h_i(\hat{X})$ is referred to a single pairs of points
and $H_i = \frac{dh_i(\hat{X})}{d\hat{X}}$
Assuming $E[hh^T] = \sigma^2 I$
then $Cov(\hat{X}) = (H^T H)^{-1}\sigma^2$

Bengtsson since, in general, the Hessian matrix is different from the dot product of Jacobian of the function square root. In particular, they showed the results exposed in Algorithm 3.3.2.

Moreover, in their work, it is possible to read the outcome of the proposed calculus for the point-to-line metric.

3.3.3 Censi's Covariance

Censi in [17] proposed a different method that inherited some features of the Hessian Method and improved the estimate of the covariance taking explicitly into account the uncertainty associated to the measurements $cov(z)$.

$$Cov(\hat{X}) = (\frac{\delta E^2((\hat{X})}{\delta \hat{X}^2})^{-1}(\frac{\delta E((\hat{X})}{\delta \hat{X} \delta z})cov(z)(\frac{\delta E((\hat{X})}{\delta \hat{X} \delta z})^T(\frac{\delta E^2((\hat{X})}{\delta \hat{X}^2})^{-1} \quad (3.8)$$

Some hints for the correct implementation can be found in Censi's `The C(anonical) Scan Matcher`[1] and presented in Algorithm 3.3.3.

3.3.4 ICP Covariance by Sampling

The ICP covariance can be also estimated using sampling techniques. Olson [61] explained how to exploit the probabilistic motion model and the correlative scan matching to robustly compute the covariance matrix. A more naive approach could be applied also; after having estimated the rototranslation, other scan matching instances could be run by sampling the initial guess in the solution neighbourhood. Standard statistical techniques can be then used to retrieve the average and compute the covariance given the set of rototranslation vectors.

[1] http://www.cds.caltech.edu/ andrea/research/sw/csm.html

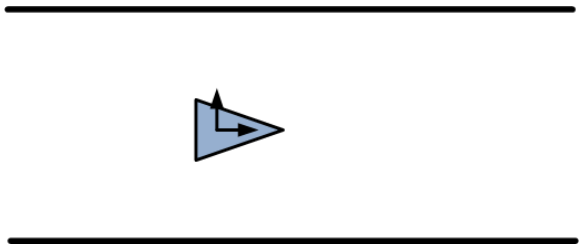


Figure 3.5: The robot in blue is moving in a corridor

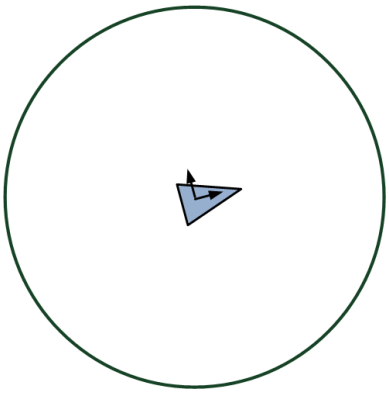


Figure 3.6: The robot in blue is moving in a circular environment

3.3.5 ICP Covariance Critical Situations

Environments, in particular artificial ones, can present characteristics that challenge the best scan matching algorithms. While performing indoor acquisitions, many scans can be affected by ambiguities arising from the simmetry of the rooms. Two critical situations can be studied: the corridor and the circular environment. This two under-constrained situation are in some way dual: in the corridors, Figure 3.5, there is an unobservable direction related to a translational movement, while in the circular environment, Figure 3.3, if the laser is in the middle, all the angles are equiprobable. In both the cases only the projections of the errors on the observable manifold are significant.

Algorithm 3.3.3 Censi Method

for all correspondences
indexes: $i, j1, j2$
where scans are expressed in polar coordinates
and the errors affects only the reading
component ρ, not the angle
$(E_k)_{1x1}$ is the error metric for the
considered corresponding pair
$(\nabla E_k)_{3x1} = \frac{\delta E_k((\hat{X})}{\delta \hat{X}}$
$(\frac{\delta E^2((\hat{X})}{\delta \hat{X}^2})_{3x3}$
$di = (\frac{\delta E^2((\hat{X})}{\delta \hat{X} \delta i})_{3x1}$ wrt ρ_i
$dj1 = (\frac{\delta E^2((\hat{X})}{\delta \hat{X} \delta j1})_{3x1}$ wrt ρ_{j1}
$dj2 = (\frac{\delta E^2((\hat{X})}{\delta \hat{X} \delta j2})_{3x1}$ wrt ρ_{j2}
$E_{tot} = E_{tot} + E_k$
$G_{tot} = G_{tot} + \nabla E_k$
$G2_{tot} = G2_{tot} + \frac{\delta E^2((\hat{X})}{\delta \hat{X}^2}$
$dgdEdi(:, i) = dgdEdi(:, i) + di$
$dgdEdj(:, j1) = dgdEdj(:, j1) + dj1$
$dgdEdj(:, j2) = dgdEdj(:, j2) + dj2$
where $dgdEdi_{3xcount(currentScan)}$, $dgdEdj1_{3xcount(referenceScan)}$
endfor
if $Cov_{scanmatching}$
$(\frac{\delta A}{\delta z})_{3xcount(referenceScan)+count(currentScan)} = G2^{-1}[di, dj1, dj2]$
$R = \sigma^2 I_{count(referenceScan)+count(currentScan)}$
$Cov = (\frac{\delta A}{\delta z}) R (\frac{\delta A}{\delta z})^T$
if $Cov_{localization}$
$(\frac{\delta A}{\delta z})_{3xcount(currentScan)} = G2^{-1} di$
$R = \sigma^2 I_{count(currentScan)}$
where σ^2 is chosen
$Cov = (\frac{\delta A}{\delta z}) R (\frac{\delta A}{\delta z})^T$

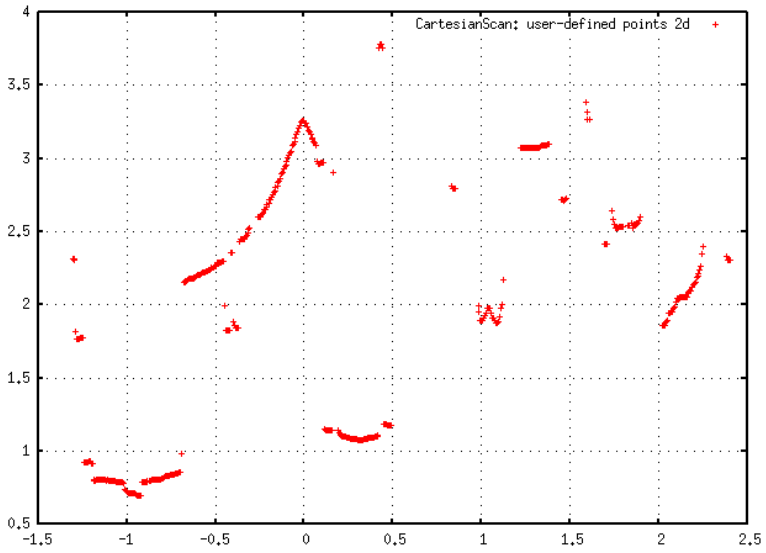


Figure 3.7: Median Filtering in polar coordinates for Hokuyo laser range finder. On x axis is displayed the angle while y axis shows the sensor readings. This figure shows the unprocessed scan

3.4 Scan Preprocessing

Many procedures are described in the literature with the aim of improving the robustness of the scan matcher or as a preliminary phase for scan segmentation. In the present work, the two simplest techniques are employed to develop a first stage outlier rejection and, in this way, deal with possible misalignment due to the wrong estimation of the normal vectors, n_i.

In the implementation, a 1D Median Filter is employed to filter the range measures in polar coordinates. This intuitive algorithm in pseudo code is described in Algorithm 3.4.1 and the default value of the `MedianWindow` is equal to 5 as suggested by many authors (e.g., [25]). In this way, in the case of a circular environment, the readings/measurements in polar scan are linear and the filter can replace at most 2 neighbours outliers out of 5. Figures 3.7, 3.8, 3.9 show the results of the median filtering for subsequent scans in polar and in Cartesian coordinates.

Whereas, a 1D (Weighted) Mobile Average filter is implemented in the case of scans handled in cartesian coordinates. The basic algorithm is the same as Algorithm 3.4.1, where the `doMedian` method is substituted by Algorithm 3.4.2.

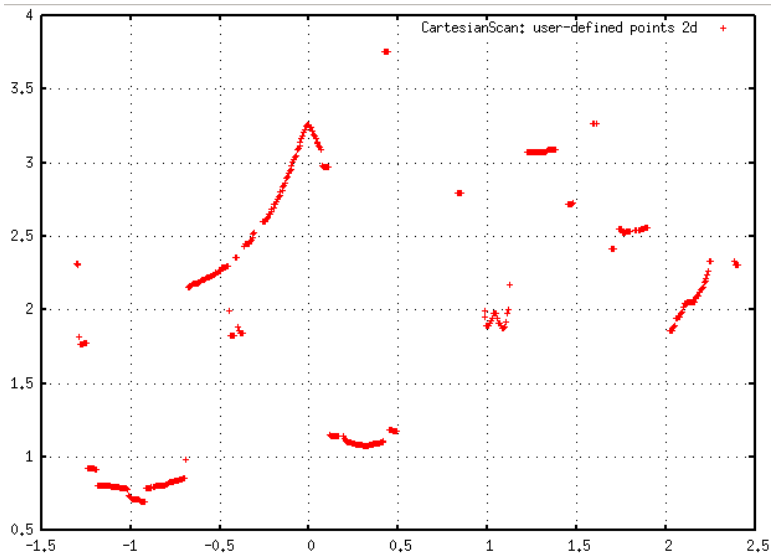


Figure 3.8: Median Filtering in polar coordinates for Hokuyo laser range finder. On x axis is displayed the angle while y axis shows the sensor readings. This figure shows the result of the median filters. As, it can be noted, most of the points remain unchanged while the most isolated one are filtered out

3.5 Outlier Rejection

Outlier rejection techniques aim to minimize the bias in the final transformation, RT, which is caused by the inclusion of outliers in the choice of corresponding pairs. Even if some outliers are directly signalled and marked by the sensors thanks to the use of out of range values or code errors, most of them are only detectable using appropriate procedures. Moreover, ad hoc methods are suggested in the literature to cope with typical source of errors. In the software library implementation, the simplest yet effective techniques are used:

Distance Threshold a maximum distance could be used as a threshold. The corresponding pairs whose distance is above that threshold are automatically excluded in the computation of the figure of merit. Particular attention must be paid to the problem of the correct estimation of the threshold parameter; very low values could decrease the speed of convergence and eventually cause an empty set of correspondences. Conversely, a loose constraint could reduce the efficacy of outlier rejection. Error Metrics which are less affected by this problem have been proposed such as [66].

Percentile Threshold as explained in Trimmed Iterative Closest Point

Algorithm 3.4.1 1D Median Filter

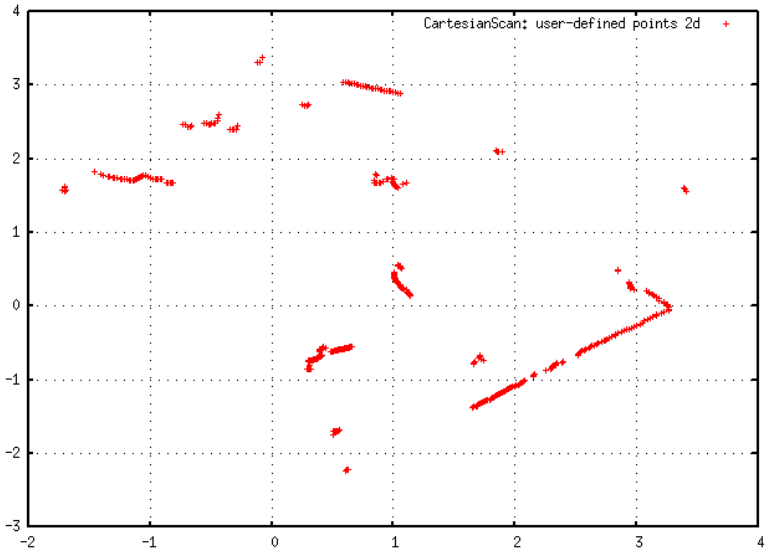
```
[measuresRet] = MedianFilter(measures, MedianWindow)
half = ⌊MedianWindow/2⌋
for i=1:size
 if i < half ∨ i > (size − half)
   measuresRet(i) = measures(i)
 else
   measuresRet(i) =
    doMedian(measures,i,i−half, i+MedianWindow−half)
 end
end
[median] = doMedian(measures,i,p,k)
 littleArray = sort(measures,p,k)
 median = littleArray(⌊(k − p)/2⌋)
```

Algorithm 3.4.2 1D Weighted Mobile Average Filter

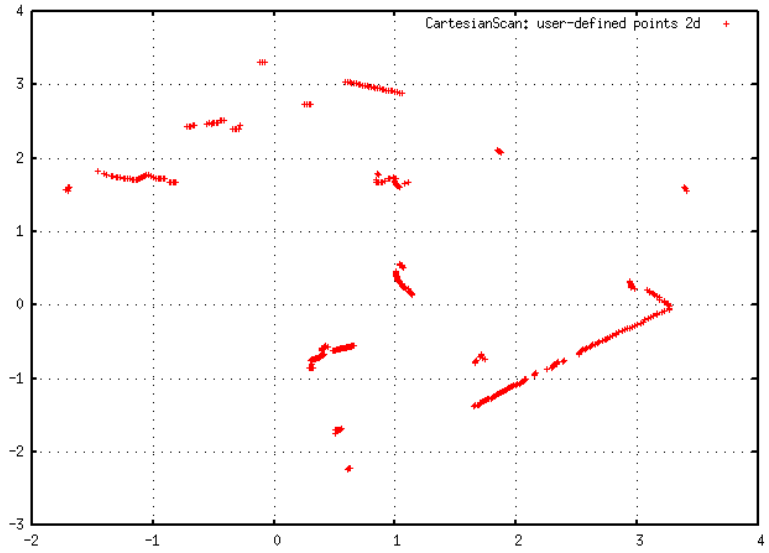
```
[dataRet] = MobileAverageFilter(data,MAWindow, dataWeights)
...
[ma] = doMobileAverage(data,i,p,k,dataWeights)
 ma = (Σ_{j=p}^{k} data_j * dataWeights_j) / (k−p)
```

[24], a fixed or adaptive percentile threshold can be used to obtain an effective outlier rejection. Having selected a number N (overlap parameter), all the distances between corresponding pairs are sorted in ascending order. Only those that have $dist < N$ are selected and used in the computation of the error metric. This method is not subject to the problem of a possible empty set. The value of the overlap parameter could be bounded between 0.4 and 1.0, as suggested in [24].

Rejection of Boundaries is an additional strategy that could be used successfully to eliminate a particular kind of error which is explained in [63]. The technique consists in excluding pairs of corresponding points that include points on mesh/scan boundaries (i.e., if the points are acquired sequentially *exclude* $\forall_{i,j}(p_i, q_j)$ *s.t.* $i \in \{0, |i|\} \vee j \in \{0, |j|\}$. This solves a common problem when the two scans are not complete, that is when the scans share the same shape only partially (See Figure 3.10).

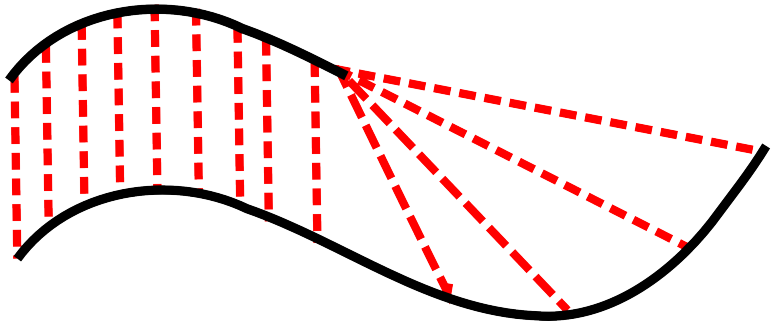


(a) Without Median Filter

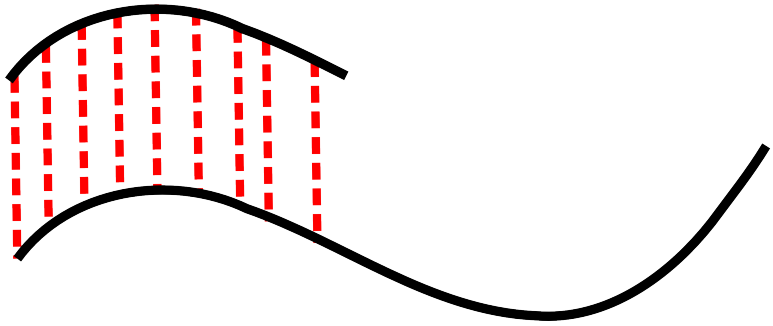


(b) With Median Filter

Figure 3.9: Median Filtering in Cartesian coordinates for Hokuyo laser range finder.



(a) Without the rejection of boundaries



(b) With the rejection of boundaries

Figure 3.10: Rejection of Boundaries: an example of outlier rejection. The first figure shows that many points corresponds to the same one in the reference scan increasing the probability of wrong alignment

Chapter 4

Kalman Filter

"One of the most insidious and nefarious properties of scientific models is their tendency to take over, and sometimes supplant, reality."

Erwin Chargaff

The aim of this chapter is to explain how the Kalman Filter can be exploited to successfully solve the problem of Simultaneous Localization and Mapping. Moreover, the particular probabilistic motion model used in the implementation is shown in order to gently introduce the scanSLAM algorithm.

4.1 The Algorithm of Kalman Filter

The Kalman Filter [70] is a recursive state estimator belonging to the family of Gaussian Filters. This filter was invented by Swerling and Kalman [41] in 1959 for filtering and predicting the behaviour of linear systems. The variables' momentary estimate, which is also named belief or state, is fully represented in the Kalman Filter, by the first and the second moments of a Gaussian distribution in continuous space (μ_t, Σ_t). Using the Kalman Filter it is possible to estimate the Gaussian posterior, which is computed with the following assumptions:

- $p(x_t|u_{t-1}, \Sigma_{t-1})$ is linear:

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t, \tag{4.1}$$

  where u_t is control vector, $\varepsilon_t =$ is a Gaussian white noise WN$\sim (0, R_t)$.

- $p(z_t|x_t)$ is linear:

$$z_t = C_t x_{t-1} + \delta_t, \tag{4.2}$$

Algorithm 4.1.1 EKF Filter

$[\mu_t, \Sigma_t] = \text{KalmanFilter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$
Prediction:
$\bar{\mu} = A_t\mu_{t-1} + B_t u_t$
$\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t$
Update:
$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$
$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$

where z_t is the measurement vector, δ_t is a Gaussian white noise WN$\sim (0, Q_t)$ and the belief of x_0 is distributed as a Gaussian N(μ_0, Σ_0).

If the three assumptions hold the posterior belief is also Gaussian and it can be computed using the Algorithm 4.1.1.

The Algorithm 4.1.1 is split into two phases: prediction and update. In the first step the probability $p(x_t|u_{t-1}, \Sigma_{t-1})$ is computed using the control input u_t and the predicted $\bar{\mu}, \bar{\Sigma}_t$ are returned. Typically, $\bar{\Sigma}_t > \Sigma_{t-1}$ because the prediction incorporates new noise in the state estimation process. The second step computes $p(z_t|x_t)$ using the information acquired using the new measurements. The Kalman update reduces the covariance by integrating new knowledge obtained through the sensors.

The Kalman Filter is usually employed as a non linear state estimator by linearising the state and measurement equation using the Taylor Series theory. The resulting algorithm is named Extended Kalman Filter (EKF). In particular, as it is explained in the following sections concerning EKF SLAM, the covariances matrices A, C are substituted by the Jacobians:

$$A \rightarrow \frac{\partial stateEquation}{\partial x_{t-1}}, \tag{4.3}$$

$$C \rightarrow \frac{\partial measurementEquation}{\partial x_t}. \tag{4.4}$$

4.2 Extended Kalman Filter for SLAM

The Simultaneous Localization and Mapping problem is a fundamental problem that needs to be solved in order to create a truly autonomous robot.

As for any autonomous agent, the robot has sensors from which it can estimate its trajectory (using odometry) and perceive the environment. In

SLAM, the robot, using the control input history, $u_{1:t}$, and the information acquired through the sensors, $z_{1:t}$, should be able to build a consistent map of the environment and simultaneously locate itself on the map. Since the real sensors are noisy and affected by errors, a filtering technique is needed to robustly estimate map and robot pose.

One of the best known techniques for SLAM is the Extended Kalman Filter. The adoption of such a filter is peculiar to feature-based maps, where salient points, named features, can be reliably detected using robot sensors.

Compared to the general Kalman Filter implementation (see Algorithm 4.1.1), the EKF SLAM specializes in the meaning of state variables and their maintenance. The state is represented by x_t:

$$x_t = \begin{pmatrix} x_v \\ y_1 \\ \dots \\ y_i \\ \dots \\ y_L \end{pmatrix} \tag{4.5}$$

where $x_v = x_{vehicle}$ defines the vehicle state variables, often represented by the robot pose (x, y, θ), while y_i stands for a feature (e.g., the pose of the feature in the space). During the iterations, the state variables can be expanded or shrunk by the addition or the elimination of features.

In the same way, the covariance matrix, $\Sigma = P_{t|t}$, shares the dynamic size and it is formed by:

$$P_{t|t(v+lL \times v+lL)} = \begin{pmatrix} P_{xx(v\times v)} & P_{xy_1(v\times l)} & \dots & P_{xy_L(v\times l)} \\ P_{y_1x(l\times v)} & P_{y_1y_1(l\times l)} & \dots & P_{y_1y_L(l\times l)} \\ \dots & \dots & \dots & \dots \\ P_{y_Lx(l\times v)} & P_{y_Ly_1(l\times l)} & \dots & P_{y_Ly_L(l\times l)} \end{pmatrix} \tag{4.6}$$

where v is the size of the vehicle state[1] while l is the size of one feature. In the EKF SLAM algorithm the equation $\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}, u_t)$, which represents $p(x_t|u_{t-1}, \Sigma_{t-1})$, is named motion model/odometry equation, and the generative model $z_i = h_i(x_v, y_i)$, which represents $p(z_t|x_t)$, is referred to as observation or measurement model.

As already mentioned, the EKF algorithm for SLAM has a dynamic size state vector and covariance matrix. The introduction of new landmarks/features into the filter, which increases the size of the first, $x_{t|t}$, and second moment, $P_{t|t}$, of the Gaussian, is commonly performed as follows [10].

[1] e.g., 3 if a 2D robot motion is represented only by its pose (x, y, θ)

Algorithm 4.2.1 EKF Filter for SLAM

$[\hat{x}_{k|k}, P_{k|k}] = \text{EKF–SLAM}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$
Prediction:
$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k)$
$P_{k|k-1} = \frac{\partial f}{\partial x} P_{k-1|k-1} \frac{\partial f^T}{\partial x} + Q_k$
Update:
$\hat{y}_k = z_k - h(\hat{x}_{k|k-1}, P_{k|k-1})$
$S_k = \frac{\partial h}{\partial x} P_{k-1|k-1} \frac{\partial h^T}{\partial x} + R_k$
$K_k = P_{k-1|k-1} \frac{\partial h^T}{\partial x} S_k^{-1}$
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k$
$P_{k|k} = (1 - K_k \frac{\partial h}{\partial x}) P_{k|k-1}$

In order to add a new landmark, the state is augmented as:

$$\begin{aligned} x_t &= \left(x_t, y_L\right)^T \\ y_L &= inverseSensorModel(x_v, z_L) \end{aligned} \tag{4.7}$$

The covariance matrix is expanded in the following way:

$$P_{t|t} = \begin{pmatrix} P_{xx} & P_{xy_1} & \dots & P_{xy_{L-1}} & P_{xx}\frac{\partial y_L}{\partial x_v}^T \\ P_{y_1x} & P_{y_1y_1} & \dots & P_{y_1y_{L-1}} & P_{y_1x}\frac{\partial y_L}{\partial x_v}^T \\ \dots & \dots & \dots & \dots & \\ P_{y_{L-1}x} & P_{y_{L-1}y_1} & \dots & P_{y_{L-1}y_{L-1}} & P_{y_{L-1}x}\frac{\partial y_L}{\partial x_v}^T \\ \frac{\partial y_L}{\partial x_v}P_{xx} & \frac{\partial y_L}{\partial x_v}P_{xy_1} & \dots & \frac{\partial y_L}{\partial x_v}P_{xy_{L-1}} & A \end{pmatrix} \tag{4.8}$$

where

$$A = \frac{\partial y_L}{\partial x_v} P_{xx} \frac{\partial y_L}{\partial x_v}^T + \frac{\partial y_L}{\partial \hat{z}_L} R \frac{\partial y_L}{\partial \hat{z}_L}^T \tag{4.9}$$

4.2.1 Motion Model

The motion model refers to the probabilistic modellization of the estimate and to the prediction of the trajectory of the robot using the control vector u_t as an input. In general, input to this model can be the robot wheel encoders, the estimated robot pose that results from a GPS localizer or integrated from other sensors, the robot speeds or others.

The model used in this thesis is taken from [70] and it is implemented in the software library due to the possibility to use directly the robot pose estimates without any particular time constraints. This aspect is particularly convenient in the scanSLAM algorithm when more than one laser is

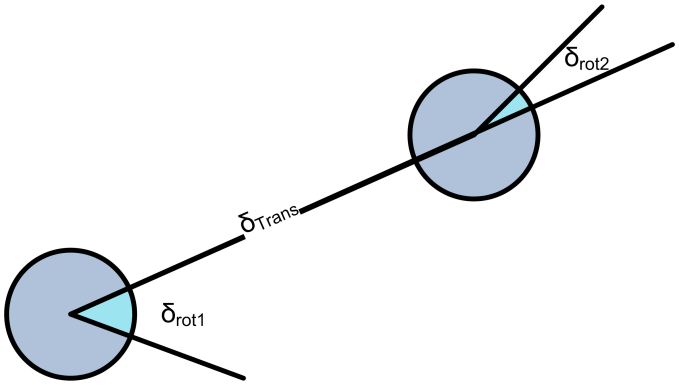


Figure 4.1: The probabilistic motion model is defined by three consecutive movements:δ_{rot1}, δ_{trans}, δ_{rot2}

employed and in which they are desynchronized one from the other. The Odometry Motion Model can be defined as follows:

$$u_t = \begin{pmatrix} x_{v,t-1} \\ x_{v,t} \end{pmatrix} \tag{4.10}$$

The motion of the robot is modelled as three consecutive movements: δ_{rot1}, δ_{trans}, δ_{rot2}. As it can be seen from Figure 4.1, the robot first performs a rotation, then a translation and again a final rotation on its axis. The three variables can be obtained directly from u_t using the Algorithm 4.2.2.

$$\begin{pmatrix} \delta_{trans} \\ \delta_{rot1} \\ \delta_{rot2} \end{pmatrix} = \texttt{FromUtoDelta}(u_t) \tag{4.11}$$

And the motion model prediction $\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}, u_t)$ is:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} cos(\theta + \delta_{rot1}) & 0 & 0 \\ sin(\theta + \delta_{rot1}) & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \delta_{trans} \\ \delta_{rot1} \\ \delta_{rot2} \end{pmatrix} \tag{4.12}$$

Algorithm 4.2.2 From u_t to ($\delta_{rot1}, \delta_{trans}, \delta_{rot2}$)

$[\delta_{rot1}, \delta_{trans}, \delta_{rot2}] = \text{FromUtoDelta}(u_t)$
$x_{v,t-1} = (\bar{x}, \bar{y}, \bar{\theta})^T$
$x_{v,t} = (\bar{x}', \bar{y}', \bar{\theta}')^T$
$\delta_{rot1} = atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$

Moreover, the EKF algorithm requires some Jacobian matrices in order to compute correctly the value of the state and the covariance. The Jacobian with respect to the state is:

$$\frac{\partial f(\hat{x}_{k-1|k-1}, u_k)}{\partial x_{v,t-1}} = \begin{pmatrix} 1 & 0 & -\delta_{trans} * sin(\theta + \delta_{rot1}) \\ 0 & 1 & +\delta_{trans} * cos(\theta + \delta_{rot1}) \\ 0 & 0 & 1 \end{pmatrix} \tag{4.13}$$

The matrix $Q_k = V_k M_k V_k^T$ is the covariance of the additional motion noise in the state space, which is derived from two matrices: M_k, which represents the noise in the control space and V_k, which allows the mapping from the control to the state space.

$$\begin{aligned} \sigma_{trans} &= \alpha_3 \delta_{trans} + \alpha_4(|\delta_{rot1}| + |\delta_{rot2}|) \\ \sigma_{rot1} &= \alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans} \\ \sigma_{rot2} &= \alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans} \end{aligned} \tag{4.14}$$

$$M_k = \begin{pmatrix} \sigma_{trans}^2 & 0 & 0 \\ 0 & \sigma_{rot1}^2 & 0 \\ 0 & 0 & \sigma_{rot2}^2 \end{pmatrix} \tag{4.15}$$

and

$$V_k = \frac{\partial f}{\partial u} = \begin{pmatrix} cos(\theta_{t-1} + \delta_{rot1}) & -\delta_{trans} * sin(\theta_{t-1} + \delta_{rot1}) & 0 \\ sin(\theta_{t-1} + \delta_{rot1}) & \delta_{trans} * cos(\theta_{t-1} + \delta_{rot1}) & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{4.16}$$

Chapter 5

ScanSLAM

"The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work"

John von Neumann

ScanSLAM is a SLAM algorithm based on the Extended Kalman Filter (EKF) which is capable to deal directly with raw laser scans. As already mentioned in Chapter 2, the method could be also seen as an example of Pose Snapshot Kalman Filter (PSKF) because the representation of the scans as features or landmarks in the state of the filter is the robot pose itself.The scientific paper about scanSLAM written by Nieto et al. [59] explains the generic observation model and the filter update mechanism. The same model is explained at the beginning of this chapter. Diosi et al. [25] extended this model without achieving the full generalization, which is showed in the rest of the chapter. Variations of the general model are proposed to handle multiple lasers and to calibrate the exstrinsic parameters. Novel strategies to tune and improve the convergence of the filter are also explained.

5.1 ScanSLAM - Sensor Centred with One Laser

In the sensor centred approach the laser sensor is located in the odometric centre of the robot, i.e., the rototranslation between the laser reference frame and the robot reference frame is an identity matrix. Under this assumption, the scanSLAM model can be reduced to Nieto et al. implementation [59].

5.1.1 State and Landmarks

The state of the Extended Kalman Filter is given by:

$$x_t = \begin{pmatrix} x_v \\ y_1 \\ \cdots \\ y_i \\ \cdots \\ y_L \end{pmatrix}, \tag{5.1}$$

where the features, from now on referred to as landmarks, consist of two parts: the state variable y_i and the associated scan $scan_i = \{p_j\}$. The state variable defines the pose of the robot in the world reference frame where that particular scan has been acquired.

The collection of scans is saved in a different data structure and does not interfer with the EKF update mechanism. The variables that refer to the vehicle x_v are defined using the motion model presented in Chapter 4.

Moreover, one landmark is added in the state x if the robot is distant more than a threshold from all the other landmarks.

5.1.2 Observation Model

Since the landmarks and the robot pose are expressed homogeneously, it is straightforward to define the observation model as the rigid transformation (rototranslation) between the robot and the landmark.

Intuitively, the measurement can be seen as the pose of the landmark frame y_i as it is observed by the robot x_v (see Figure 5.1). This can be reduced to:

$$\hat{z} = h(y_i, x_v) = (\ominus x_v) \oplus y_i, \tag{5.2}$$

which is equal to:

$$\begin{pmatrix} \hat{z}_x \\ \hat{z}_y \\ \hat{z}_\theta \end{pmatrix} = \begin{pmatrix} (x_{yi} - x_t)cos(\theta_t) + (y_{yi} - y_t)sin(\theta_t) \\ -(x_{yi} - x_t)sin(\theta_t) + (y_{yi} - y_t)cos(\theta_t) \\ \theta_{yi} - \theta_t \end{pmatrix}. \tag{5.3}$$

5.1.3 Update

Every time the robot pose is near to one of the landmarks, it is probable that the scan associated to the landmark and the current one are representing a common portion of the same environment. Recalling the signature of the scan matcher function which was shown in Chapter 3:

$$[RT, Cov] = scanMatch(referenceScan, currentScan, RT_{guess}) \tag{5.4}$$

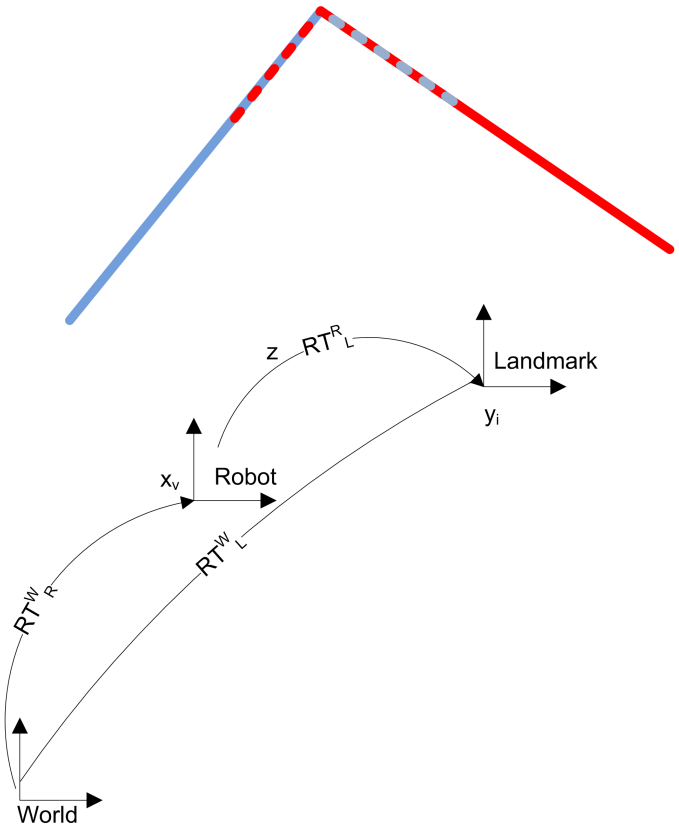


Figure 5.1: The reference frames are displayed together with a couple of scans

it could be noted that the scan matcher applied to the couple of scans $(currentScan, landmarkScan_i)$ returns exactly $RT = z$, where z is the measurement of the Extended Kalman Filter. In fact, the scan matching algorithm retrieves the rototranslation superimposing the $landmarkScan_i$ to $currentScan$. Moreover Cov can be directly used in the filter as the covariance matrix of the additional measurement noise R_k (see the Algorithm 4.2.1).

5.1.4 EKF Filter and Jacobians

In order to implement the Extended Kalman Filter it is useful to refer to the Algorithm 5.1.1 which shows the signatures of the most fundamental functions which compose the EKF. The methods `covariancePrediction`, `doUpdate` and `addLandmarks` are implemented using the formulae of Algorithm 4.2.1. The Jacobians needed for the implementation of the sensor centred scanSLAM are exposed in the following.

Algorithm 5.1.1 EKF Filter for SLAM

$[\hat{x}_{k|k}, P_{k|k}] = \text{EKF–SLAM}(\hat{x}_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$
Prediction:
$[\hat{x}_{k|k-1}, Q_k] = \text{predict}(u_k, (\hat{x}_{k-1|k-1}))$
$[P_{k|k-1}] = \text{covariancePrediction}(P_{k-1|k-1}, Q_k)$
Update:
$[z_k, \hat{z_k}, R_k] = \text{observe}(\hat{x}_{k|k-1}, P_{k|k-1})$
$[\hat{x}_{k|k}, P_{k|k}] = \text{doUpdate}(\hat{x}_{k|k-1}, P_{k|k-1}, z_k, \hat{z_k}, R_k)$
New Landmarks:
$[\hat{x}_{k|k}, P_{k|k}] = \text{addLandmarks}(\hat{x}_{k|k-1}, P_{k|k-1}, z_k, \hat{z_k}, R_k)$

The Jacobian of the observation with respect to the state x is:

$$\frac{\partial h}{\partial x} = \begin{pmatrix} \frac{\partial h}{\partial x_{vehicle}} & \dots & 0 & \dots & \frac{\partial h}{\partial y_i} & \dots & 0 & \dots \end{pmatrix}, \tag{5.5}$$

$$\frac{\partial h}{\partial x_v} = \begin{pmatrix} -cos(\theta_t) & -sin(\theta_t) & -(x_{yi} - x_t)sin(\theta_t) + (y_{yi} - y_t)cos(\theta_t) \\ sin(\theta_t) & -cos(\theta_t) & -(x_{yi} - x_t)cos(\theta_t) - (y_{yi} - y_t)sin(\theta_t) \\ 0 & 0 & -1 \end{pmatrix}, \tag{5.6}$$

$$\frac{\partial h}{\partial y_i} = \begin{pmatrix} cos(\theta_t) & sin(\theta_t) & 0 \\ -sin(\theta_t) & cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.7}$$

The Jacobians needed for adding new landmarks according to the Equations 4.7, 4.8, 4.9:

$$\frac{\partial y_L}{\partial x_v} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5.8}$$

$$\frac{\partial y_L}{\partial \hat{z}_L} = \begin{pmatrix} cos(\theta_t) & -sin(\theta_t) & 0 \\ sin(\theta_t) & cos(\theta_t) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5.9}$$

In this last step, since the landmark has the same value of the current pose of the robot, the matrix R_k is zero and the initial covariance submatrix $A = P_{y_L y_L} = P_{xx}$.

5.1.5 Loop Closure

The most important characteristic of the use of an Extended Kalman Filter for SLAM is related to the phenomenon of loop closure. In Figure 5.2, it is

shown a sequence of iterations taken from a simulation of the sensor centred scanSLAM made in Matlab. Here, the motion of the robot is supposed to describe a circle. During the movement, at every metre, a new landmark is added to the filter.

It could be noted that the robot pose uncertainty increases in time, as indicated by the uncertainty ellipses of growing diametres. This is due to the intrinsic uncertainty associated to the motion model. As direct consequence, the uncertainty associated to landmarks increases too, since it is given by P_{xx} at the time that new landmark is created. The amazing fact happens when the robot senses, after long time, an old landmark. In this case, since the covariance matrix becomes fully correlated after few iterations, the EKF update step affects not only the robot pose uncertainty, which drastically shrinks, but also, indirectly, the uncertainty of all the previous landmarks which are correlated to the sensed one. The result is that the location of all previous landmarks becomes more certain thanks to the information brought by that single observation.

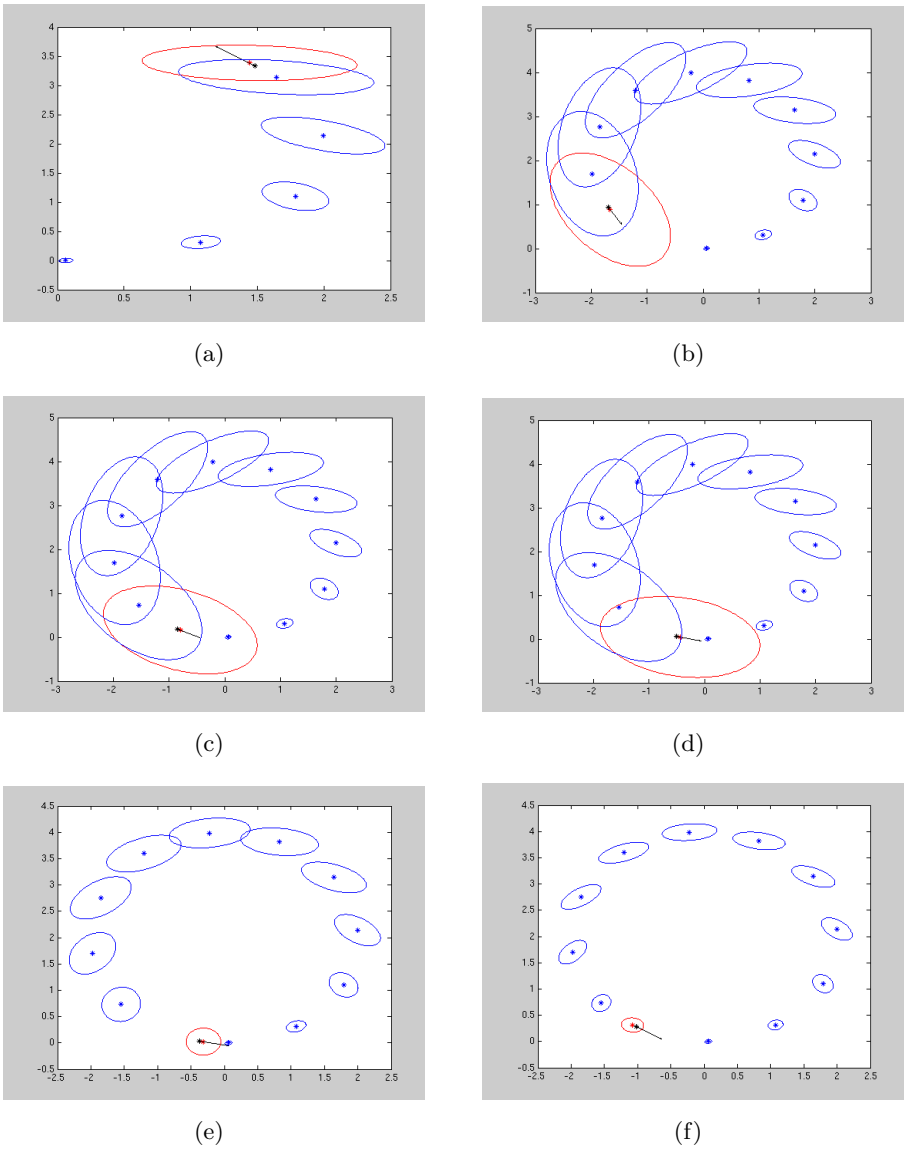


Figure 5.2: This is a sequence generated in simulation to test scanSLAM and its loop closure capability. The uncertainty covariance associated to the robot pose is shown in red while the landmark one is drawn in blue. The arrows show the direction of the vector pose (x, y, θ). When the robot senses an old landmark all the covariances are updated and reduced. The last image is taken after a second counterclockwise round.

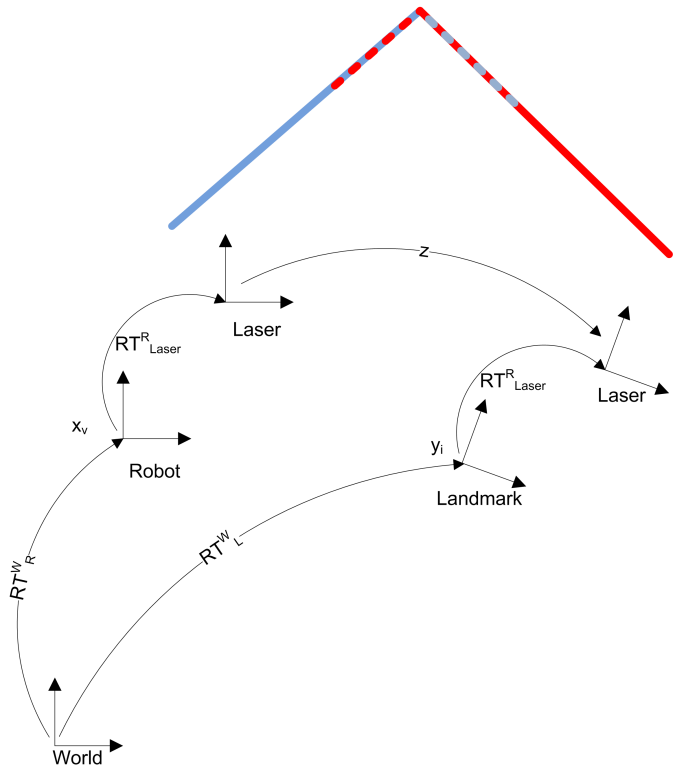


Figure 5.3: The main reference frames for scanSLAM - Robot Centred with One Laser

5.2 ScanSLAM - Robot Centred with One Laser and Exstrinsic Calibration

The scanSLAM algorithm which has just been described can be seen as a specialization of a more general framework where the transformation between the robot reference frame and the laser reference frame is not negligible.

In this case, it is possible to add to the state x the parameters related to that rototranslation $RT^{robot}_{laser} = (tx, ty, t\alpha)$. This operation allows to calibrate the extrinsic parameters, RT^{robot}_{laser}, if the covariance submatrix $P_{x_{param}|x_{param}} \neq 0_{3x3}$ at the initialization step of the Extended Kalman Filter. The graphical representation of this model is shown in Figure 5.3.

5.2.1 State and Observation Model

The state of the Extended Kalman Filter is now defined as:

$$x_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t tx \\ ty \\ t\alpha \\ y_1 \\ \dots \\ y_i \\ \dots \\ y_L \end{pmatrix} \tag{5.10}$$

In this case, $x_{vehicle} = \begin{pmatrix} x_t, & y_t, & \theta_t, & tx, & ty, & t\alpha \end{pmatrix}$ comprises also the rototranslation parameters which remain unchanged in the prediction step. The landmarks y_i are defined as in the previous model: the pose of the robot in the world reference frame registered when the $scan_i$ is acquired as new landmark.

The observation model $h(\hat{x}_{k|k-1}, P_{k|k-1})$ is modified in order to be compatible with the measurement z_k returned by the scan matcher algorithm. Following the representation in Figure 5.3, the observation equation becomes:

$$\begin{pmatrix} \hat{z}_x \\ \hat{z}_y \\ \hat{z}_\theta \end{pmatrix} = (\ominus x_{param}) \oplus (\ominus x_v) \oplus y_i \oplus x_{param} = \begin{pmatrix} c_1 + c_2 \\ c_3 + c_4 \\ \theta_{yi} - \theta_t \end{pmatrix},$$

where:

$$\begin{aligned} c_1 &= (x_{yi} - x_t)cos(\theta_t + t\alpha) + (y_{yi} - y_t)sin(\theta_t + t\alpha) \\ c_2 &= -txcos(t\alpha) + txcos(\theta_{yi} - t\alpha - \theta_t) - tysin(t\alpha) - tysin(\theta_{yi} - t\alpha - \theta_t) \\ c_3 &= -(x_{yi} - x_t)sin(\theta_t + t\alpha) + (y_{yi} - y_t)cos(\theta_t + t\alpha) \\ c_4 &= -tycos(t\alpha) + tycos(\theta_{yi} - t\alpha - \theta_t) + txsin(t\alpha) + txsin(\theta_{yi} - t\alpha - \theta_t) \end{aligned} \tag{5.11}$$

In this way $\tilde{y}$ can be still defined as $z_k - \hat{z}_k$ because both the scan matcher and the observation model equation return the rototranslation $R^{laserRobot}_{laserLandmark}$, which is the rototranslation between the two laser scans.

The Jacobians can be computed in Matlab or GNU/Octave using the Algorithm 5.2.1. The matrix $\frac{\partial h}{\partial x}$ changes in accordance to:

$$\frac{\partial h}{\partial x} = [\frac{\partial h}{\partial x_{vehicle}} \frac{\partial h}{\partial x_{param}} \dots 0 \dots \frac{\partial h}{\partial y_i} \dots 0 \dots] \tag{5.12}$$

Algorithm 5.2.1 Jacobians Computation for Robot Centred scanSLAM

```
% Scan Matcher: z_k
syms Xsm Ysm Tsm real
SM = [Xsm Ysm Tsm]'
% Robot: x_{t-1}, x_t
syms xtm1 ytm1 ttm1 xt yt tt real
XT = [xt,yt,tt]';
% Landmark Pose
syms xL yL tL real
L = [xL,yL,tL]';
% Laser Pose
syms Xh Yh Th real
HOK = [Xh Yh Th]'
deltaX = compound(inv_compound(XT),L)
deltaSM = compound(inv_compound(HOK),deltaX);
deltaSM = compound(deltaSM,HOK);
%  h(x̂_{k|k-1}, P_{k|k-1})
deltaSM1 = simplify(deltaSM)
% ∂h/∂x
ALL = [XT;HOK;L];
JACH = jacobian(deltaSM1, ALL)
% Vehicle
jacHdxv = jacobian(deltaSM1, XT)
% Laser sensor (e.g., Hokuyo)
jacHdhok = jacobian(deltaSM1, HOK)
% Landmark
jacHdL = jacobian(deltaSM1, L)
disp('Find the inverse transformation h^{-1} to add landmark')
deltaXRis = compound(HOK,SM);
deltaXRis = compound(deltaXRis,inv_compound(HOK));
deltaXRis1 = simplify(deltaXRis);
LwrtWorld = simplify(compound(XT,deltaXRis))
subs(LwrtWorld,SM,[0,0,0])
% Vehicle
jacLwrtWorlddxv = jacobian(LwrtWorld,XT)
subs(jacLwrtWorlddxv,SM,[0,0,0])
% Laser
jacLwrtWorlddHOK = jacobian(LwrtWorld,HOK)
subs(jacLwrtWorlddHOK,SM,[0,0,0])
%wrt z_k which is not used R=zeros
jacLwrtWorlddSM = jacobian(LwrtWorld,SM)
subs(jacLwrtWorlddSM,SM,[0,0,0])
```

5.3 ScanSLAM - Multiple Laser with Exstrinsic Calibration

The use of cheap laser range finders, such as Hokuyo URG-04LX, makes attractive the equipment of more than one laser on a single robot.

Few approaches can be pursuit in order to incorporate information gathered by multiple sensors into the filter: sensor fusion using Covariance Intersection [72, 73, 38, 57, 58, 40], decentralized sensor fusion algorithms [62, 46, 74], or a generalization of the scanSLAM algorithm.

The scanSLAM framework offers the opportunity to deal with the problem of multiple laser using an elegant solution. In fact, it is possible to add another laser in a seamless way and simultaneously refine its calibration parameters, by adding the extrinsic calibration information, the rototranslation $RT^{robot}_{laser_j} = (tx_j, ty_j, t\alpha_j)$, in the EKF state x as it was just done for the previous case. In the data structure that collects all the scans only an additional variable must be set in parallel to each new landmark describing the laser sensor in use. Modifications to deal with the problem of data association and multiple desynchronized laser acquisitions are discussed in the following sections. Figure 5.4 shows that the observation made by one laser with respect to the other and its related Jacobian can be computed as follows. In addition to the Jacobians presented for the case of a single laser range finder, it is also possible to obtain as observation the rototranslation between the two lasers. If the currentScan is acquired by the laser k and an associated landmark is registered by the laser h then:

$$\begin{pmatrix} \hat{z}_x \\ \hat{z}_y \\ \hat{z}_\theta \end{pmatrix} = (\ominus x_{param_k}) \oplus (\ominus x_v) \oplus y_h \oplus x_{param_h} = \begin{pmatrix} c_1 + c_2 \\ c_3 + c_4 \\ \theta_{t\alpha_h} - \theta_{t\alpha_k}\theta_{yh} - \theta_t \end{pmatrix},$$

where:

$$\begin{aligned} c_1 &= (x_{yh} - x_t)cos(\theta_t + t\alpha_k) + (y_{yh} - y_t)sin(\theta_t + t\alpha_k) \\ c_2 &= -t\alpha_k - \theta_t) - ty_k sin(t\alpha_k) - ty_k sin(\theta_{yh} - t\alpha_k - \theta_t) \\ c_3 &= -(x_{yh} - x_t)sin(\theta_t + t\alpha_k) + (y_{yh} - y_t)cos(\theta_t + t\alpha_k) - ty_k cos(t\alpha_k) \\ c_4 &= +ty_h cos(\theta_{yh} - t\alpha_k - \theta_t) + tx_k sin(t\alpha_k) + tx_k sin(\theta_{yh} - t\alpha_k - \theta_t) \end{aligned} \tag{5.13}$$

5.4 Data Association Problem

The term data association refers to the problem of finding the correct associations between the current measurement and the corrisponding landmarks.

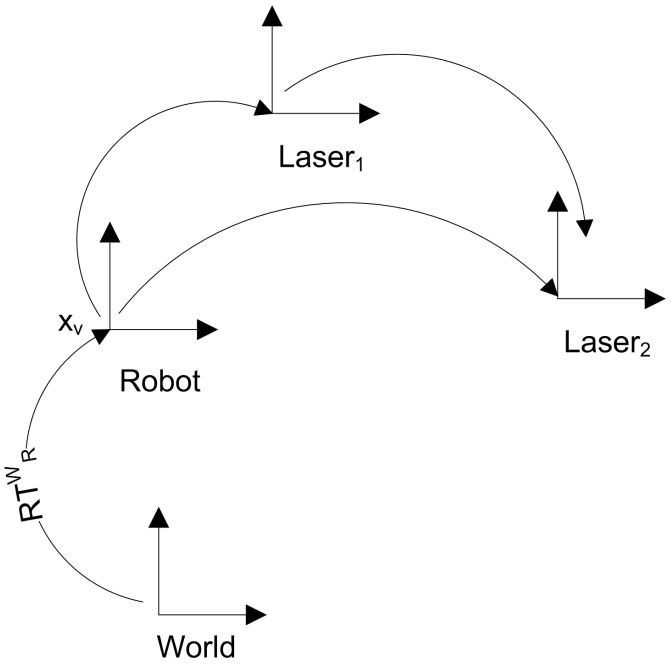


Figure 5.4: Reference frames involved in Multiple Laser scanSLAM model

An incorrect procedure can easily result in unconsistency of the Extended Kalman Filter with the consequences of a wrong estimation in the pose of the robot and in the creation of a global map of the environment.

On the one hand, the nature of the observation in scanSLAM, which are constituted by raw laser scans, allows some improving in the robustness of this crucial step with respect to the bearing-range SLAM [10, 70]. On the other hand, the assumptions made in the computation of the covariance matrix R_k related to the additional noise in the measurement arises new problems which must be carefully handled. In the next subsections, all the techniques used in the implementation are described. In the final implementation the techinques are typically employed together to solve the data association problem.

5.4.1 Selection of Landmarks Based on Distances

The selection of the landmarks which are candidates to be matchable to the current scan is made using the information derived from the distance between the robot pose, x_v at time t, and each landmark. Two distance metrics are used in this stage: the Euclidean and the Mahalanobis distance, see Appendix D. The details for the correct implementation are described

Algorithm 5.4.1 Selection of Landmarks

$[Set] = \text{SelectionOfLandmarks}(\hat{x}_{k|k}, P_{k|k})$
$P_v = P(1:3, 1:3)$
for $i = 1 : numLandmarks$
$dist_{euclidean} = (x_t - y_{x_i})^2 + (y_t - y_{y_i})^2$
$dist_{orientation} = |\theta_t - y_{\theta_i}|$
$dist_{mahalanobis} = (x_v - y_i)^T P_v^{-1} (x_v - y_i)$
if $(dist_{euclidean} < d_{eu} \wedge dist_{orientation} < d_{or})$
$\vee$ $(dist_{mahalanobis} < d_{ma})$
$Set = Set \bigcup \{i\}$
endif
endfor

in the Algorithm 5.4.1.

The two distance metrics are employed as follows. A landmark l is chosen only if its euclidean distance along (x, y) and its orientation is sufficiently near the robot or if it is compatible to the Gaussian representation of the pose of the robot.

It should be noted that if the uncertainty related to the robot is relatively low, the active constraint is the one given by the Euclidean distance. Viceversa, if the robot pose is very uncertain then the constraint on the Mahalanobis distance permits to choose candidates in accordance to the motion model in use.

It could happen that no candidates can be considered given the distances thresholds. In that case, the update of the EKF will be skip for the current iteration. This is reasonable since a very distant landmark could probably not be matchable with the current scan and the scan matcher would returns the wrong transformation z_k.

5.4.2 ML Estimator

Once, the landmarks are selected on the basis of the distances, a stricter method can be employed to improve the robustness of the scanSLAM algorithm with the purpose of rejecting a measurement if it is not compatible with its prediction.

The technique described in the Algorithm 5.4.2 is taken from [70] and implements a Maximum Likelihood estimator. If there are N landmarks which are candidate to be matchable with the current scan, the scan matching procedure is repeated for all the couples producing $(z_k, \hat{z}_k, R_k)_{i=1..N}$.

Algorithm 5.4.2 ML Data Association

$[j]$ = MLDataAssociation$(\{z_k\}, \{\hat{z}_k\}, \{R_k\}, P_{k|k})$
for $i = 1 : numLandmarks$
$S = \frac{\partial h_i}{\partial x} P_{k|k} \frac{\partial h_1}{\partial x}^T + R_{k,i}$
$d_i = (z_{k,i} - \hat{z}_{k,i})^T S^{-1} (z_{k,i} - \hat{z}_{k,i})$
endfor
$j = \arg\min_i \ d_i$

The measurement which it will be used in the update step of the EKF it is choosen employing for the second time the Mahalanobis distance, now computed on the mearurements z and not on the robot pose and features, to the corresponding prediction. The measurements with the lowest distance are selected.

Intuitively, the maximum likelihood estimator chooses the measurement that will change the state the least, in accordance to the pose estimation performed so far. In addition to this procedure, it is possible to consider the value of quality indexes that can be computed by the scan matcher such as the ratio given by the number of corresponding pairs within a distance threshold over all the corresponding pairs.

Indexes like the one just described can partially solve the problem given by the wrong estimation of R_k, the covariance of the scan matcher. As mentioned before, the covariance matrix estimated by the ICP covariance algorithm can be optimistic if the scan matcher converges to the wrong minimum.

5.4.3 Multiple Data Association

If more than one landmarks candidates are selected through the previous processes, it is possible to perform a EKF update taking into account all the measurements and, therefore, obtaining a faster convergence rate.

The update is done by stacking the matrix $\frac{\partial h}{\partial x}$,$R_k$ and the vector z_k, $\hat{z}_k$ in the following way:

$$H = \begin{pmatrix} \frac{\partial h_1}{\partial x} \\ \dots \\ \frac{\partial h_{numObservation}}{\partial x} \end{pmatrix}, \tag{5.14}$$

$$z = \begin{pmatrix} z_{k,1} \\ \dots \\ z_{k,numObservation} \end{pmatrix}, \tag{5.15}$$

$$\hat{z} = \begin{pmatrix} \hat{z}_{k,1} \\ \dots \\ \hat{z}_{k,numObservation} \end{pmatrix}, \tag{5.16}$$

$$R = \begin{pmatrix} R_{k,1} & 0 & \dots & 0 \\ 0 & R_{k,2} & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & R_{k,numObservation} \end{pmatrix}, \tag{5.17}$$

and then following the Algorithm 4.2.1.

5.4.4 Dealing with Desynchronized Measures

The phenomenon of desynchronized measures commonly affects the implementation of multiple laser models. For this reason, some laser manufacters release laser sensor with a dedicated physical input to synchronize data acquired from an array of sensors. However, this solution is applicable only in those particular cases.
The solution proposed in this work is based on the following three rules:

- If a laser scan i is acquired by a laser sensor k and no other landmarks made by the same sensor are present in the neighbourhood, add the currentscan as new landmark. Otherwise, perform the update as it is done in the case of the single sensor;
- Only if the robot is not moving, add the current scans acquired from all the lasers in the collection of scans related to the landmark described by $y_i = x_v$. In this step, it is possible to perform the measurement update of one laser with respect to the other one (see Section 5.3) if the two scans retrieved by the lasers are theoretically matchable (e.g., the relative location of the lasers permits the scan to share at least a constant portion of the environment);
- In the other cases, perform only the EKF prediction step

This approach permits to reuse most of the code of the single sensor scanSLAM and leave unchaged the structure of the state vector x and its covariance matrix P. Even if most of the landmarks are associated to a single laser sensor, the presence of few landmarks which are updated by multiple sensors pemits to improve the estimate of all the landmarks since they remain correlated.

Another feasible approach is fusing all the laser scans acquired by different sensors into a single scan. The EKF update is delayed until all the

scans are retrieved. Once the information is gathered, all the scans should be rototranslated by the $\delta motion$ of the robot during the acquisition phase. Only after that, the scan can be matched against another landmark or store as a new one.

It should be noted that in the second case, the fusion between scans is performed without taking into consideration the uncertainty of the transformations between each laser reference frame; this can result in a wrong data fusion process. Conversely, the first method is much more affected by the modelization of the errors related to the extrinsic parameters of each laser sensor.

5.4.5 High Speed Rotations

An often understated problem can arise when a laser range finder with low scanning speed is mounted on a robotic platform.

This issue manifests when the scanning speed is too low in comparison to the rotational speed of the robot ω, which is considered negligible in most of the robotic application. The correct mathematical equation for the computation of the angle of each readings for a sensor subjected to a rotational speed ω_t, expressed in the world reference frame, becomes:

$$\forall i : realAngle_i = angleSensor_i + \theta_t + \omega_t \Delta t \; i \tag{5.18}$$

where $angleSensor_i$ is the i^{th} value of the angle of a laser beam expressed in the laser reference frame, θ_t is the robot orientation at time t, ω_t is the rotational speed of the robot at time t, Δt is the interval of time necessary to obtain a single laser measurement.

This problem is particularly relevant when the dataset from L.U.R.C.H (see Chapter 6). is used. In fact, considering the worst case, on the hypothesis that the Hokuyo laser is located at the centre of the robot, and defining:

- V_{max}: wheel maximum speed equals to $2\frac{m}{s}$
- b: the wheelchair baseline equals to $0.5m$
- Hokuyo scanning speed equals to 10Hz, over 1024 motor steps

The maximum rotational speed becomes:

$$\omega_{max} = \frac{2V_{max}}{b} = 8\frac{rad}{s} \tag{5.19}$$

and results, simulated using Matlab, are graphically shown in Figure 5.5, 5.6. This can be considered an open problem and in the software implementation

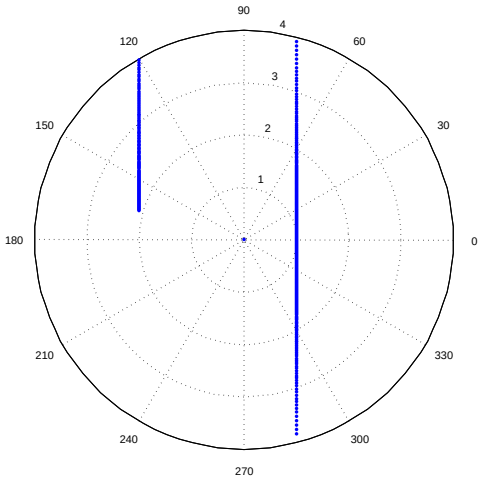
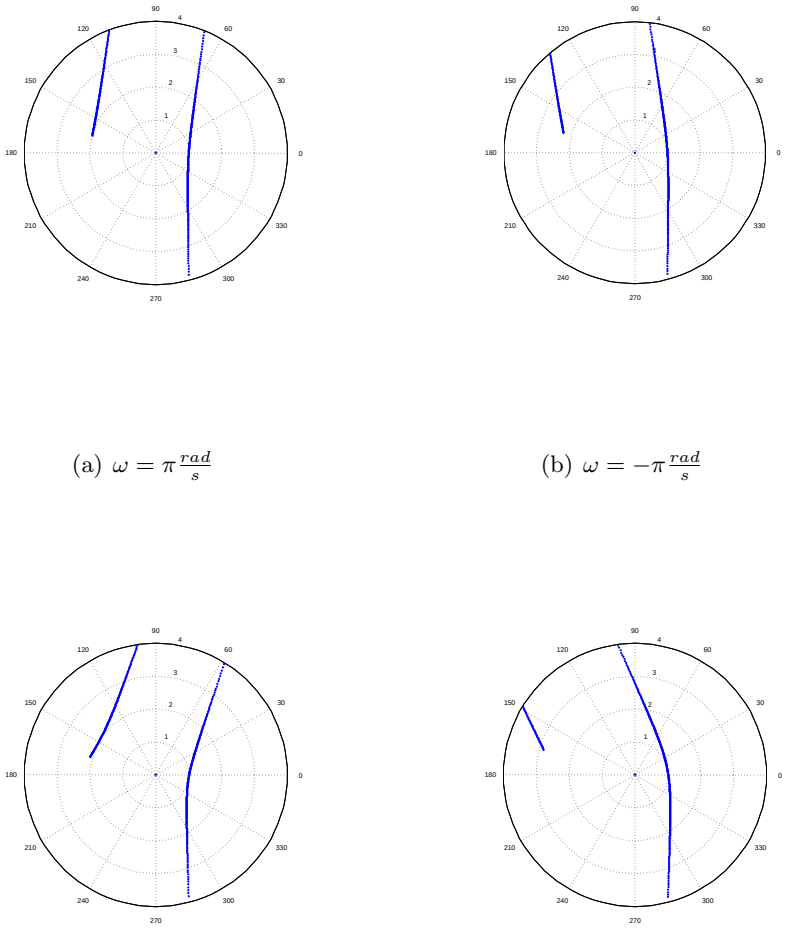


Figure 5.5: This is a polar plot representing a typical Hokuyo laser range finder acquisition of a corridor. The laser is located in the origin of the axis with the orientation of LURCH right laser, whose angles are set in the range $[-1.307, 2.872], [rad]$. *The rotational speed* ω *is set to zero. The laser range is 4m only, the out-of-range are set to zero to simulate the behaviour of the real laser*

it is simply choosen to discard the laser scans if the robot speed is too high.



(a) $\omega = \pi \frac{rad}{s}$ (b) $\omega = -\pi \frac{rad}{s}$

(c) $\omega = +8 \frac{rad}{s}$ (d) $\omega = -8 \frac{rad}{s}$

Figure 5.6: This is a polar plot representing a Hokuyo laser range finder acquisition of a corridor. The sensor is subjected to a rotational speed ω while during data acquisition. Distorsions make harder the problem of scan matching, in particular when scans taken at different rotational speed are used as input. The value of Δt of Equation 5.18 is set to $\frac{100ms}{1024motorstep}$

Chapter 6

Datasets

'The Scientist must set in order. Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house."

Henri Poincaré, Science and Hypothesis (1901)

This chapter describes which datasets have been selected during the testing phase of the software library and the different kind of laser sensors that have been taken into consideration. Two different robotic platforms have been studied in relation to Rawseeds and LURCH projects. The choice of testing different laser range finders, which have been selected from available datasets of the two projects, made necessary an appropriate tuning of the parameters of the scan matching and Extended Kalman Filter algorithms. Particular emphasis is given to LURCH project to evaluate the possibility of using the software library to enhance the robot with navigation features and extend the current functionalities.

6.1 Rawseeds Project

Rawseeds[1] is a Specific Support Action funded under the European Union's Sixth Framework Programme. The project is dedicated to the generation and publication of a high quality benchmarking toolkit for robotics.

In order to complete that suite, high quality datasets have been published together with the description of the location and the characteristics of the environment: static, if there are no object moving or doors being opened, dynamic, in the opposite case, with natural or artificial lighting.

[1] http://rawseeds.org

Within Rawseeds terminology, the word dataset is used to identify the "set of synchronized data streams obtained by recording the output of the sensors mounted on a robot when it explores an environment". Whereas, in the same context, a single instance is called data-gathering session.

The dataset comprises data acquired by multiple sensors: low cost short-range lasers, medium and long-range ones, sonars, digital cameras and odometric data.

Moreover, the parameters of the calibration of each sensor both the extrinsic, the rototranslation between the sensor and the robot reference frame, and the intrinsic ones, the calibration matrix for digital cameras, are accurately estimated and the ground truth for at least a part of each session is provided.

6.1.1 Dataset: "Bicocca Indoor"

The dataset used to test the scanSLAM algorithm is named "Bicocca (indoor)"[2]. In particular, a fragment of the session "Bicocca 2009-02-25b" is selected[3] for the presence of a loop. The location is set into the building of Università di Milano Bicocca, Italy, and, in particular, the session is recorded in a static environment with artificial lighting. The blueprint of the building with the superimposed approximated robot path is shown in Figure 6.1.

The robotic platform designed for Rawseeds project is named Robocom[4] which is jointly developed by POLIMI and UNIMIB. The robot is equipped with an array of the most common sensors available in the market:

- Inertial Measurement Unit, IMU, which provides 3-axis angular orientation, acceleration, rate-of-turn and Earth magnetic field data (Xsense MTi)
- Short-Range laser range finders provided by two lasers (Hokuyo URG-04LX)
- Medium-Range laser (SICK LMS200)
- Long-Range laser (SICK LMS291)
- Three on-board computers, PCBricks[5]
- Ultrasound transducers arranged to form a belt of 12 sonar (Maxbotix EZ-2) with limited range

[2] http://www.rawseeds.org/rs/datasets/view/6
[3] http://www.rawseeds.org/rs/capture_sessions/view/5
[4] http://www.rawseeds.org/home/2007/12/14/robocom-robot/
[5] http://airwiki.elet.polimi.it/mediawiki/index.php/PCBricks

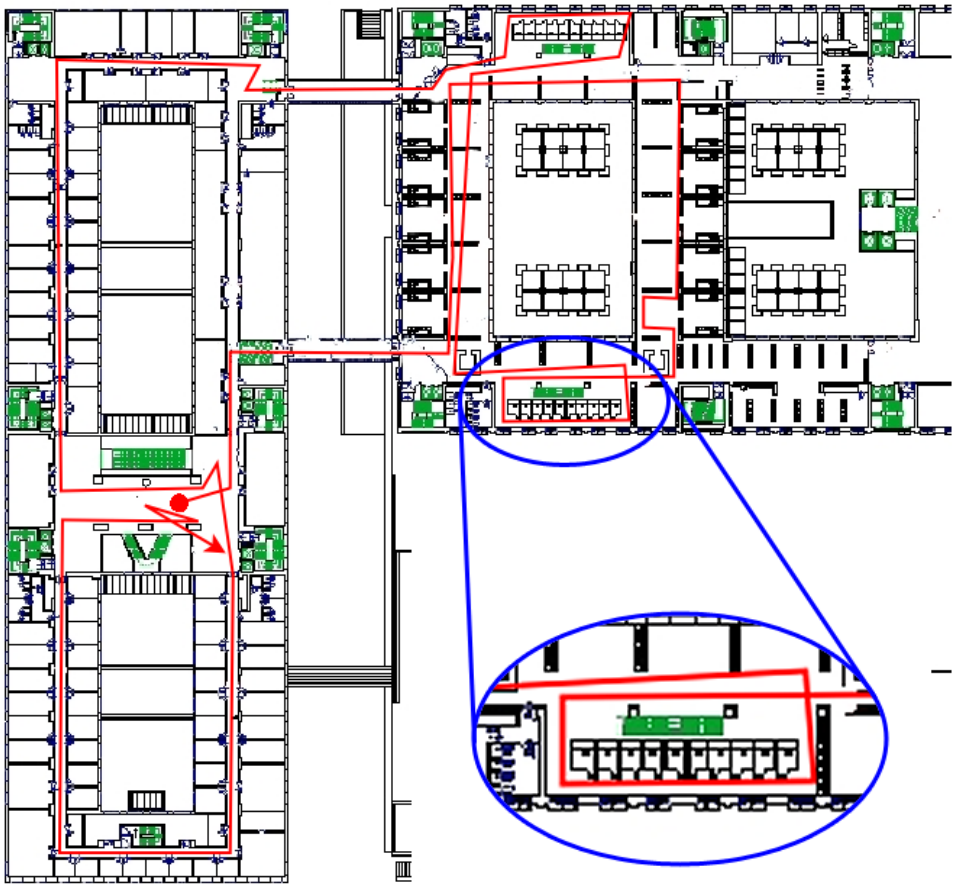


Figure 6.1: The blueprint of the Bicocca building related to the Rawseeds session "Bicocca 2009-02-25b"

- Stereo Vision System (Videre Design STH-DCSG-VAR) made by two B/W, 640x480 pixel digital cameras
- Trinocular Vision System by coupling the binocular one with another (Videre Design DCSG) camera
- Monocular Vision System provided by a colour, 640x480 pixel (Unibrain Fire-i 400) digital camera
- Omnidirectional Vision System obtained with a digital camera (Prosilica GC1020C) and an hyperbolic mirror (Vstone).

The principal reference frames related to the robot are shown in Figure 6.2. For this case of study, it is used the data acquired by the SICK LMS291 frontal laser in order to check the correctness of the robot centred

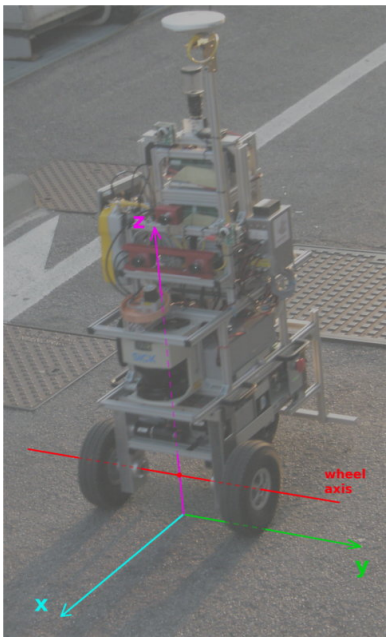


Figure 6.2: The robot reference frames for the Rawseeds project. The SICK laser sensor is on the x axis. The main reference frames involved are the same of the Figure 5.3

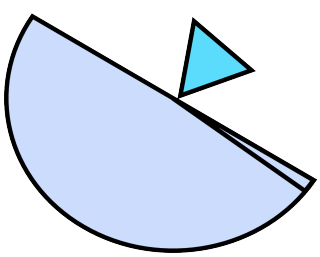
single laser scanSLAM implementation. The results are shown in Section 7.2.2 of the following chapter.

6.1.2 The SICK laser

The laser range finder SICK LMS291 is a very accurate long-range laser sensor designed for indoor applications. The measurement are based on the time-of-flight of an infrared laser beam reflecting back on a rotating mirror. The instrument has a coverage area of 180°with a selectable angular resolution of 0.25°, 0.50°, 1°.

The long maximum range, 80m, together with a measurement resolution of 10mm and a 75Hz scanning speed, allows the detection of a large area of the environment in a single scan improving the scan matching results. Nevertheless, the long range can cause problems in the rototranslation estimation because many corresponding points may have a high distance (e.g., while moving in corridors). The main features, which has just been elicited, are shown in Figure 6.3.

(a)



Coverage Area:	180°
Max Distance:	80m
Angular Resolution:	0.25° 0.50° 1°
Accuracy:	10mm
Scanning Speed:	75Hz

(b)

Figure 6.3: Technical specification for SICK LMS291

6.2 The LURCH Project

LURCH, "Let Unleashed Robots Crawl the House", is a project developed by AIRLab, Artificial Intelligence and Robotics Laboratory, of Politecnico di Milano. The aim of the project is to create an autonomous wheelchair by extending the functionality of a commercial electric wheelchair (Rabbit by Ottoblock) with sensors and robotics software. For this reason, the standard wheelchair is equipped by:

- Two on-board computers, PCBricks[6], powered by wheelchair batteries
- An odometry system based on encoders applied to the rear wheels
- Two laser scanners Hokuyo URG 04LX
- A 7-inch touch screen monitor, Xenarc 700TSV, 800x480 resolution (16:10 AR)

[6]http://airwiki.elet.polimi.it/mediawiki/index.php/PCBricks

Figure 6.4: The LURCH autonomous wheelchair

- A colour camera FireI400 (resolution 640x480)

Both the laser sensors are used to test the scanSLAM algorithm. In the case of the wheelchair configuration and in contrast with Robocom, the two (Hokuyo) lasers are mounted on the left and the right of the seat, instead of being located in front and rear position, as can be seen in Figure 6.4. The two lasers can sense a common part of the environment thanks to their wide detection area. Therefore it is possible to perform the scan matching algorithm between the two lasers to estimate the relative rototranslation. The picture that represents the common area is shown in Figures 6.7. The different reference frames involved are the same of the ScanSLAM model named Multiple Laser with Exstrinsic Calibration, described in Chapter 5.

The calibration between the two laser sensors and the odometric centre of the robot, i.e., the matrices $RT^{robot}_{hokLeft}$ and $RT^{robot}_{hokRight}$, has been computed by hand using an ad hoc method.

6.2.1 The Hokuyo Laser

The laser range finder Hokuyo URG-04LX is a short-range laser scan designed for indoor application which measurement are based on the calculation of the phase difference of an infrared laser.

This instrument, in comparison with SICK, has a wider coverage area, 240°, with an angular resolution of 0.36°(obtaining 683 measurement per scan). The maximum range is 4m which, thanks to the wide detection area, permits to obtain effective results in robotics applications such as collision

Figure 6.5: The Hokuyo URG-04LX

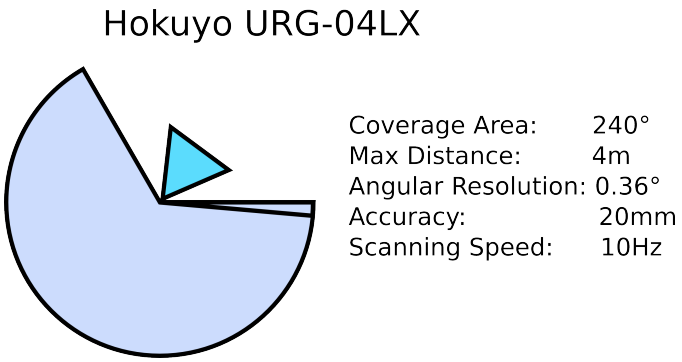
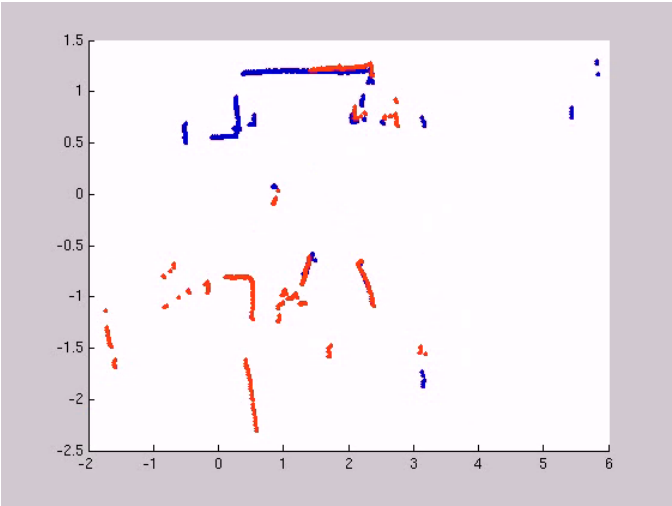


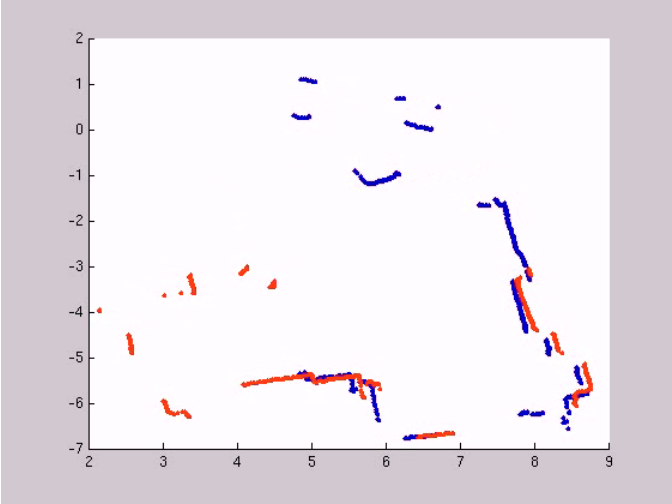
Figure 6.6: Technical specification for Hokuyo URG-04LX

detection and obstacle avoidance. The accuracy of the Hokuyo sensor is lower than the SICK laser one: 10-20mm against 5-10mm.

The relative low scanning speed, 10Hz, makes the scans more prone to distortion if the robot is travelling at high speed (see Section 5.4.5). These features are synthesized and shown in Figure 6.6.



(a)



(b)

Figure 6.7: The figure shows the combined scans made by transforming the scans acquired by the two Hokuyo, using the manually calibrated rototralsations, in robot reference frame. In particular, the blue scans refers to the left sensors, while the red one refers to the right laser. As it can be seen from the Figure b, the calibration parameters are not optimally estimated. Moreover, when the wheelchair performs fast rotation the scans resulted distorted (see Chapter 5)

Chapter 7

Software Implementation and Evaluation

Sijaki banida (Starting is half the task)

A Korean proverb.

This chapter offers in-depth look at the projectual choices on the basis of the software library which has been developed. The functionalities provided by the library and the software architecture are explained together with considerations about optimization and execution time. Finally, the testing activities and the analysis of the results are described in relation to scan matching algorithms, ICP covariance estimators, scanSLAM and self calibration.

7.1 The Software Library

The software library is implemented in C++ using dedicated libraries (Eigen [29], ANN [1] and FLANN [31]) in order to obtain good performances in the iteration procedures and reliable results. Many functions are also implemented in Matlab to test the algorithms, make prototypes and propose and test alternatives.

In spite of the efficient algorithms for matrix manipulation, the long execution time of the simulations which have been implemented in Matlab and the opportunity to use the software for LURCH project made necessary the development of a fast C++ library. This section is dedicated to the description of the software architecture and considerations about the resulting implementation.

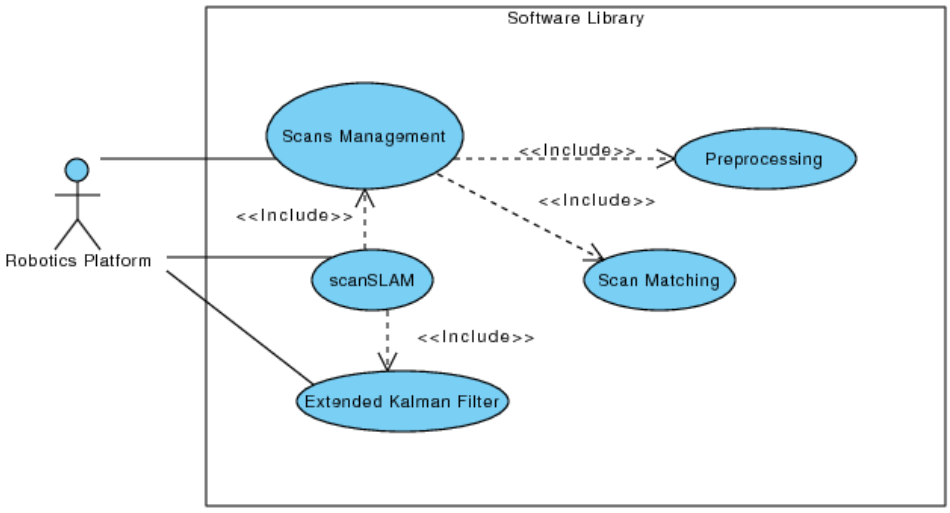


Figure 7.1: The use case diagram for the software library

7.1.1 Functionalities Provided

The aim of the final implementation is to create a software library for the management of the acquisition obtained from laser scans and their successive manipulation. The scan matching algorithms are implemented to use the laser for navigation tasks such as mapping and localization. In order to complete the set of functionalities, algorithms to compute the ICP covariance for every scan matcher have been developed.

A novel, generic and templatized, Extended Kalman Filter framework is added to implement and test scanSLAM algorithm. Finally, multiple laser model and self calibration model are implemented as final applications to evaluate the proposed modification and challenge the software library capabilities. The main functionalities provided to the user are shown in the use case diagram in Figure 7.1.

7.1.2 The Software Architecture

The software architecture is graphically described in the UML package diagram in Figure 7.2. As it can be seen in the package diagram, the software library is logically divided into six macrostructures:

Scan which implements the scans both in polar and Cartesian representation, the scan matcher and the algorithm for estimating the covariances associated to the error in the minimization procedure

Correspondences which solve the problem of finding k nearest neighbours

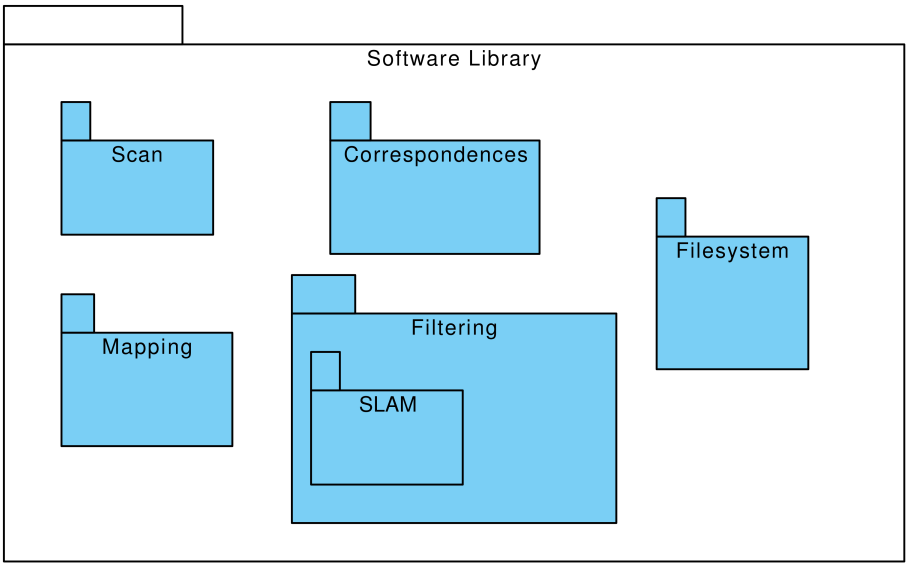


Figure 7.2: The package diagram for the software library

given two array representing 2D data clouds. Internally, it uses a general $O(n^2)$ algorithm, ANN, and FLANN, introduced later in this section

Mapping which includes the test for incremental mapping that uses the scan matchers. It also implements simple maps made of set of scans or 2D points

Filtering comprises general filtering techniques such as templatized median filter, mobile average and Extended Kalman Filter

SLAM which implements the scanSLAM algorithm, including both sensor centred and robot centred approaches

Filesystem contains the simple software classes for the handling of log files

More in details, the classes are made generalizable preferring the C++ template mechanism to virtual functions in order achieve the maximum efficiency. For example, the class that implements the scan matcher is templatized in the type of scan you handle: polar or Cartesian. In this way (see Figure 7.3), the class does not need to internally convert the scans at each iteration since the programmer is committed to provide the right implementation of the scan. The conversion from polar to Cartesian scan, which is needed by all the scan matcher implemented so far, (Classic ICP, Metric ICP,

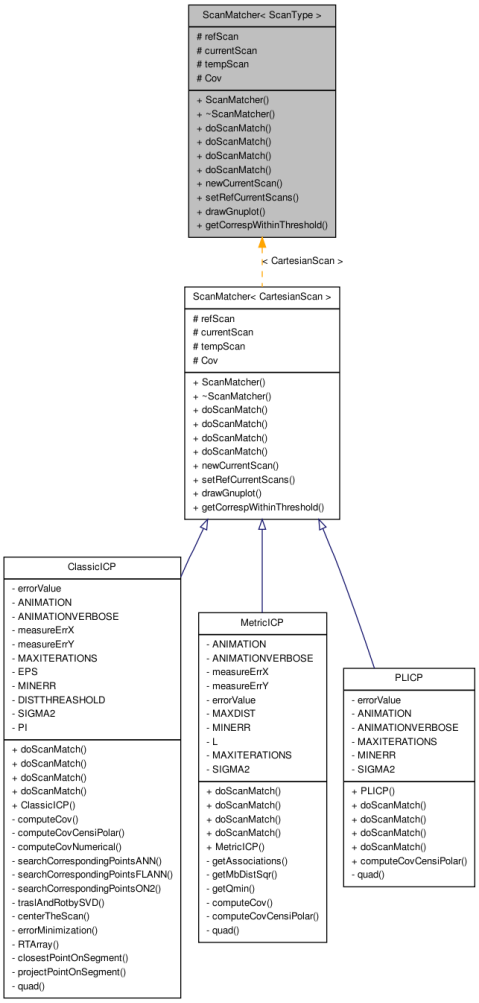


Figure 7.3: The class diagram related to the scan matcher

PLICP), can be simply performed before scan matcher is invoked using dedicated functions for scan manipulation. Nonetheless, the scan matcher generality allows future implementation of different scan matching algorithms such as Polar Scan Matching or probabilistic based scan matchers.

Many other classes such as the one that deals with the search for corresponding pairs are templatized on the datatype to be able to deal with float, double or other built-in datatypes. Moreover, external libraries are used to improve the performance, the reusability and the portability of the software and the single algorithms.

In particular, the following libraries are used:

Eigen is a C++ template library for linear algebra [29] which manipulates

vectors and matrices of static or dynamic size. The API is simple and elegant in comparison to BLAS and LAPACK libraries without the risk of losing performance[1]

ANN is a C++ library for Approximate Nearest Neighbor Searching in d-dimensional space using kd-trees and box-decomposition trees [1]. Given a query point p, the k nearest neighbours are extracted from the dataset M. The generality of the algorithm allows the use of any Minkowsky distance (e.g., Euclidean, Manhattan, max distance)

FLANN Fast Library for Approximate Nearest Neighbors is a C++ library similar to ANN. This library is able to find the best searching algorithm among linear, k-tree, k-means and a combination of them thanks to the computation of a particular figure of merit [56]

GPC Generalized 2D Point Corrispondence solution, [32], is a C software library developed by A. Censi to efficiently find the solution of the point-to-line distance metric

In the testing phase it is found out that the three algorithms for computing the k nearest neighbours, for this specific application, are almost equivalent to the $O(n^2)$ approach. In particular, a value of $k = 2$ is used to retrieve the segment endpoints for the computation of the normals to the segment. This is probably due to the low number of points considered, 680 as maximum. Nevertheless, the optimized algorithms can be reused in future development to deal with scans of variable lenght.

7.1.3 Metaprogramming with Templates and Generalization

The aforementioned C++ template feature is exploited to create reusable components without losing performance thanks to the fact that most of the decisions are made at compile-time. In fact, even if virtual functions introduce less run-time overhead than dynamic_cast or typeid, they can cause a lot of performance penalty when they do not contain much code and they are called frequently (see [76]).

Particular attention is paid to the creation of a generic Extended Kalman Filter class for SLAM which takes inspiration from the EKF implementation made by two distinct project: the Mobile Robot Programming Toolkit (MRPT) [55] and TooN Algorithm Library (TaG) [68].

The Extended Kalman Filter class is defined as:

[1] see benchmarks: `http://eigen.tuxfamily.org/index.php?title=Benchmark`

Algorithm 7.1.1 Example of use of the Extended Kalman Filter class

```
ExtendedKalmanFilter<State1, MotionModel1, Paramters1> ekf;
ekf.state.x = xStart;
ekf.state.P = covStart;
// Prediction
ekf.prediction(u);
// Update
MeasurementModel1 m1;
MeasurementModel1 m2;
ekf.updateAndAddLandmarks(m1);
ekf.updateAndAddLandmarks(m2);
```

```
template <class State, class MotionModel, class Parameters>
            class ExtendedKalmanFilter {...}
```

and the main member functions are: `prediction(...)`, `update(...)` and `addNewLandmarks(...)`. The signature of the class permits to abstract from the particular model in use and test different couples of `State`s and `MotionModel`s, which are designed to be compatible one another. The template class `Parameter` is only used to set constant values and improve Eigen library execution-time thanks to the use of static size matrices which can so be known at compile-time.

In order to achieve the maximum generalization the member function `update(...)` is templatized:

```
template <class MeasurementModel> void update (...)
```

This technique is particularly useful to create EKF SLAM filters where different sensors are employed to get measurements. An example of use of the class is shown in Algorithm 7.1.1. In this algorithm, the EKF object is instantiated and initialized with the starting values of state and covariance (which are implemented using Eigen library). After that, an Eigen vector u is used as input to the prediction step of the EKF. The dimension of u is checked at compile-time by using the information stored in `Parameters` class. Two measurement models are instantiated and the update step is performed. The measurement z is retrieved by a method of `MeasurementModel` class.

Data	*ScanMatcher/Iterations*	*scanSLAM/Iterations*
Hokuyo	$54.583s/2542$	$42.899s/231$
SICK	$32.443s/5020$	$8m51.350s/1355$

Table 7.1: The table shows the average time required to perform incremental mapping based on scan matching and scanSLAM. The time comprises the acquisition of data from log file, the saving of information for plotting in files and the estimation of the ICP covariance. For the scanSLAM algorithm 10 laser acquisition are skipped to improve the speed.

7.1.4 Computational Complexity and Time Performance

The scanSLAM algorithm shares the same characteristic of the EKF model from which it is specialized. The Extended Kalman Filter is computationally bounded in the number of landmarks, N, and its computational complexity is generally classified as $O(N^3)$.

In fact, as it is shown by Blanco in [10], the complexity depends on the inversion of the covariance matrix P which is an N by N matrix. In the same scientific paper is reported that, if the covariance matrix associated to the uncertainty in the measurements R_k is diagonal, and in most scan matching iterations the ICP covariance can be approximated to a diagonal matrix, the computational complexity can be reduced to $O(N^2)$

It should be noted that, at each update step of the EKF, at least one instance of the scan matching algorithm is run. Its computational complexity is difficult to estimate since it depends on the number of iterations, the minimization method used and the computational time due to the estimation of the ICP covariances. Since the ICP algorithm typically converges after few iterations, from 1 to 20, and since the algorithm involves only 3 by 3 matrix inversions, its computational complexity becomes almost negligible when the EKF state covariance P grows.

The Table 7.1 shows the average time requested to perform scan matching and scanSLAM. In comparison to the scanning speed of the two laser sensors, it can be concluded that for the Hokuyo laser it is possible to perform real time incremental mapping because the time to compute a single scan, $0.02s$ is lower than the scanning speed $0.1s$. The same consideration can not be applied to scanSLAM since, even considering 10 scans for each iteration, the time requested for a single iteration is nearly $0.2s$. Also for SICK laser the incremental mapping can be performed in real time because $0.0065s$ is lower than $\frac{1}{75}s$.

7.2 Experimental Results

In this section the main experimental results are shown and discussed. The exposition of the content follows a complexity ordering, from the simplest and self-contained component to the most complex one. Therefore, the section begins with scan matching algorithms and the experimental evaluation of the validity of the algorithms for computing the ICP covariance. These tests are followed by the implementation results obtained using a scanSLAM approach. Also in this case, first the motion model is described and only in the end the calibration extension is critically discussed. In the implementation, in contrast with the majority of the scientific papers, the scan matchers are tested both with high reliable SICK lasers and with low-cost Hokuyo lasers. The scanSLAM algorithm is for the first time in literature tested for Hokuyo laser range finder obtaining good result even with the most general model defined in Chapter 5.

7.2.1 Scan Matching Algorithm

Three scan matchers are implemented in the library: Classic ICP, Metric Based ICP and Point-to-Line ICP. The implementation follows the guidelines proposed in Chapter 3. The algorithms are tested with few modifications with Hokuyo and SICK laser range finders.

In particular, it was necessary to modify the maximum distance related to the **Distance Threshold**, Chapter 3, since the laser range varies from few metres, 4m for Hokuyo lasers, to many, 80m for SICK ones. For this reason, it is omitted the test on the **Distance Threshold** for scans acquired by SICK, while, for Metric Based ICP, the threshold is set to 0.5 [m] with $L = 3$. The **Percentile Threshold** is set to 80% both for Classic and Point-to-Line ICP. While, the **Rejection of Boundaries** is always applied. The scan preprocessing procedure is performed for the scans acquired from Hokuyo lasers using a 1D median filter on range measures with filter window equals to 5.

Six testing case are selected to show the accuracy of the scan matching algorithms and evaluate the estimated covariance. In all the cases, a couple of non consecutive scans is selected. The initial guess obtained by odometry is set to zero in order to prevent the correct alignment given by good estimate and test the scan matchers capabilities in the worst situations.

The cases Rawseeds 1 and Rawseeds 2, see Figure 7.4, are constituted by two scan couples taken from the Rawseeds dataset (see Chapter 6). Rawseeds 1 refers to the couple of scans numbers 8000 and 8125. Therefore,

the second scan is acquired after $\frac{125}{75Hz} = 1.7s$. Rawseeds 2 refers to the couple 5300, 5305.

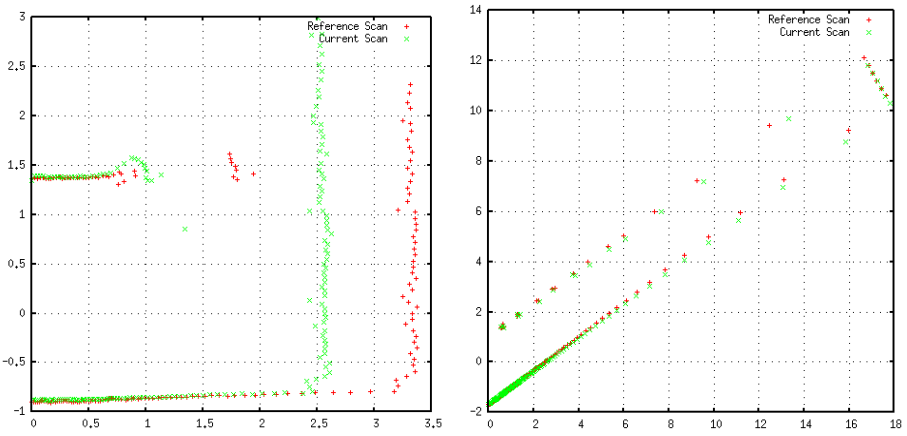
The test case named Hokuyo, see Figure 7.5, is obtained using the autonomous wheelchair (see Chapter 6) in the AIRLab laboratory in Milan. The scans couple is formed by two non consecutive scans 200, 205. Therefore in this case, the time interval is $\frac{5}{10Hz} = 0.5s$.

The test cases named CorridorX, CorridorY, and Circular, see Figures 7.9, 7.10 and 7.11, are obtained from a simulated dataset with the aim of recreating a corridor along the principal axis x,y and a circular environment. The data are noise free but comply the Hokuyo requirements: 4m maximum range and angular resolution of 0.36°.

The results are shown in Figures 7.6, 7.7, 7.8, 7.9, 7.10 and 7.11. From a visual inspection it can be seen that, in general, all scan matchers which are implemented in the software library show good performance. Classic ICP in two cases (Figures 7.6 and 7.8) converges to a wrong minimum. This is probably due to the poor selective outlier rejection criteria employed. Metric ICP performs well in every dataset but the Rawseeds 1, in which Point-to-Line ICP performs visually better. The case of test taken from the simulation environment, the two corridors and the circular environment, show visually the same results for each scan matcher and the images are not shown.

Finally, it can be concluded that both Metric Based ICP and Point-to-Line ICP returns good results in estimating the unknown rototranslation. The numerical results are enlisted in the Tables 7.2 and 7.3. The reader should note that the two methods have theoretically different pros and contras. The Metric Based ICP is implemented using a point-to-point minimization metric, even if, in the scientific paper written by Minguez et al. [53], the point-to-line metric is briefly introduced. This implementation choice makes the Point-to-Line ICP more accurate if the segment could be estimated with precision. However, a wrong estimation of the normal to segment (e.g., caused by a noisy laser scan) could corrupt the estimation process. Moreover, the Metric Based ICP, even if is theoretically slower than Point-to-Line ICP, should be more robust towards uncertainty in the rotational components thanks to the particular distance metric employed.

In terms of convergence speed, all the algorithm are considered fast enough even if ClassicICP is certainly the slowest.



(a) Rawseeds 1 (b) Rawseeds 2

Figure 7.4: Rawseeds test cases for scan matching

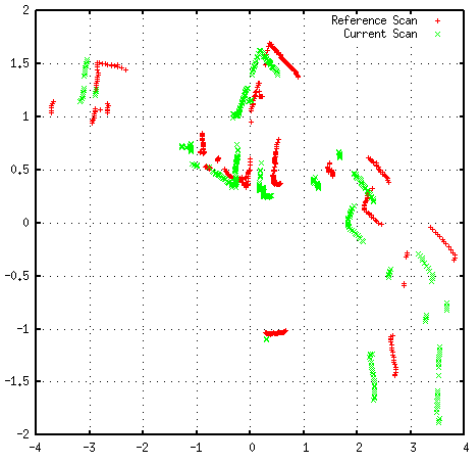
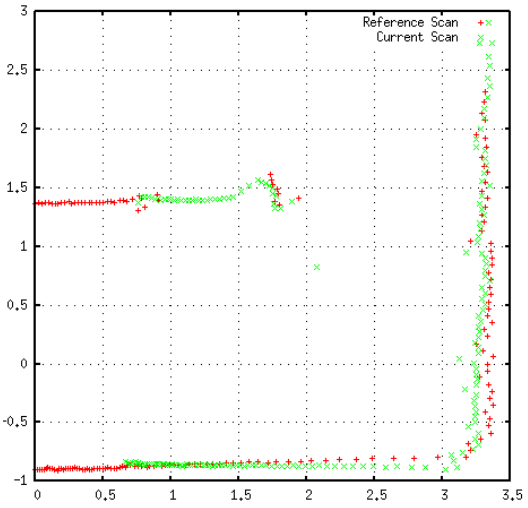
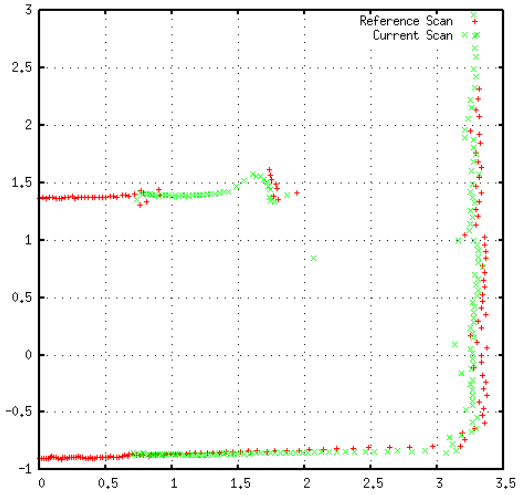


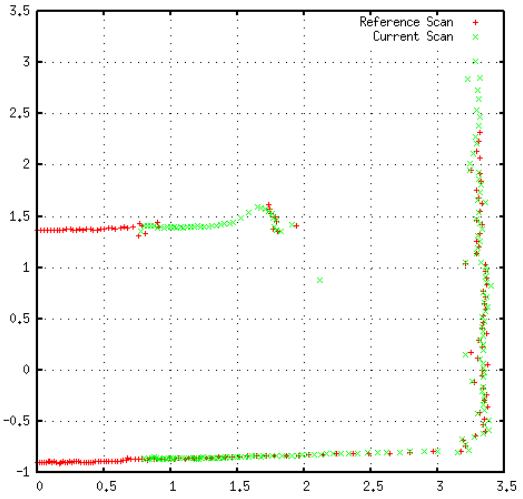
Figure 7.5: Hokuyo



(a) Classic ICP

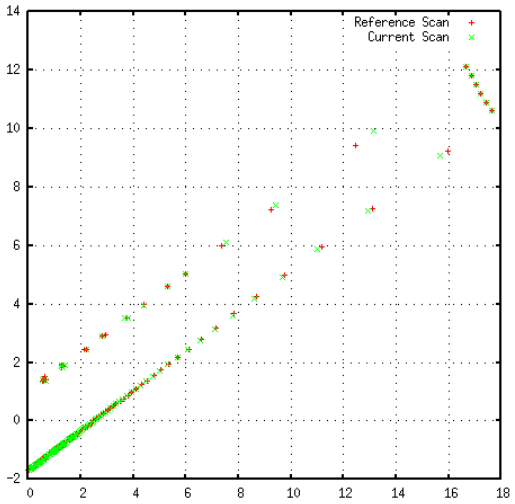


(b) Metric Based ICP

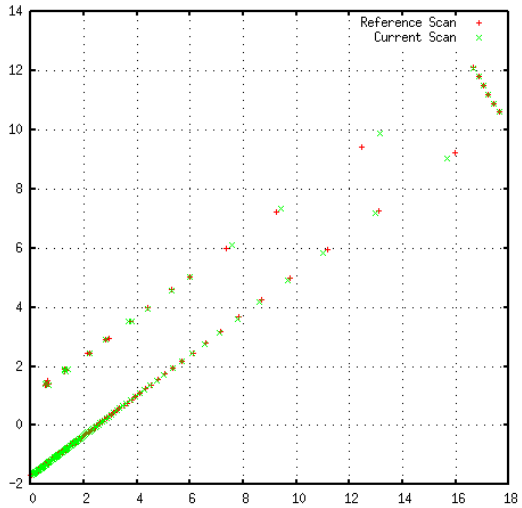


(c) Point-to-Line ICP

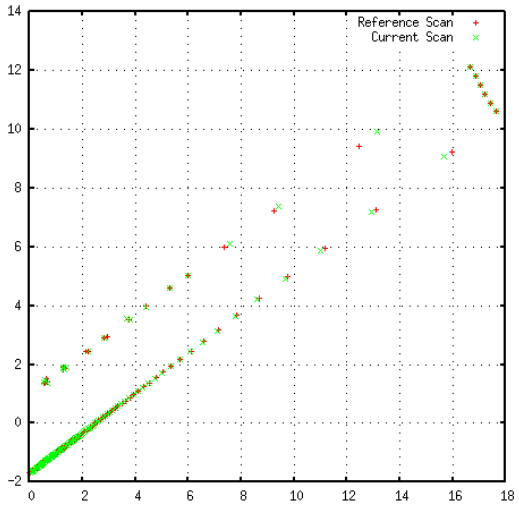
Figure 7.6: Rawseeds 1: Results



(a) Classic ICP

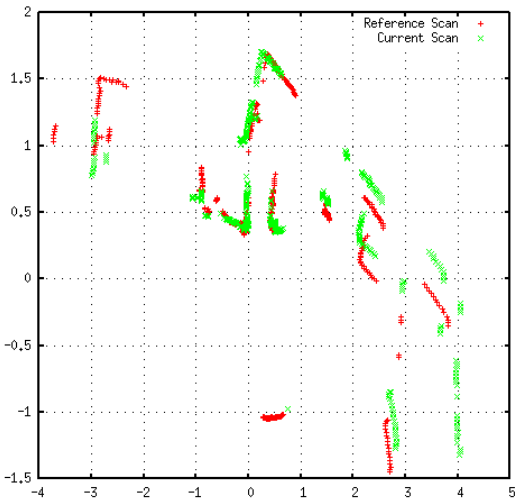


(b) Metric Based ICP

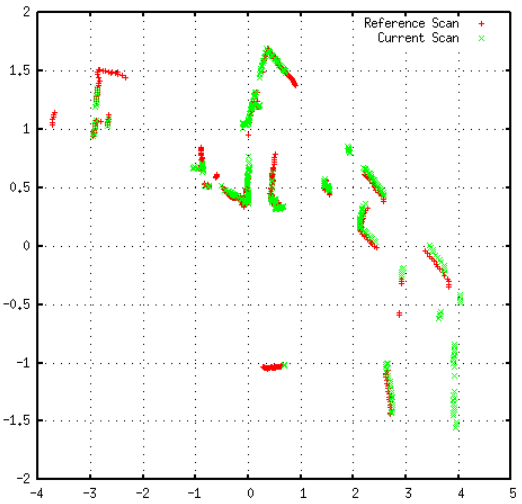


(c) Point-to-Line ICP

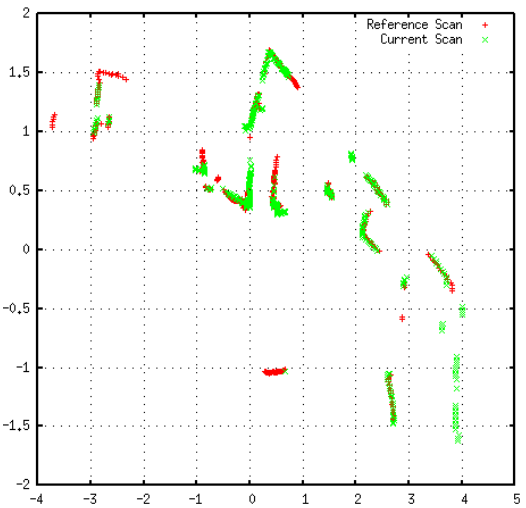
Figure 7.7: Rawseeds 2: Results



(a) Classic ICP

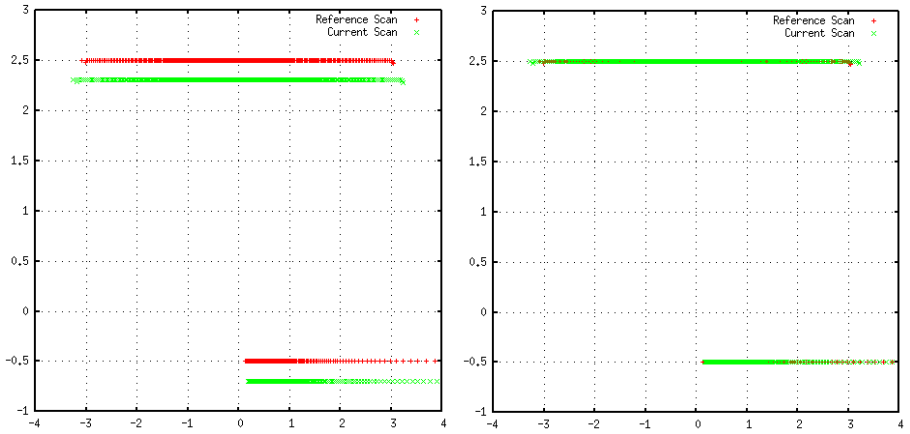


(b) Metric Based ICP



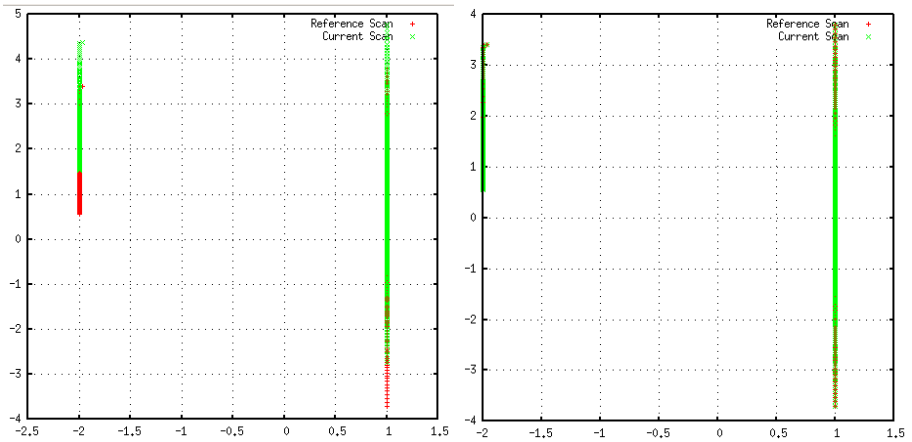
(c) Point-to-Line ICP

Figure 7.8: Hokuyo: Results



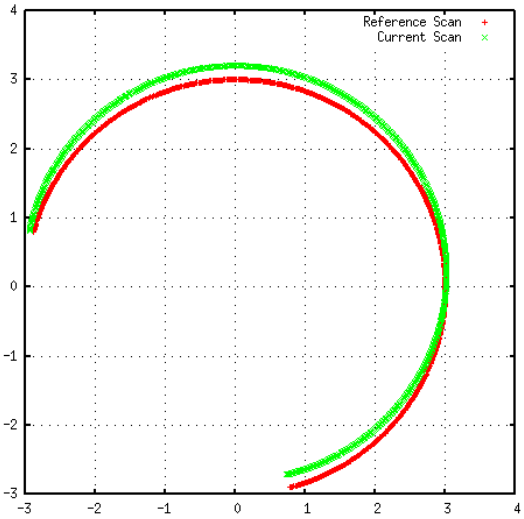
(a) Without Scan Matching (b) Point-to-Line ICP

Figure 7.9: Corridoiox: Results

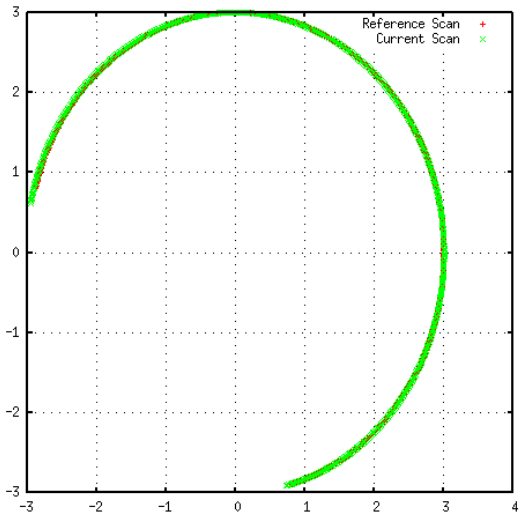


(a) Without Scan Matching (b) Point-to-Line ICP

Figure 7.10: Corridoioy: Results



(a)



(b) Point-to-Line ICP

Figure 7.11: Circular Environment: Results

Iterative Closest Point Covariance

In order to implement the scanSLAM algorithm which is based on the Extended Kalman Filter, it is necessary to correctly estimate the uncertainty of the measurement model R_t which results equal to the uncertainty associated to the error in the minimization procedure. Many techniques are implemented for each scan matcher in order to evaluate the best one. All the analytical methods cited in Chapter 3 are tested. For all the scan matchers the correct Jacobian and Hessians are derived using Matlab and its functions for symbolic computation. The formulae obtained, typically very lengthy, are then written in C++ code without further optimizations. For the Classic ICP scan matcher, two further algorithms are tried, Bengtsson's covariance estimation computed using a Numeric Hessian estimation (named in the tables NumBeng) and the first approach to ICP covariance create by Lu and Milios. The computation of the Hessian matrix is explained in Appendix B.

Lu and Milios' covariance is computed linearising the classic Euclidean error function for small value of θ and using the results as the M matrix of the Hessian Algorithm 3.3.1.

$$\begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{7.1}$$

$$\begin{pmatrix} x'_i - x_i \\ y'_i - y_i \end{pmatrix} = \begin{pmatrix} 1 & 0 & -y_i \\ 0 & 1 & x_i \end{pmatrix} \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = M \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{7.2}$$

Bosse's Covariance method was implemented to empirically confirm an observation that it is later explained. It is chosen to keep the formulae in the equations equal for all the scan matchers. The implementation reflects the scientific paper [14] where:

$$H_i = \begin{pmatrix} -n_x \\ -n_y \\ \begin{pmatrix} n_x \\ n_y \end{pmatrix} \begin{pmatrix} sin(\theta) & cos(\theta) \\ -cos(\theta) & sin(\theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \end{pmatrix}^T \tag{7.3}$$

where H_i is the i^{th} row of the matrix H, (x_i, y_i) is a point of the current scan and n is the normal to the segment obtained by the two nearest neighbours.

The results are shown in Tables 7.2 and 7.3. Using the data it is possible to conduct an empirical analysis about the performance of the algorithms for the estimation of the error covariance.

Criteria for Performing the Analysis of ICP Covariances The comparison between resulting values computed by the algorithm of ICP covariance is theoretically possible only for simulated data since the true value is known. The criterion used in the following analysis is based on a heuristic. By visual inspection, the best scan matcher estimate is found and its value is hypothesised to be the correct one. In this way, as figure of merit beside the visual inspection itself, it is considered the distance in standard deviation, σ, of each estimate retrieved by the scan matcher with respect to the reference one. Furthermore, it is also hypothesised that the covariance matrix is diagonal, and so σ_x, σ_y, σ_θ are independent, which is a tolerable approximation in practice.

We can conclude that an estimate $RT^{method2}$ is in accordance with the reference estimate $RT^{method1}$ if the following equality holds:

$$RT_i^{method2} = RT_i^{method1} \pm k\sigma_i, \ with\ i \in \{x, y, \theta\}\ and\ k \in \{0..3\} \quad (7.4)$$

Analysis of ICP Covariances In well-constrained test cases, like Rawseeds 1 and 2 and Hokuyo, it is possible to check the correctness of the estimate selecting the estimate of the transformation RT given by Point-to-Line metric as reference. In fact, it is always visually correct. In particular, from the previous analysis it is shown that the Metric ICP and Classic ICP perform worse than Point-to-Line ICP in Rawseeds 1 and in Hokuyo test case.

For the test case named Rawseeds 1, it can be observed that the covariance evaluated for Metric ICP for Bengtsson and Censi's formulation is well estimated since it is possible to write:

$$RT_i^{PLICP} = RT_i^{MetricICP} \pm k\sigma_i, \ with\ i \in \{x, y, \theta\}\ and\ k \in \{0..3\} \quad (7.5)$$

An exception is done for the x component, since the numbers are very close. For the same test case Classic ICP performs well with Bengtsson's both analytic and numeric and Lu Milios'. In particular, Lu Milios' estimate is comparable to analytic Bengtsson's one and numeric Bengtsson's returns the best estimate. Censi's estimate seems to underestimate the covariance. It should be noted that the value σ^2 in Censi's Algorithm 3.3.3 is set to $(20mm)^2$ that is the estimate accuracy for the worst laser in use (Hokuyo laser range finder).

The same considerations can be applied for the test case named Hokuyo where Bengtsson's estimate is the best one.

For the under-constrained cases, which are the most important to detect for assuring the correctness of the Kalman Filter, the results are not good as one could expected. In particular, for Corridor X, only two methods

seem correct: Bosse's method for Classic ICP returns good results even if overestimate the covariance in the θ component, while the Censi's method applied to Point-to-Line ICP obtains the most correct ones. It should be observed that the results of Bengtsson's method are partially in conflict with the scientific paper written by Censi [17] where also that method obtains good performance. Moreover, the reader should consider that theoretically the manifold related to the x axis is not observable and, for this reason, the results in that component should not be taken for granted (in fact, it should be infinite). In any case, the detectability of this event is amenable if the scan matcher is employed in a filtering framework such as EKF SLAM.

Corridor Y, presents the same characteristic of Corridors X and the results and considerations are the same. In the Circular Environment, Bosse's method gives good results for Classic ICP and Metric ICP, while Point-to-Line ICP Censi's method results the best one. It is interesting to underline the NaN values in Bengtsson's estimate that in this case can be a symptom of an under-constraint situation. Nevertheless, the NaN value is also present in Hokuyo test case without any evident meaning.

A possible reason for the failure of all or most of the algorithm for the estimation of the covariance is that the derivatives, Jacobians and Hessians, cancels some coefficient or create constants, masking the differences between the constraint and under-constraint situations. An example is shown in Lu Milios' formula where the dependences on $\begin{pmatrix} x'_i & y'_i \end{pmatrix}^T$ cancel out. For this reason, Bosse's covariance is tested using the point-to-line distance metric which, thanks to the dot product between the normal n and the Euclidean distance metric (see Chapter 3 in section Point-to-Line ICP), excludes the elimination of important coefficients. In fact, it is shown that even for the under-constrained cases Bosse's method performs better than Bengtsson's and Censi's method for Classic and Metric Based ICP. However, the overestimation in σ_θ shows that it is not possible to interchange this method with the more formally correct ones which are derived properly from their respective error functions.

Dataset	*ScanMatcher*	*RT*	*Censi*	*Bengtsson*	*Bosse*	*NumBeng*	*LuMilios*
Rawseeds1	*ClassicICP*	$\begin{pmatrix}0.6926\\0.03377\\-0.0478\end{pmatrix}$	$\begin{pmatrix}0.00378\\0.007594\\0.1634\end{pmatrix}$	$\begin{pmatrix}0.0101\\0.01364\\0.3773\end{pmatrix}$	$\begin{pmatrix}0.319\\0.1872\\19.94\end{pmatrix}$	$\begin{pmatrix}0.02672\\0.01749\\0.5851\end{pmatrix}$	$\begin{pmatrix}0.01014\\0.01706\\0.3769\end{pmatrix}$
Rawseeds1	*MetricICP*	$\begin{pmatrix}0.7034\\0.01653\\-0.01967\end{pmatrix}$	$\begin{pmatrix}0.007713\\0.01476\\0.7049\end{pmatrix}$	$\begin{pmatrix}0.01283\\0.01651\\0.9563\end{pmatrix}$	$\begin{pmatrix}0.1838\\0.1259\\8.745\end{pmatrix}$	–	–
Rawseeds1	*PLICP*	$\begin{pmatrix}0.7772\\0.01722\\0.001388\end{pmatrix}$	$\begin{pmatrix}0.006013\\0.007222\\0.2477\end{pmatrix}$	$\begin{pmatrix}0.02108\\0.0227\\0.6973\end{pmatrix}$	$\begin{pmatrix}0.02581\\0.01613\\1.574\end{pmatrix}$	–	–
Rawseeds2	*ClassicICP*	$\begin{pmatrix}0.01062\\-0.005052\\0.01799\end{pmatrix}$	$\begin{pmatrix}0.002981\\0.003313\\0.01758\end{pmatrix}$	$\begin{pmatrix}0.00735\\0.008453\\0.08259\end{pmatrix}$	$\begin{pmatrix}0.1241\\0.07611\\8.056\end{pmatrix}$	$\begin{pmatrix}0.009693\\0.009307\\0.08027\end{pmatrix}$	$\begin{pmatrix}0.007349\\0.008461\\0.08259\end{pmatrix}$
Rawseeds2	*MetricICP*	$\begin{pmatrix}0.01171\\-0.00504\\0.01728\end{pmatrix}$	$\begin{pmatrix}0.006821\\0.01076\\0.4838\end{pmatrix}$	$\begin{pmatrix}0.01124\\0.01536\\0.9274\end{pmatrix}$	$\begin{pmatrix}0.1168\\0.07923\\5.415\end{pmatrix}$	–	–
Rawseeds2	*PLICP*	$\begin{pmatrix}0.01288\\0.0002494\\0.01761\end{pmatrix}$	$\begin{pmatrix}0.004301\\0.002547\\0.01106\end{pmatrix}$	$\begin{pmatrix}0.07847\\0.08132\\0.4607\end{pmatrix}$	$\begin{pmatrix}0.2198\\0.1347\\14.26\end{pmatrix}$	–	–
Hokuyo	*ClassicICP*	$\begin{pmatrix}0.3033\\0.07281\\0.1345\end{pmatrix}$	$\begin{pmatrix}0.003344\\0.003296\\0.06482\end{pmatrix}$	$\begin{pmatrix}0.008109\\0.00809\\0.2799\end{pmatrix}$	$\begin{pmatrix}0.2307\\0.4754\\18.07\end{pmatrix}$	$\begin{pmatrix}0.01056\\0.01271\\0.3499\end{pmatrix}$	$\begin{pmatrix}0.008226\\0.00863\\0.2835\end{pmatrix}$
Hokuyo	*MetricICP*	$\begin{pmatrix}0.2983\\0.05177\\0.07762\end{pmatrix}$	$\begin{pmatrix}0.008215\\0.006494\\0.3379\end{pmatrix}$	$\begin{pmatrix}0.006567\\0.005569\\0.3902\end{pmatrix}$	$\begin{pmatrix}0.0571\\0.0242\\2.288\end{pmatrix}$	–	–
Hokuyo	*PLICP*	$\begin{pmatrix}0.2937\\0.04069\\0.06132\end{pmatrix}$	$\begin{pmatrix}0.09496\\0.1059\\0.1733\end{pmatrix}$	$\begin{pmatrix}0.1777\\0.1711\\NaN\end{pmatrix}$	$\begin{pmatrix}0.1854\\0.2107\\7.948\end{pmatrix}$	–	–

Table 7.2: Table: Rawseeds 1, 2 and Hokuyo. In the table the transformation estimate is written in column RT in the format $([m],[m],[rad])$ while for the covariance matrix, only $(\sigma x, \sigma y, \sigma\theta)$ is reported, in $([m],[m],[deg])$. The angle conversion is chosen for simplify the visual inspection.

Dataset	*ScanMatcher*	*RT*	*Censi*	*Bengtsson*	*Bosse*	*NumBeng*	*LuMilios*
CorridorX	*ClassicICP*	$\begin{pmatrix}-0.06354\\0.1997\\-8.337\text{E-}5\end{pmatrix}$	$\begin{pmatrix}0.001166\\0.001127\\0.01695\end{pmatrix}$	$\begin{pmatrix}0.00107\\0.0009439\\0.02578\end{pmatrix}$	$\begin{pmatrix}0.318\\0.03014\\3.501\end{pmatrix}$	$\begin{pmatrix}0.004918\\0.0009578\\0.03394\end{pmatrix}$	$\begin{pmatrix}0.001117\\0.0009385\\0.02579\end{pmatrix}$
CorridorX	*MetricICP*	$\begin{pmatrix}-0.06411\\0.1995\\2.232\text{E-}5\end{pmatrix}$	$\begin{pmatrix}0.001982\\0.002396\\0.0643\end{pmatrix}$	$\begin{pmatrix}0.0008263\\0.0008082\\0.05007\end{pmatrix}$	$\begin{pmatrix}0.03261\\0.01776\\2.003\end{pmatrix}$	–	–
CorridorX	*PLICP*	$\begin{pmatrix}-0.02312\\0.2\\8.933\text{E-}6\end{pmatrix}$	$\begin{pmatrix}0.1257\\0.005018\\0.1005\end{pmatrix}$	$\begin{pmatrix}0.006064\\0.0007322\\0.01491\end{pmatrix}$	$\begin{pmatrix}0.008827\\0.0008375\\0.09722\end{pmatrix}$	–	–
CorridorY	*ClassicICP*	$\begin{pmatrix}0.0002433\\0.09924\\0.0003765\end{pmatrix}$	$\begin{pmatrix}0.0009915\\0.0008136\\0.008086\end{pmatrix}$	$\begin{pmatrix}0.0004522\\0.0004489\\0.01323\end{pmatrix}$	$\begin{pmatrix}0.01863\\0.2252\\2.174\end{pmatrix}$	$\begin{pmatrix}0.0004574\\0.007411\\0.01727\end{pmatrix}$	$\begin{pmatrix}0.0004586\\0.0004502\\0.01327\end{pmatrix}$
CorridorY	*MetricICP*	$\begin{pmatrix}0.0001572\\0.07213\\0.0003106\end{pmatrix}$	$\begin{pmatrix}0.002035\\0.001973\\0.05255\end{pmatrix}$	$\begin{pmatrix}0.0004845\\0.0005251\\0.03101\end{pmatrix}$	$\begin{pmatrix}0.01422\\0.02727\\1.635\end{pmatrix}$	–	–
CorridorY	*PLICP*	$\begin{pmatrix}1.521e-06\\0.9666\\-2.616e-06\end{pmatrix}$	$\begin{pmatrix}0.003373\\0.5466\\0.7483\end{pmatrix}$	$\begin{pmatrix}0.006461\\0.02418\\0.1605\end{pmatrix}$	$\begin{pmatrix}0.004674\\0.02639\\0.4748\end{pmatrix}$	–	–
Circular	*ClassicICP*	$\begin{pmatrix}-0.003696\\-0.2031\\0.01862\end{pmatrix}$	$\begin{pmatrix}0.001007\\0.0008751\\0.008582\end{pmatrix}$	$\begin{pmatrix}0.0009427\\0.0009278\\0.01835\end{pmatrix}$	$\begin{pmatrix}0.08031\\0.0826\\3.477\end{pmatrix}$	$\begin{pmatrix}0.001317\\0.001287\\0.07927\end{pmatrix}$	$\begin{pmatrix}0.0009246\\0.0009291\\0.0184\end{pmatrix}$
Circular	*MetricICP*	$\begin{pmatrix}-0.003457\\-0.2029\\0.01859\end{pmatrix}$	$\begin{pmatrix}0.00182\\0.001873\\0.05066\end{pmatrix}$	$\begin{pmatrix}0.000438\\0.0004419\\0.02716\end{pmatrix}$	$\begin{pmatrix}0.0275\\0.02696\\1.179\end{pmatrix}$	–	–
Circular	*PLICP*	$\begin{pmatrix}-0.0111\\-0.2005\\-0.001187\end{pmatrix}$	$\begin{pmatrix}0.0333\\0.001827\\7.937\end{pmatrix}$	$\begin{pmatrix}NaN\\0.0004301\\NaN\end{pmatrix}$	$\begin{pmatrix}0.001432\\0.001445\\0.06141\end{pmatrix}$	–	–

Table 7.3: Table: Corridor x,y and Circular (environment). In the table the transformation estimate is written in column RT in the format $([m],[m],[rad])$ while for the covariance matrix, only $(\sigma x, \sigma y, \sigma\theta)$ is reported, in $([m],[m],[deg])$. The angle conversion is chosen for simplify the visual inspection.

7.2.2 ScanSLAM

In this section, the results of the implementation of the scanSLAM algorithms using both SICK and Hokuyo lasers are shown. The modification of the algorithm for calibration purpose is also discussed to investigate the reasons of failure.

Motion Model

The motion model is the last component of the EKF that should be properly characterized in order to implement the scanSLAM algorithm. For simplicity, the errors are estimated using only the parameters α_1 and α_3 of the model explained in Chapter 4 while, the other parameters are set to zero. This choice implies that no error is added to the translation components due to rotations and, conversely, that the uncertainty due to translation does not affect the rotation components.

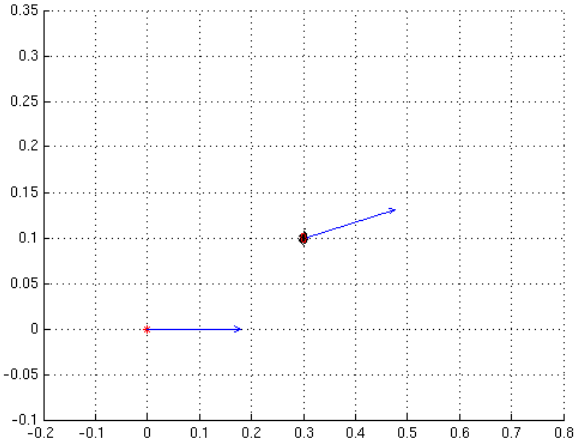
In Figures 7.12 and 7.13, the reader can observe how the values of those parameters affect the covariance related to the uncertainty of the motion model. In this plots, the motion starts from the robot pose $(0, 0, 0)$[2] and ends at the robot pose $(0.3, 0.1, 10)$. The graphics are empirically classified into two categores: the ones having low error in the motion model and the others which shares high errors in the motion model.

Recalling the motion model equation in Chapter 4 the meaning of the parameters can be explained in this way:

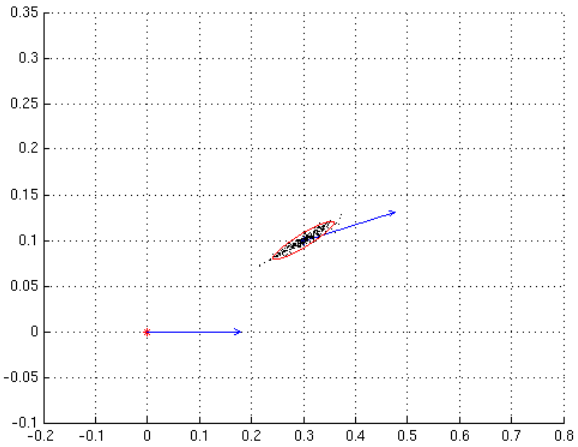
- $\alpha_1 = \frac{x}{180}$ means that if the robot rotates itself by 180° then $\sigma_{rot} = x$
- $\alpha_3 = x$ means that if the robot translated itself by 1 metre then $\sigma_{trans} = x$

This parameters are used in the implementation of the scanSLAM filter.

[2]The units of measurement are $([m], [m], [deg])$

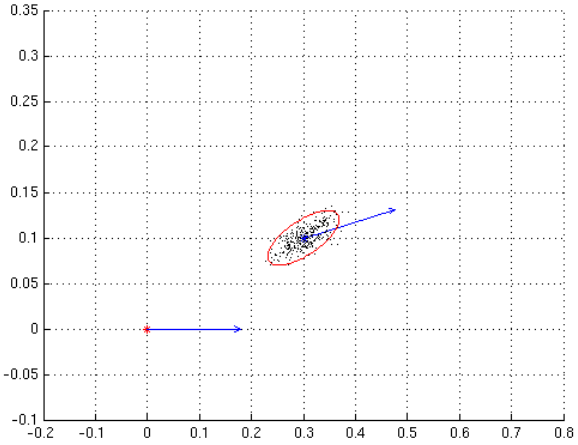


(a) $\alpha_1 = \frac{5}{180}$, $\alpha_3 = 0.01$

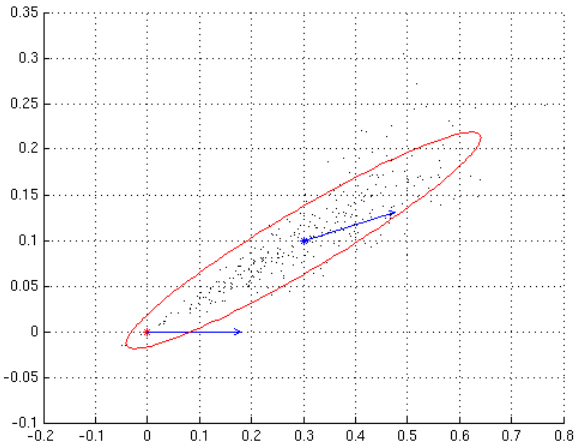


(b) $\alpha_1 = \frac{5}{180}$, $\alpha_3 = 0.1$

Figure 7.12: Motion Models: Low Errors



(a) $\alpha_1 = \frac{15}{180}$, $\alpha_3 = 0.1$



(b) $\alpha_1 = \frac{35}{180}$, $\alpha_3 = 0.5$

Figure 7.13: Motion Models: High Errors

Model Simulation

The scanSLAM model is simulated under Matlab in order to get insight on possible sources of errors and outcomes. Since the objective is to check the behaviour of the EKF and not the scan matcher correctness, the landmarks are simply defined by the state variables that represents the pose of the robot at the moment the landmark is saved.

A new landmark is acquired when the Euclidean distance of the pose of the robot to all the landmark is above 0.5 metres. The measurement z_k is obtained from the true measurement z_{real} computed using the correct robot pose with the addition of a white noise whose covariance is used as measurement uncertainty R_k.

In order to minimize the sources of errors, the matrix M is set to zero and R_k is:

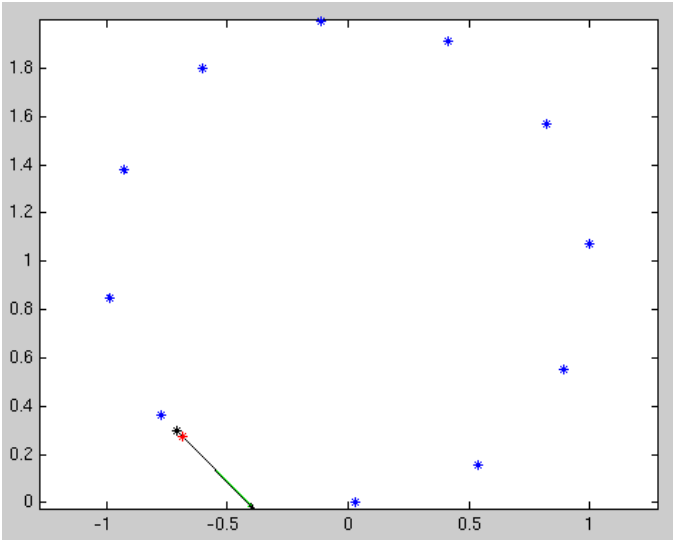
$$R_k = \begin{pmatrix} (0.005)^2 & 0 & 0 \\ 0 & (0.005)^2 & 0 \\ 0 & & (0.01\frac{\pi}{180})^2 \end{pmatrix} \tag{7.6}$$

The results, in Figures 7.14, show that the canonic scanSLAM approach performs well. In the figure, the robot describes a circle which is perfectly drawn since the map estimation is consistent. The simulation has been already explained in Chapter 5 under the section named Loop Closure. The main interest in performing this simulation is to test the possibility to calibrate the extrinsic parameters which comprise the rototranslation between the laser and the odometric centre of the robot. The results are shown in Table 7.4 and in Figure 7.15. From the table it can be seen that the calibration covariance does not converge to zero but seems to remain stable to a particular value. This is probably due to the fixed value of the covariance R_k. It should be noted that particular movements, such as only translational ones, do not properly excite the translational parameters (x, y).

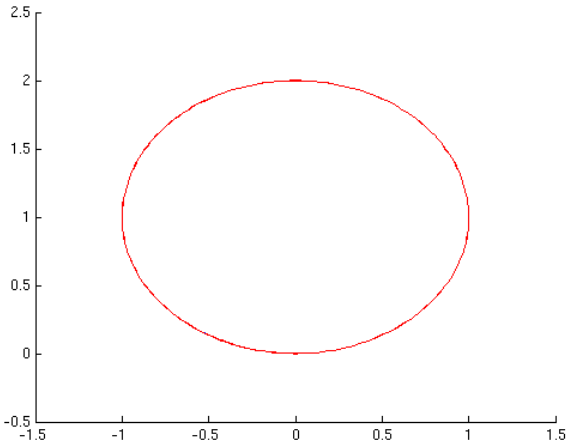
Though the result is limited to the fixed covariances, it could be seen that the estimate of the covariance given in the table are in accordance to the true ones. Therefore, it can be concluded that the calibration procedure, using this method, is pursuable.

$Parameter$	$Measured$	$It = 1$	$It = end$	σ_1	σ_{end}
x	0.8	1.0453	0.8289	0.2	0.0344
y	0.25	0.1593	0.2826	0.2	0.0395
θ	-1.2217	-1.1849	-1.1818	0.0524	0.0478

Table 7.4: In the table are shown the (extrinsic) rototranslation parameters (t_x, t_y, t_θ), their measured value, initial value, the value at first and last iterations and the standard deviation

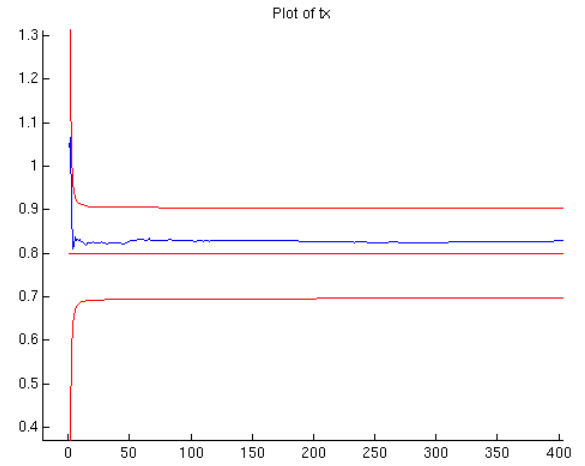


(a)

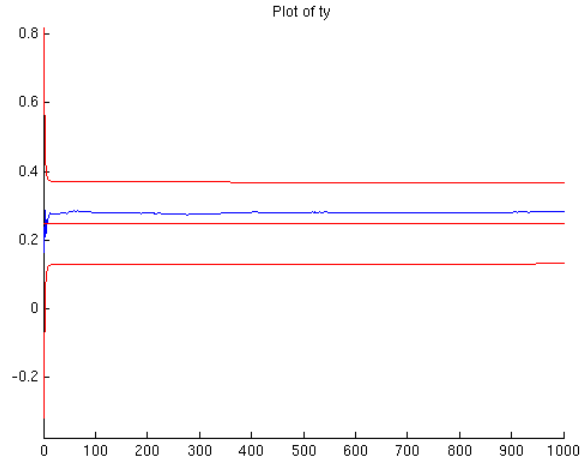


(b)

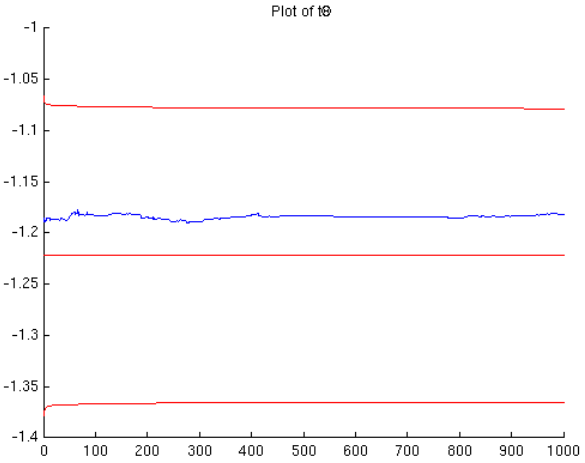
Figure 7.14: scanSLAM Model Simulation



(a)



(b)



(c)

Figure 7.15: scanSLAM Model Simulation: Calibration Results. The red line in the middle is the real value of the parameter. The others refer to 3σ from the real value

Rawseeds Dataset

A portion of a session of a Rawseeds dataset acquired in Bicocca University (see also Chapter 6) is used to test the scanSLAM capability and the calibration extension. The map of the environment using odometric data only is shown in Figure 7.16, while Figure 7.17 represents an incremental mapping approach.

The incremental mapping is performed using only the results obtained by the scan matcher. The incremental pose of the robot, computed using the odometric robot pose $x_{inc} = \ominus x_{k-1} \oplus x_k$, is used to compute (using the known rototranslation between robot and laser) the guess RT_{guess}. The formula is pretty the same as the one explained in Chapter 5 for Robot Centred with One Laser and Extrinsic Calibration. The current scan is then saved in a map and used as reference scan in the next iteration. The scan matcher selected for this and the following dataset is Point-to-Line ICP because of the overall good covariance estimation.

In this case, the incremental mapping approach gives worse results than the odometry itself. This result is acceptable since if the scan matcher fails all the successive scans will be wrongly placed. Figure 7.17 shows that the algorithm fails the most in the estimation of the rotational components when entering corridors, probably because of the uniform weighting of the data. Nevertheless, if the Extended Kalman Filter is properly tuned, the filter mechanism is capable to solve this issue. Figures 7.18, 7.19, 7.20, 7.21, 7.22 show the results of scanSLAM approach without extrinsic calibration. The resulting map is consistent and is visually better than the one obtained by odometry only, see Figures 7.20 and 7.16. The specific parameters of scanSLAM algorithm are set as follows.

Motion Model The model in Figure 7.13 uses $\alpha_1 = \frac{15}{180}$, $\alpha_3 = 0.1$

Scan Matcher Point-to-Line scan matcher with Censi's ICP covariance

Extrinsic Parameters The calibration laser-robot is taken for granted $(0.08, 0, 0)$, $([m], [m], [rad])$

New Landmark A landmark is placed if the robot is more distant than 0.5m or rotates more than 35deg from the nearest landmark. Otherwise the EKF update is performed

Measurement Selection Maximum Likelihood data association is implemented as explained in Chapter 5. Only the best measurement is used in the filter update step. The Multiple Data Association approach

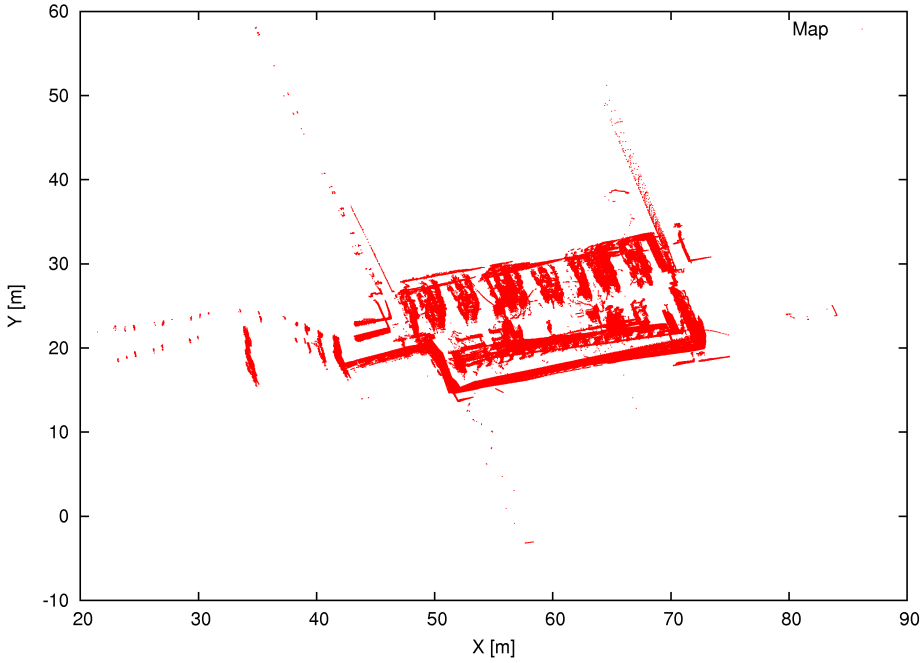


Figure 7.16: Rawseeds: Map resulting from odometric data

is not used since considering more than one landmarks increases the probability of wrong estimations of the scan matcher (this is particularly true for the next dataset, LURCH). In fact, the scan matcher typically underestimates the ICP covariance when converging to the wrong minimum

Iterations The EKF step is performed every 10 laser acquisition. The others are simply skipped.

The calibration procedure does not give successful results. In Figure 7.23 can be observed the final map obtained by using the value in Table 7.5. From the Figure 7.24 it can be noticed that, even if the values of the calibration parameters are affected by sudden variations, their value remains almost stable after some iterations. The covariance associated to each parameter decreases very fast, giving evidence to the hypothesis stated before in Model Simulation,which can be explained in the following way: since R_k has variable values (in contrast to the previous, fixed one), it is now possible for the filter to decrement the covariances associated to the extrinsic parameters (see Figure 7.25).

Although the covariances associated to the extrinsic parameters show an optimal behaviour during time, the filter converges to the wrong estimates

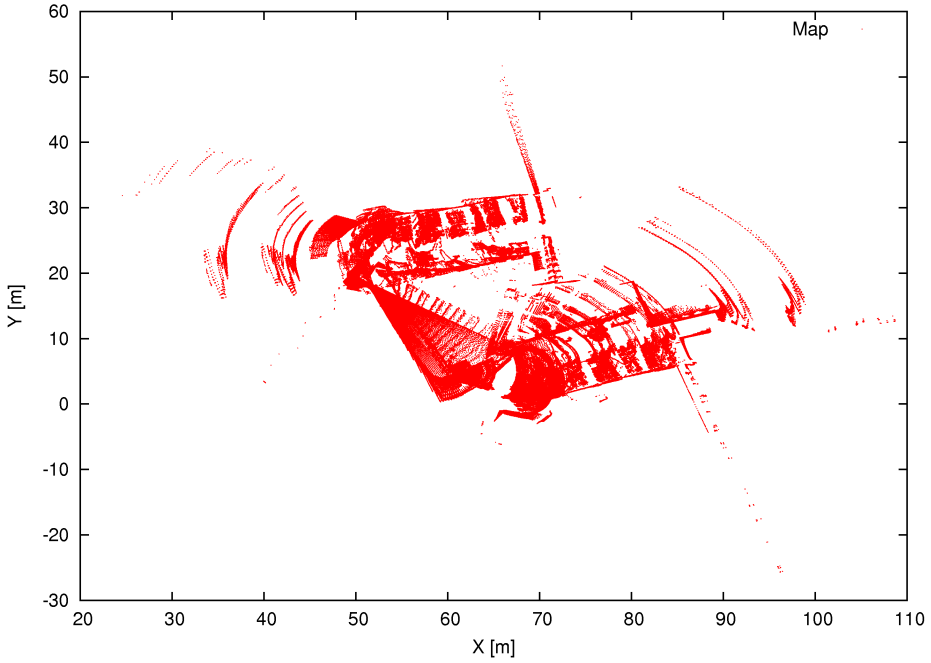
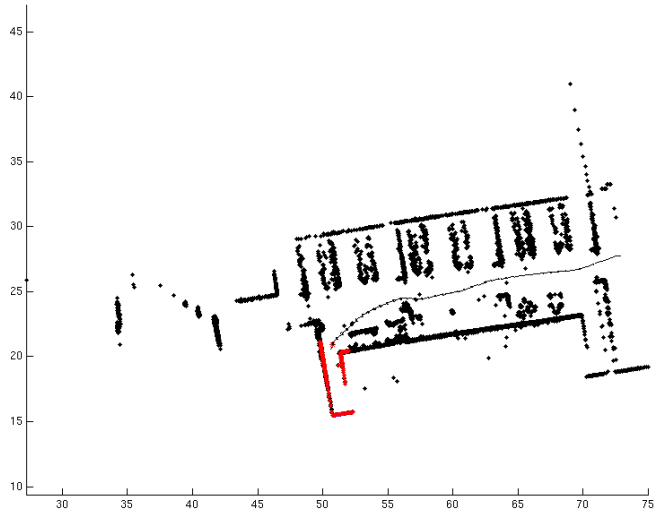


Figure 7.17: Rawseeds: Incremental Mapping

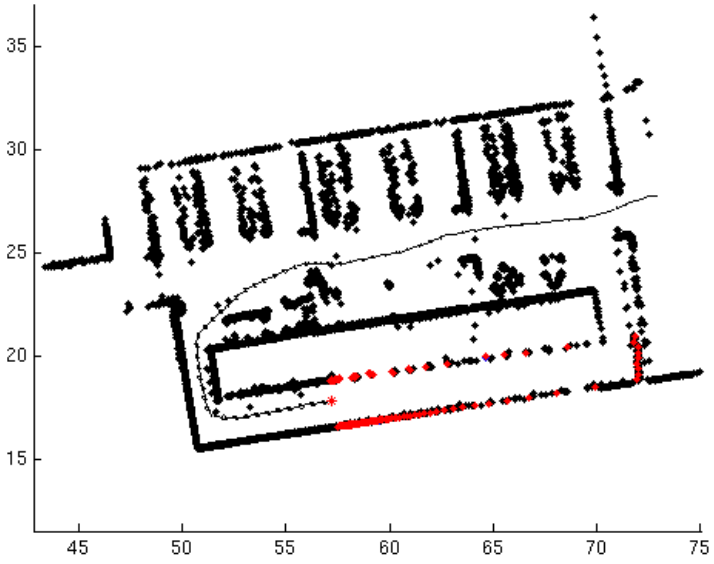
causing inconsistency in the map. The reasons for this result should be searched in the non observability or excitability of the parameters, the big initial covariance with respect to the accuracy of the parameters and the poor motion model in use. In fact, the probabilistic motion models does not take into account the robot speed that should remain consistent during the robot motion with the prevention of phenomena such as sudden changes in robot pose.

$Parameter$	$Measured$	$It = 1$	$It = end$	max	min	σ_1	σ_{end}
x	~ 0.08	0.08	0.069	0.6045	0.0428	0.1	0.0029
y	~ 0	0	-0.0126	0.0113	-0.0991	0.1	0.003
θ	~ 0	0	0.0039	0.1865	-0.0106	$5\frac{\pi}{180}$	$8.6087e-04$

Table 7.5: In the table are shown the rototranslation parameters (x, y, θ), their measured, initial value, the value at first and last iterations, their maximum and minimum value and the standard deviation

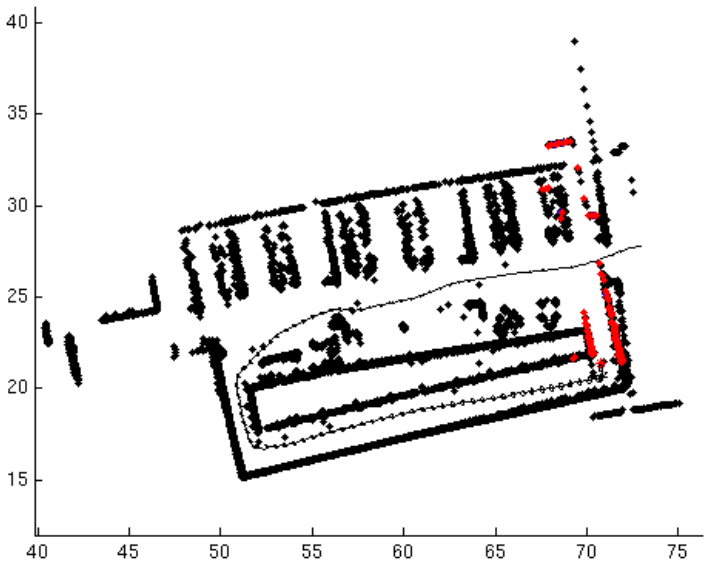


(a) Iteration number 500

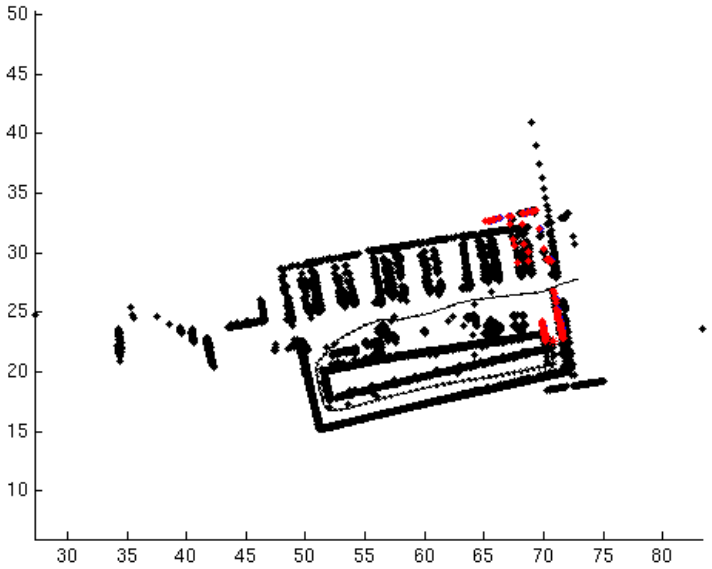


(b) Iteration number 700

Figure 7.18: Rawseeds: without calibration.

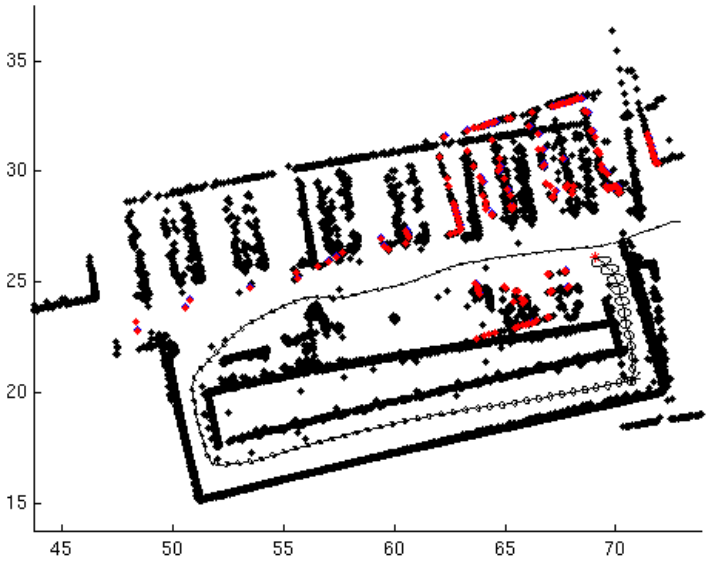


(a) Iteration number 900

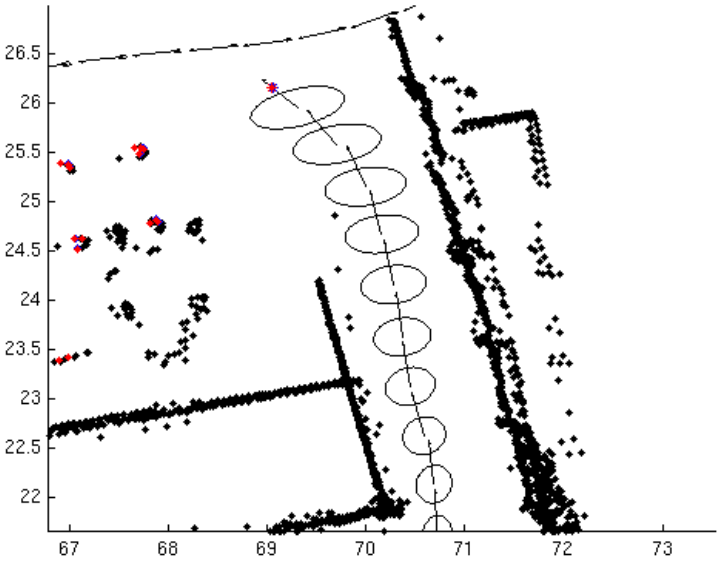


(b) Iteration number 1000

Figure 7.19: Rawseeds: without calibration. The robot pose at Frame a is wrong estimated, since it is still an open loop it is difficult for the robot to recover the correct pose.

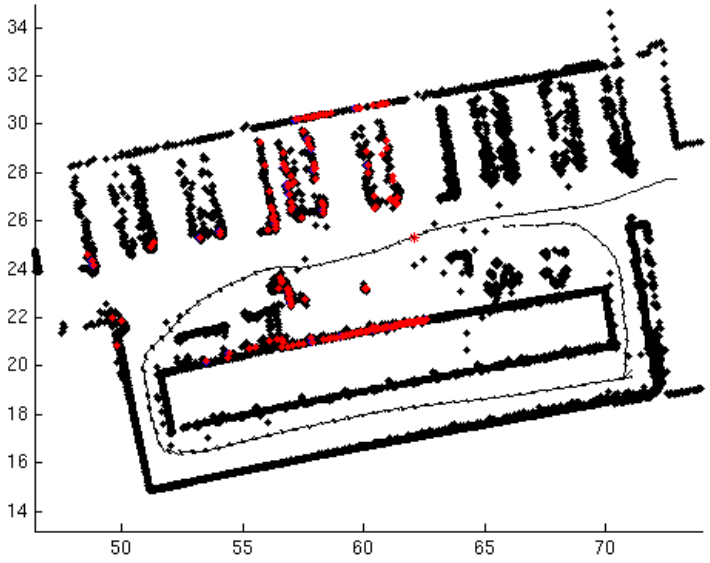


(a) Iteration number 1323

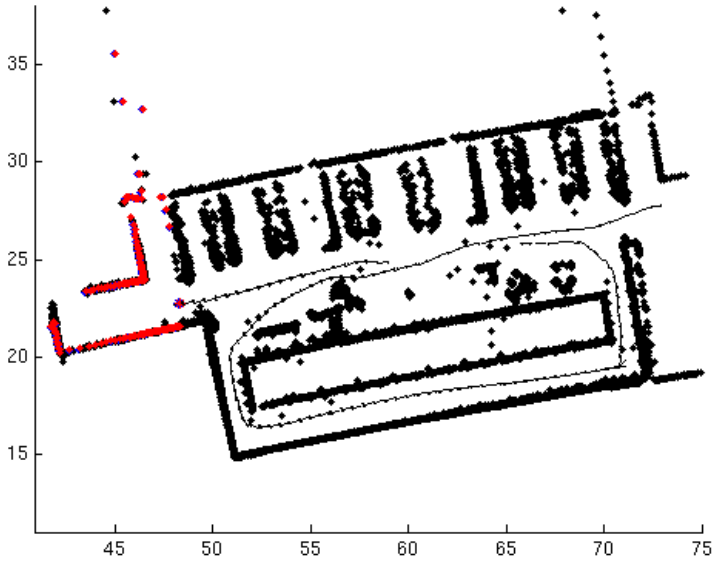


(b) Iteration number 1323

Figure 7.20: Rawseeds: without calibration. The robot is approaching to a loop closure

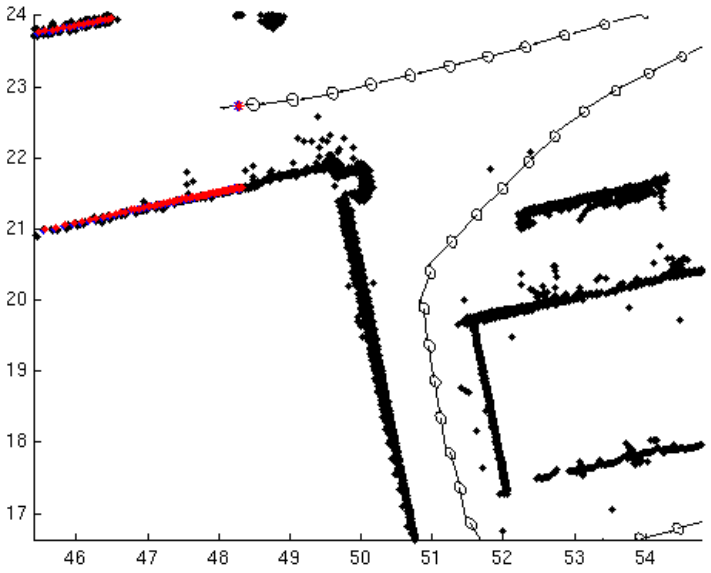


(a) Iteration number 1400

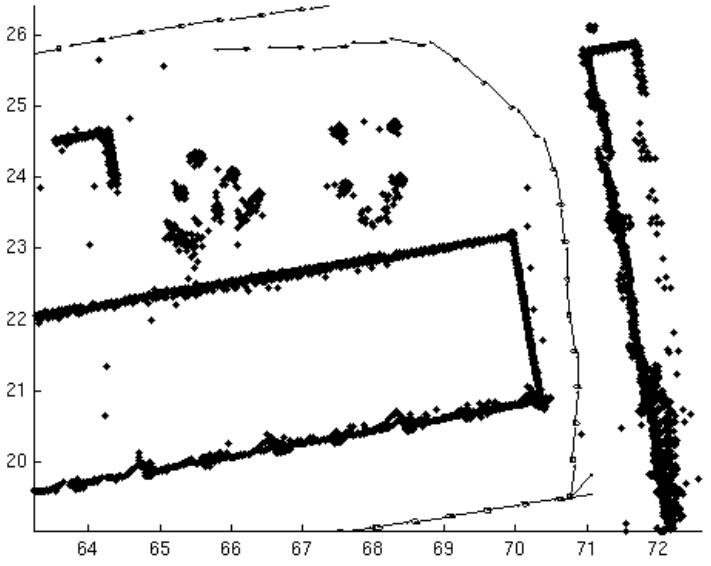


(b) Iteration number 2100

Figure 7.21: Rawseeds: without calibration. The robot closes the loop, the resulting map is consistent

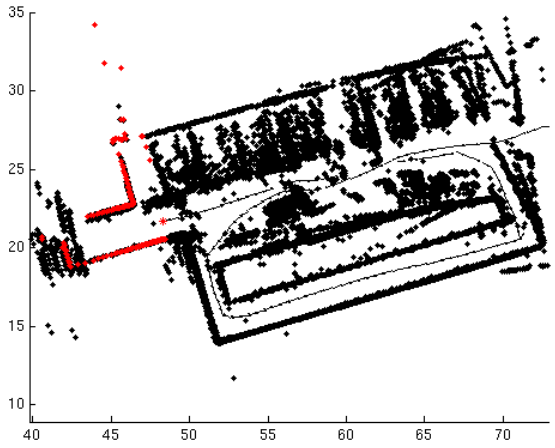


(a) Iteration number 2100

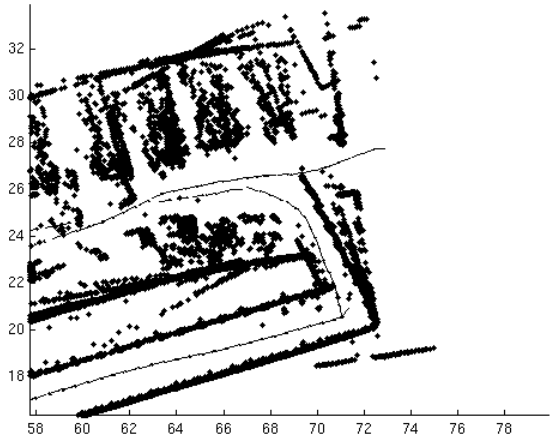


(b) Iteration number 2100

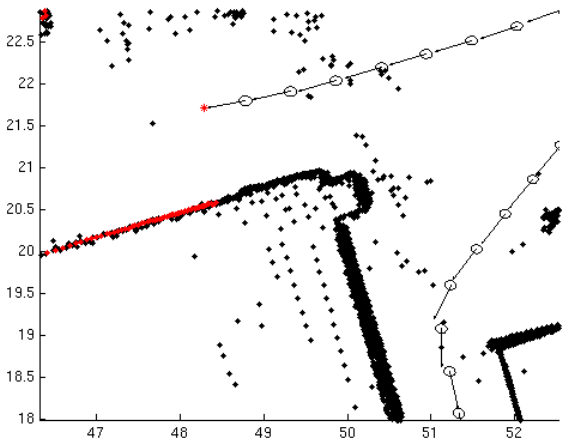
Figure 7.22: Rawseeds: without calibration. Some details, it can be observed that the covariances related to the initial landmarks, in Frame a, are now very small in comparison to the latest ones, in Frame b



(a) Iteration number 2100



(b) Iteration number 2100



(c) Iteration number 2100

Figure 7.23: Rawseeds: with calibration

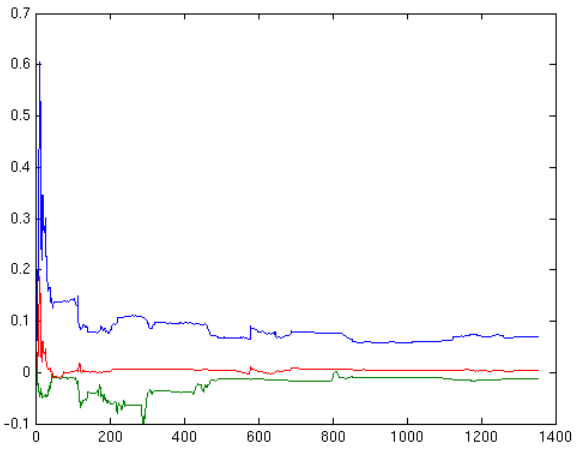
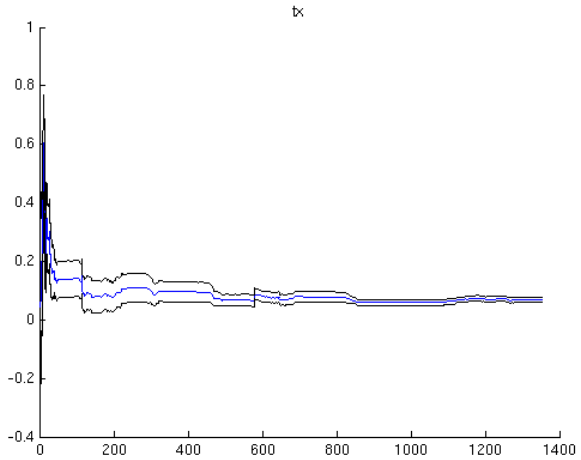
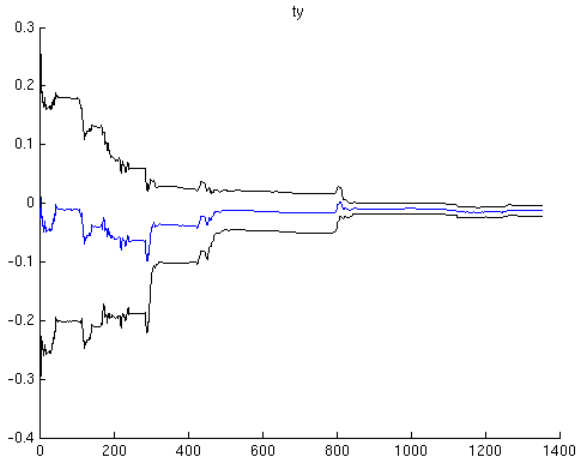


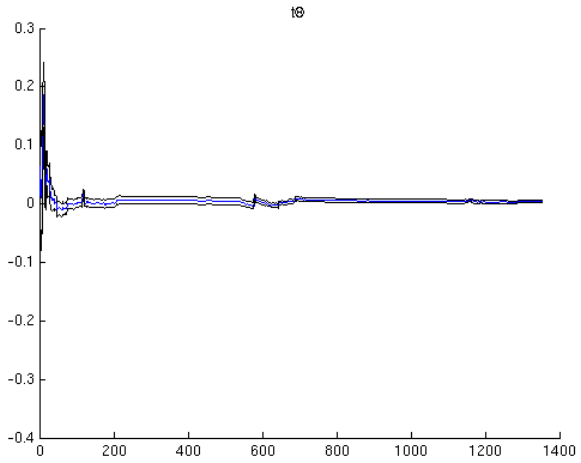
Figure 7.24: Rawseeds: calibration parameters



(a)



(b)



(c)

Figure 7.25: Rawseeds: calibration parameters

LURCH Dataset

The LURCH dataset is challenging because of the use of Hokuyo laser range finders which has lower scanning speed and lower accuracy in comparison to SICK lasers. Moreover, the lateral placement of the scan is not optimal since it is not on the principal axis of wheelchair movements. The maximum speed of the wheelchair can cause also the problems described in Chapter 5 under the section High Speed Rotations.

Figure 7.26 shows the map of the AIRLab laboratory as can be estimated using only the odometric data given by the encoders on the wheels. In this case, the incremental mapping approach gives good results as can be seen in 7.27.

The scanSLAM algorithm without calibration gives outstanding results given the aforementioned limitation. From Figures 7.28, 7.29 and 7.30 it can be observed that the resulting map is good and can be used for the typical path planning operations. In particular, the figures show in blue the current scan as it should be placed if the algorithm trusts the prediction while the red one shows the correction made after the EKF update.

In order to obtain this results, the motion model parameters have been set to $\alpha_1 = \frac{35}{180}$, $\alpha_3 = 0.5$. In this way, all the possible source of errors are considered to depend on the motion model. The other parameters remain the same as for the Rawseeds dataset except for the calibration parameters $(0.792, 0.243, 0.0718)$, $([m], [m], [rad])$.

The wrong estimate of the robot motion model could be also a cause of the unsuccessful calibration procedure in the scanSLAM, see Figures 7.31 and 7.32. The same considerations made for Rawseeds dataset are valid also in this case. In contrast to the results proposed in the previous section, it can be observed that the covariances associated to the parameters (see Figures 7.33 7.34) do not converge to zero.

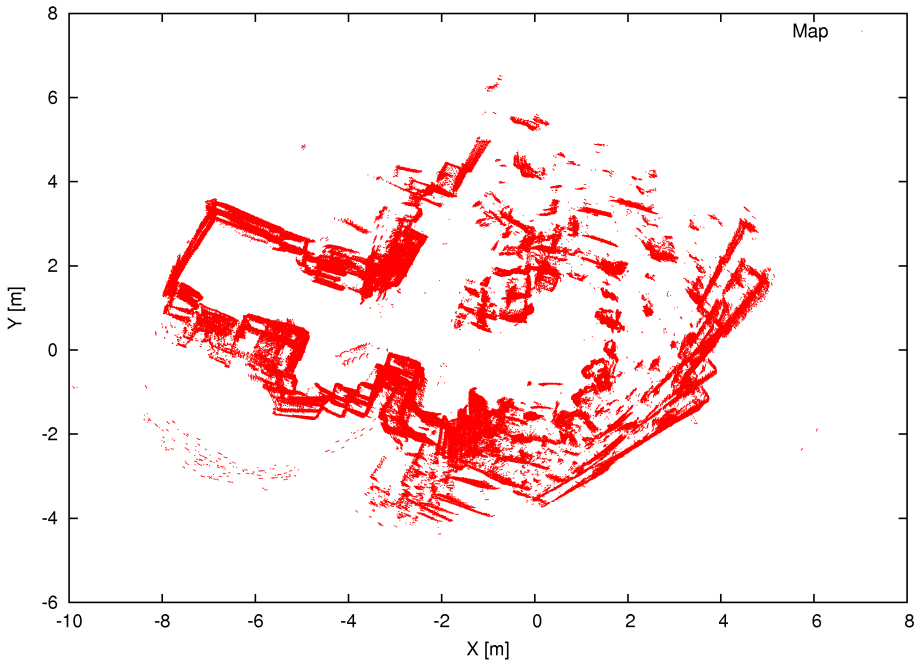


Figure 7.26: LURCH: Map resulting from odometric data. Iteration number 2290

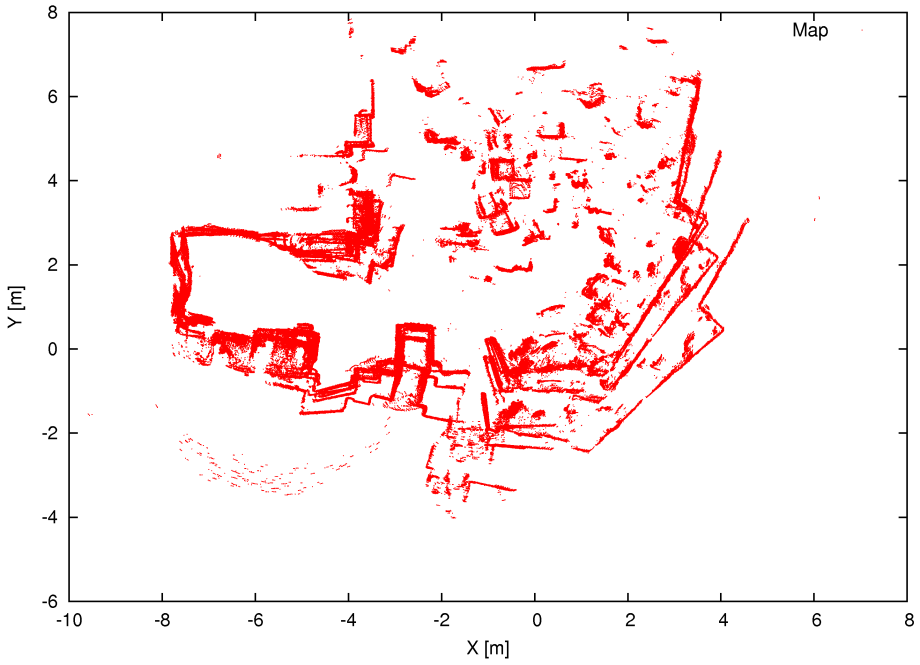
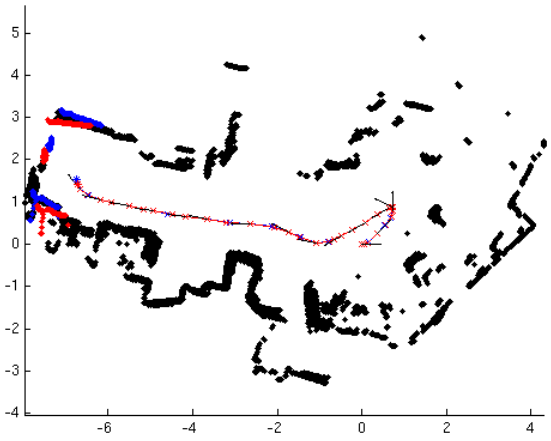
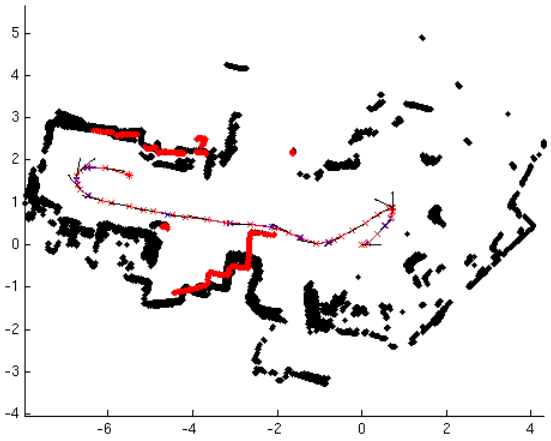


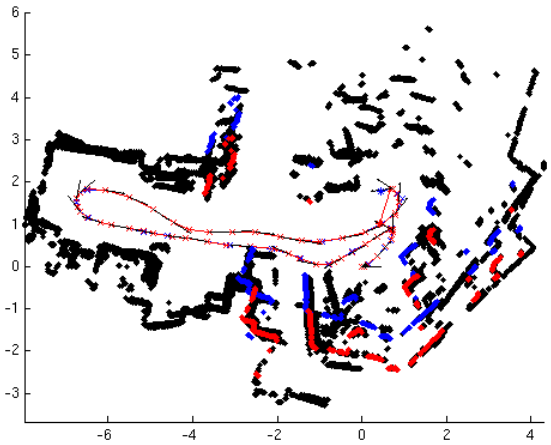
Figure 7.27: LURCH: Incremental Mapping. Iteration number 2290



(a) Iteration number 40

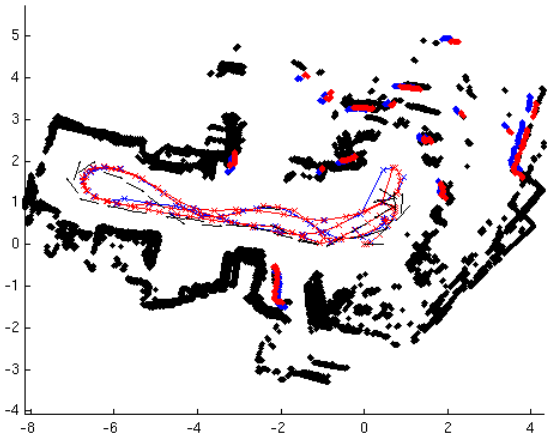


(b) Iteration number 45

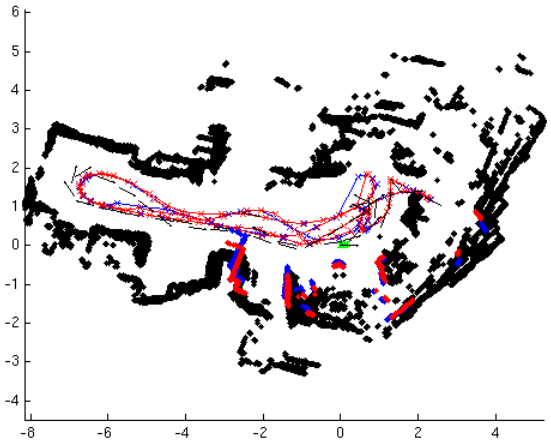


(c) Iteration number 60

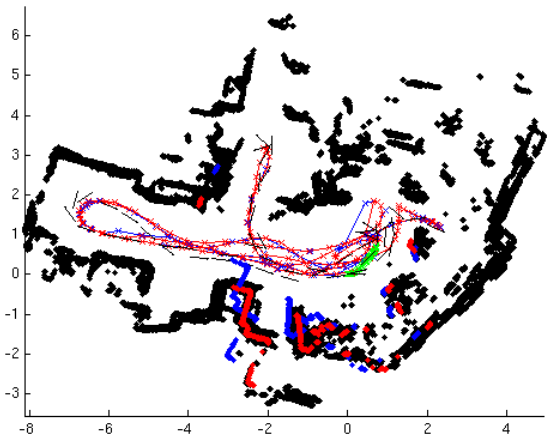
Figure 7.28: LURCH: without calibration



(a) Iteration number 93

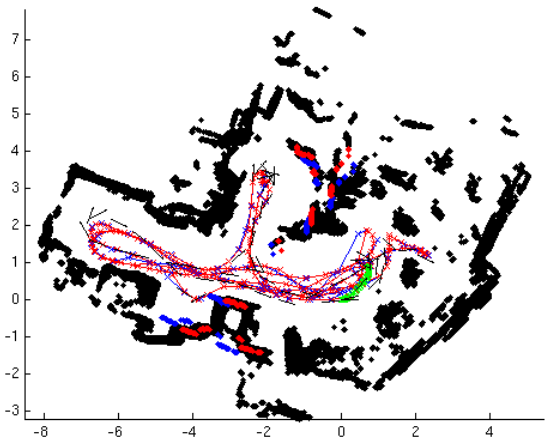


(b) Iteration number 106

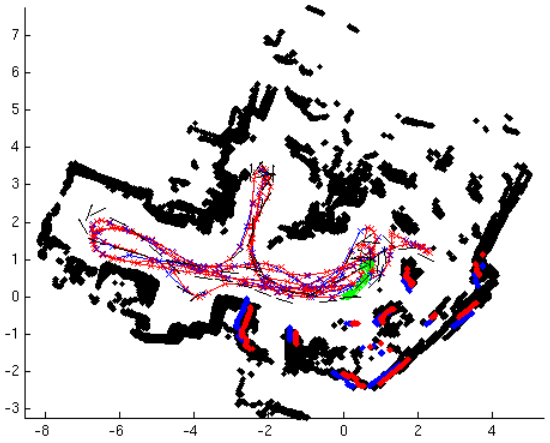


(c) Iteration number 140

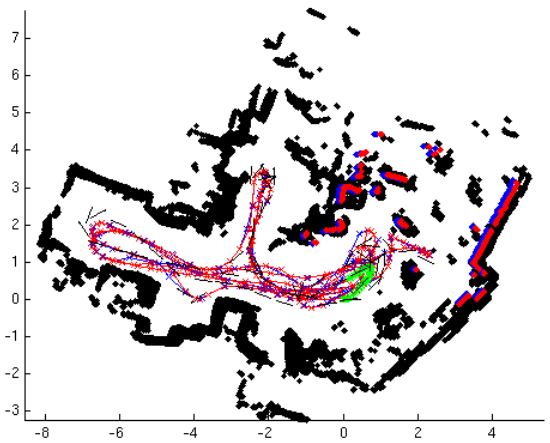
Figure 7.29: LURCH: without calibration



(a) Iteration number 156

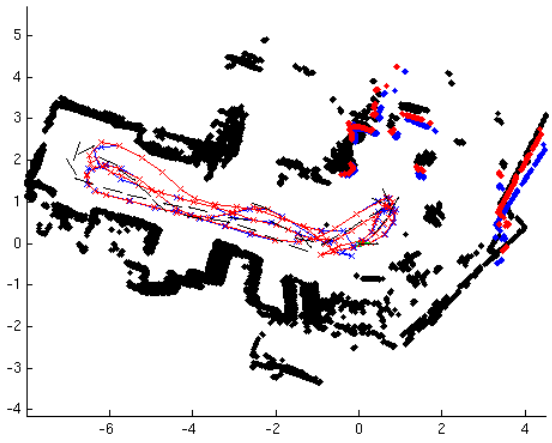


(b) Iteration number 160

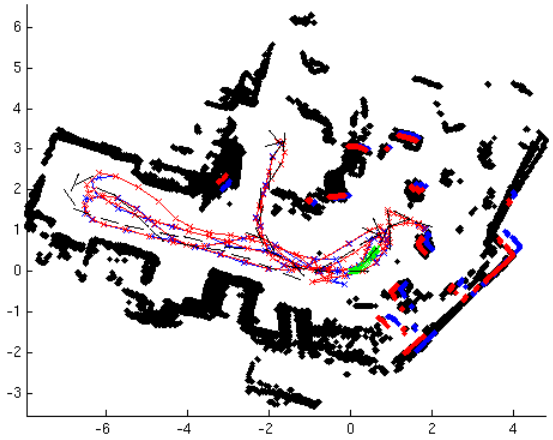


(c) Iteration number 190

Figure 7.30: LURCH: without calibration

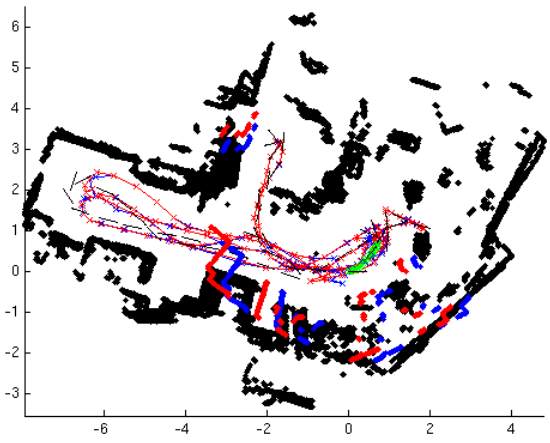


(a) Iteration number 93

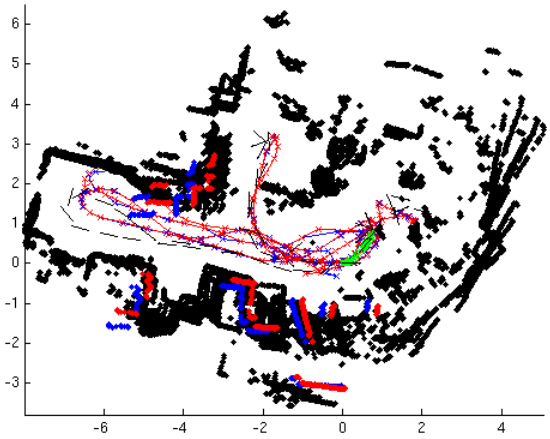


(b) Iteration number 140

Figure 7.31: LURCH: with calibration



(a) Iteration number 160



(b) Iteration number 167

Figure 7.32: LURCH: with calibration

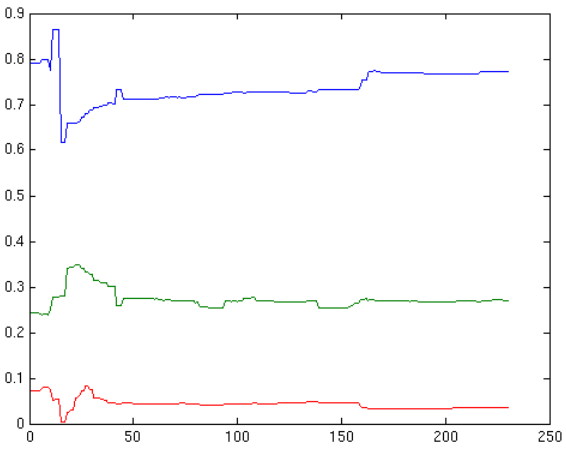
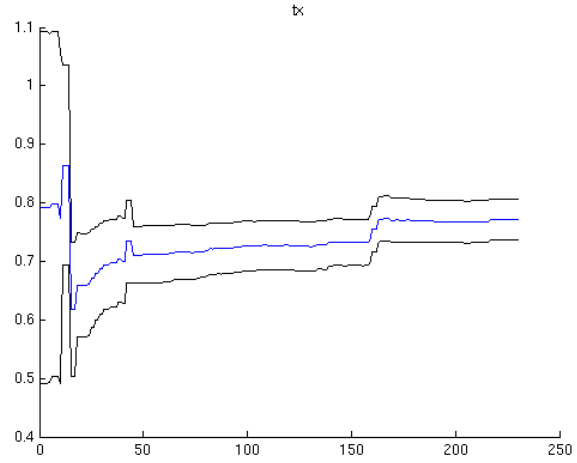
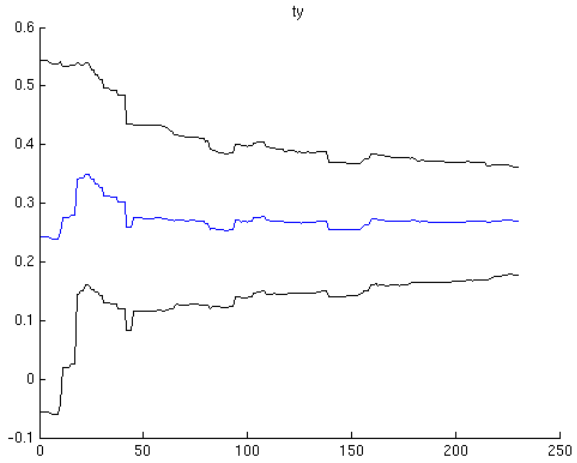


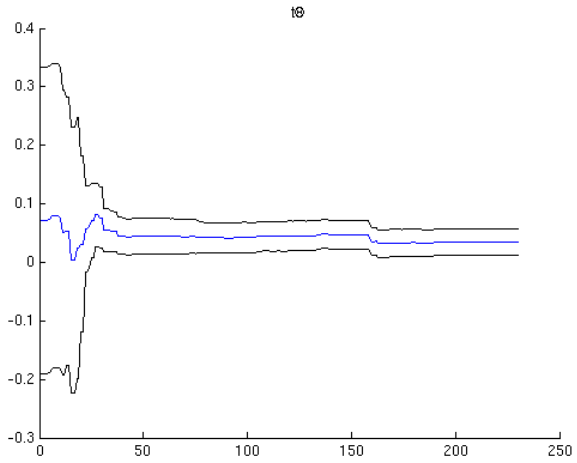
Figure 7.33: LURCH: calibration parameters



(a)



(b)



(c)

Figure 7.34: LURCH: calibration parameters

Chapter 8

Conclusions and Future Directions

"Discovery is seeing what everyone sees yet thinking what no one has thought"

Szent-Györgyi Albert

This dissertation deals with the particular field of mobile robotics named probabilistic robotics. Within probabilistic robotics, it is dealt with the problem of Simultaneous Localization and Mapping, SLAM, that aims to simultaneously estimate the pose of the robot and the map of the environment given noisy sensor data.

The present work is focused on the extension and implementation of a specific SLAM model named scanSLAM, which is conceptualized in two scientific papers [59, 25]. The model, which is based on Extended Kalman Filter, exploits raw laser sensor data and scan matching algorithms to obtain the measurements needed for the EKF SLAM. In order to give the correct importance to the measurement in the probabilistic filter, it is necessary to compute the covariance associated to the uncertainty in the minimization procedure. Therefore, in the present work, an extensive review of the methods to estimate the ICP (Iterative Closest Point scan matcher) covariance is conducted together with an analysis of the performances of each algorithms.

The scanSLAM algorithm for the robot centred approach, here defined, is tested both in simulated environment and with real datasets obtaining good results. These results show the pros and cons and the real capabilities of the algorithm. In particular, the algorithm is strongly dependent on the characterization of the covariance associated to motion model and to the measurement. In the case of the LURCH project, an overestimate of the

motion model parameters is needed to achieve the results shown in Chapter 7.

To the author's knowledge, this can be considered the first work in the literature which regards to extensively test the scanSLAM and which shows the results obtained with a real dataset using the recent advancements in the estimation of the ICP covariance (Censi [17]). In fact, Bosse [14], uses an optimized version of the algorithm and it is not clear how large the local maps are and if they are made entirely using this algorithm.

The software library is also developed with the aim to use this algorithm as the main SLAM software for LURCH project. For this reason, a model extension is developed to deal with the problem of the extrinsic calibration, (e.g., the rototranslation parameters between the laser and the robot reference frames) and multiple lasers.

Even if the procedure of self calibration fails with the standard method proposed, the author is confident that applying a different motion model, which takes into account the speed of the robot, and coupling the SLAM algorithm with the localization methods in use now, based on AR tags using ARToolkitPlus[1] (see also [22]), it will be possible to obtain good results also in this direction.

Many improvements to the scanSLAM algorithm can be the subject for further research. In particular, the correct implementation of the multiple lasers models and its extensions to deal with the case of high rotation is still needed. The multiple scan algorithm proposed in Chapter 5 was implemented in Maltlab but, after that, the final work had been more focused on the analysis of ICP covariance and the correct implementation of scanSLAM and the self calibration method. Different motion models should be tested in order to choose the best one. With more detailed motion model the calibration of the odometry (i.e., the baseline and the wheel radii) could be also feasible. Moreover, it is possible to improve the scan matchers capability by implementing the point-to-line metric for Metric Based ICP and evaluating and choosing heuristic algorithms to validate the scan matching results (i.e., testing if the scan matcher converges to the wrong minimum).

To improve the scanSLAM performance, an algorithm based on the Fisher matrix of the scan matcher can be implemented following the suggestions in Censi [18] to detect the under-constraint situations: corridors and circular environments. Finally a method to progressively diminish the number of landmarks by some sort of scan fusion procedure can be pursued too.

[1] http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php

Bibliography

[1] Approximate Nearest Neighbor Library ANN. `http://www.cs.umd.edu/~mount/ANN/`.

[2] G. Antonelli, S. Chiaverini, and G. Fusco. A calibration method for odometry of mobile robots based on the least-squares technique: Theory and experimental validation. *IEEE Transactions on Robotics*, 21(5):994–1004, 2005.

[3] T. Bailey. *Mobile robot localisation and mapping in extensive outdoor environments*. PhD thesis, Australian Center for Field Robotics, University of Sydney, 2002.

[4] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[5] O. Bengtsson and AJ Baerveldt. Localization in changing environments-estimation of a covariancematrix for the IDC algorithm. In *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, volume 4, 2001.

[6] O. Bengtsson and A.J. Baerveldt. Robot localization based on scan-matching estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems*, 44(1):29–40, 2003.

[7] P.J. Besl and H.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[8] P. Biber, S. Fleck, and W. Strasser. A probabilistic framework for robust and accurate matching of point clouds. *Lecture notes in computer science*, pages 480–487, 2004.

[9] P. Biber and W. Straßer. nScan-matching: simultaneous matching of multiple scans and application to slam. In *IEEE International Conference on Robotics and Automation*. Citeseer, 2006.

[10] J.L. Blanco. Derivation and implementation of a full 6D EKF-based solution to bearing-range SLAM. Technical report, Technical report, 2008.

[11] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1899–1906. Citeseer, 2003.

[12] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research*, 23(12):1113, 2004.

[13] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. In *2007 IEEE International Conference on Robotics and Automation*, pages 4820–4826, 2007.

[14] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667, 2008.

[15] A. Burguera, Y. González, and G. Oliver. On the use of likelihood fields to perform sonar scan matching localization. *Autonomous Robots*, 26(4):203–222, 2009.

[16] A. Censi. Scan matching in a probabilistic framework. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2291–2296, 2006.

[17] A. Censi. An accurate closed-form estimate of ICP's covariance. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[18] A. Censi. On achievable accuracy for range-finder localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4170–4175, 2007.

[19] A. Censi. An ICP variant using a point-to-line metric. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008.

[20] A. Censi. On achievable accuracy for pose tracking. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2009.

[21] A. Censi, L. Marchionni, and G. Oriolo. Simultaneous maximum-likelihood calibration of robot and sensor parameters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008.

[22] S. Ceriani. "sviluppo di una carrozzina autonoma d'ausilio ai disabili motori". Master's thesis, Politecnico di Milano, 2007.

[23] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, 2009.

[24] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *International Conference on Pattern Recognition*, volume 16, pages 545–548. Citeseer, 2002.

[25] A. Diosi and L. Kleeman. Laser scan matching in polar coordinates with application to SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3317–3322, 2005.

[26] A. Diosi and L. Kleeman. Fast laser scan matching using polar coordinates. *The International Journal of Robotics Research*, 26(10):1125, 2007.

[27] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[28] DW Eggert, A. Lorusso, and RB Fisher. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5):272–290, 1997.

[29] Eigen.
`http://eigen.tuxfamily.org`.

[30] A.I. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *Proceedings of the twenty-first international conference on Machine learning*. ACM New York, NY, USA, 2004.

[31] Fast Library for Approximate Nearest Neighbors FLANN. `http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN`.

[32] Generalized 2D Point Correspondence solution GPC. `http://www.cds.caltech.edu/~andrea/research/sw/gpc.html`.

[33] G. Grisetti, D.L. Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood map learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, 2008.

[34] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. *IEEE Transaction on Intelligent Transportation Systems*, 2008.

[35] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE international symposium on computational intelligence in robotics and automation (CIRA)*. Citeseer, 2000.

[36] Hähnel. *Mapping with Mobile Robots*. PhD thesis, Albert Ludwigs University, Freiburg, 2004.

[37] D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 496–501. Citeseer, 2002.

[38] M.B. Hurley. An information theoretic justification for covariance intersection and its generalization. In *Proc. of the 5th International Conference on Information Fusion*, 2002.

[39] R. Iser and F.M. Wahl. Building local metrical and global topological maps using efficient scan matching approaches. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, pages 1023–1030, 2008.

[40] S.J. Julier and J.K. Uhlmann. Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, 55(1):3–20, 2007.

[41] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[42] K. Konolige and K. Chou. Markov localization using correlation. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 1154–1159. Citeseer, 1999.

[43] K. Lenac, E. Mumolo, and M. Nolich. Fast genetic scan matching using corresponding point measurements in mobile robotics. *Lecture Notes in Computer Science*, 4448:375, 2007.

[44] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[45] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.

[46] C. Martin, E. Schaffernicht, A. Scheidig, and HM Gross. Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking. *Robotics and Autonomous Systems*, 54(9):721–728, 2006.

[47] A. Martinelli. Local Decomposition and Observability Properties for Automatic Calibration in Mobile Robotics. 2009.

[48] A. Martinelli. Using the Distribution Theory to Simultaneously Calibrate the Sensors of a Mobile Robot. 2009.

[49] A. Martinelli and R. Siegwart. Estimating the odometry error of a mobile robot during navigation. *In other words*, 1:3.

[50] A. Martinelli and R. Siegwart. Observability properties and optimal trajectories for on-line odometry self-calibration. In *Proceedings of the IEEE Conference on Decision and Control, San Diego, CA*, pages 3065–3070. Citeseer, 2006.

[51] A. Martinelli, N. Tomatis, A. Tapus, and R. Siegwart. Simultaneous localization and odometry calibration for mobile robot. *In other words*, 1:3.

[52] J. Minguez, F. Lamiraux, and L. Montesano. Metric-based scan matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, page 3557. Citeseer, 2005.

[53] J. Minguez, L. Montesano, and F. Lamiraux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047–1054, 2006.

[54] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *IEEE international conference on intelligent robots and systems (IROS)*. Citeseer, 2005.

[55] the Mobile Robot Programming Toolkit MRPT. `http://babel.isa.uma.es/mrpt/index.php/Main_Page`.

[56] M. Muja and D.G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *Preprint*, 2008.

[57] D. Nicholson, S. Julier, and J. Uhlmann. DDF: an evaluation of covariance intersection. In *Proc. 4th International Conference on Information Fusion*, volume 1, 2001.

[58] W. Niehsen. Information fusion based on fast covariance intersection filtering. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, 2002.

[59] J. Nieto, T. Bailey, and E. Nebot. Scan-slam: Combining ekf-slam and scan correlation. In *International Conference on Field and Service Robotics (FSR)*. Springer, 2005.

[60] J. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching SLAM. *Robotics and Autonomous Systems*, 55(1):39–49, 2007.

[61] E.B. Olson. Real-time correlative scan matching. *Ann Arbor*, 1001:48109.

[62] L.L. Ong, M. Ridley, B. Upcroft, S. Kumar, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte. A comparison of probabilistic representations for decentralised data fusion. *Proceedings of Intelligent Sensors, Sensor Networks and Information Processing*, 2005.

[63] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152. Citeseer, 2001.

[64] JZ Sasiadek and A. Monjazeb. A Comparison between EKF-SLAM and Fast-SLAM.

[65] JZ Sasiadek and A. Monjazeb. EKF as a Classical Solution to SLAM Problem.

[66] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. RSS, 2009.

[67] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, 1:167–193, 1990.

[68] TooN Algorithm Library TaG. `http://mi.eng.cam.ac.uk/~er258/cvd/tag.html`.

[69] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *IEEE international conference on robotics and automation (ICRA)*, volume 1, pages 321–328. Citeseer, 2000.

[70] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). 2005.

[71] F. Tungadi and L. Kleeman. Multiple laser polar scan matching with application to SLAM. In *Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia*. Citeseer, 2007.

[72] JK Uhlmann, S. Julier, and HF Durrant-Whyte. A culminating advance in the theory and practice of data fusion, filtering and decentralized estimation, 1997.

[73] J.K. Uhlmann, S.J. Julier, and M. Csorba. Nondivergent simultaneous map building and localization using covariance intersection. *Navigation and control technologies for unmanned systems II*, pages 2–11, 1997.

[74] B. Upcroft, L.L. Ong, S. Kumar, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte. Rich probabilistic representations for bearing only decentralised data fusion. In *Proceeding of The Eighth International Conference on Information Fusion*, pages 1999–2000.

[75] G. Welch and G. Bishop. An introduction to the Kalman filter. *University of North Carolina at Chapel Hill, Chapel Hill, NC*, 2006.

[76] D. Yang. *C++ and object-oriented numeric computing for scientists and engineers*. Springer Verlag, 2001.

[77] H. Yoshitaka, K. Hirohiko, O. Akihisa, and Y. Shin'ichi. Mobile Robot Localization and Mapping by Scan Matching using Laser Reflection Intensity of the SOKUIKI Sensor. In *IECON*, volume 6, pages 302–318, 2006.

[78] C. Ze-Su, H. Bing-Rong, and L. Hong. An Improved Polar Scan Matching Using Genetic Algorithm. *Information Technology Journal*, 6(1):89–95, 2007.

[79] Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Citeseer.

[80] Q. Zhang and R. Pless. Constraints for heterogeneous sensor auto-calibration. In *IEEE Workshop on Realtime 3D Sensors and Their Use*, pages 38–43. Citeseer, 2004.

[81] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):152, 1994.

Appendix A

Notation: 2D Rototraslations

This appendix is dedicated to the concise enlisting of the mathematical notations that are used in this final dissertation.

A 2D rototranslation is a geometric transformation that is composed by a rotation followed by a translation. The transformation can be given in matrix or vector form. The graphic notation used in this dissertation is shown in Figure A.1. In the dissertation, the following notation is employed:

RT_B^A is the transformation between the two reference frames A and B, which applied to a point p expressed in reference frame B returns p in the reference frame A.

In matrix form can be represented in homogeneous coordinates as:

$$RT_B^A = \begin{pmatrix} cos(\theta) & -sin(\theta) & x \\ sin(\theta) & cos(\theta) & y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \tag{A.1}$$

Two rototranslation matrix can be composed using the standard matrix product:

$$RT_B^A = RT_C^A RT_B^C \tag{A.2}$$

The application of the trasformation to a point p is also given by the matrix product keeping in mind that p must be expressed in homogeneous coordinate:

$$p = [x, y, 1]^T,\ p_A = RT_B^A p_B \tag{A.3}$$

Since the rotation is given by an orthonormal matrix the inverse of the rototranslation, which is also the matrix inverse, inherits a simple structure:

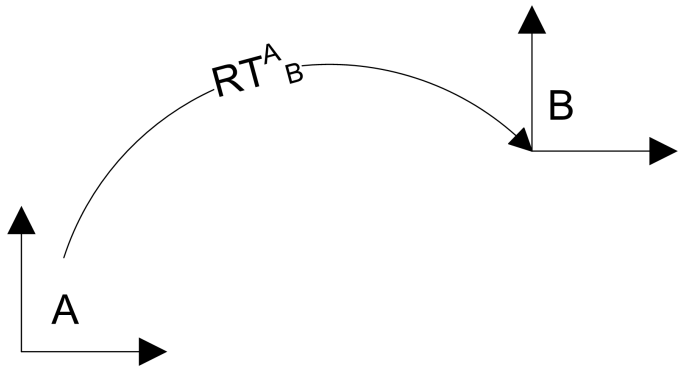


Figure A.1: RT_B^A is the rototranslation that maps points in the reference frame B to the same points in the reference frame A

$$RT_A^B = (RT_B^A)^{-1} = \begin{pmatrix} cos(\theta) & sin(\theta) & -x * cos(\theta) - y * sin(\theta) \\ -sin(\theta) & cos(\theta) & x * sin(\theta) - y * cos(\theta) \\ 0 & 0 & 1 \end{pmatrix} \quad \text{(A.4)}$$

$$RT_A^B = \begin{pmatrix} R^T & -R^T * t \\ 0 & 1 \end{pmatrix} \quad \text{(A.5)}$$

The rototranslation can be represented in vector form as:

$$RT_B^A = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad \text{(A.6)}$$

The composition between the rototranslation can be defined using the symbol $\oplus$ as binary operator.

$$RT_B^A = RT_C^A \oplus RT_B^C \quad \text{(A.7)}$$

$$RT_C^A \oplus RT_B^C = t1 \oplus t2 = \begin{pmatrix} x_{t2} * cos(\theta_{t1}) - y_{t2} * sin(\theta_{t1}) + x_{t1} \\ x_{t2} * sin(\theta_{t1}) + y_{t2} * cos(\theta_{t1}) + y_{t1} \\ \theta_{t1} + \theta_{t2} \end{pmatrix} \quad \text{(A.8)}$$

The symbol used for inverse transformation is $\ominus$., used as a unitary operator. The result is given by:

$$RT_A^B = \ominus RT_B^A = \begin{pmatrix} -x * cos(\theta) - y * sin(\theta) \\ x * sin(\theta) - y * cos(\theta) \\ -\theta \end{pmatrix} \quad \text{(A.9)}$$

Appendix B

Hessian Matrix of the Error Function

The Hessian matrix of the error function can be numerically estimated. The value H is used in numerous algorithms that compute the covariance of the scan matching minimization error (see Algorithm 3.3.1, 3.3.2).

The general form for the Hessian matrix for the error $E(z = RT)$ is:

$$H(E(z)) = \frac{d^2E}{dz} = \begin{pmatrix} \frac{\partial^2 E}{\partial x^2} & \frac{\partial^2 E}{\partial x \partial y} & \frac{\partial^2 E}{\partial x \partial \theta} \\ \frac{\partial^2 E}{\partial x \partial y} & \frac{\partial^2 E}{\partial y^2} & \frac{\partial^2 E}{\partial y \partial \theta} \\ \frac{\partial^2 E}{\partial x \partial \theta} & \frac{\partial^2 E}{\partial y \partial \theta} & \frac{\partial^2 E}{\partial \theta^2} \end{pmatrix} \tag{B.1}$$

Where

$$\frac{\partial^2 E}{\partial x^2} \approx \frac{E(z + 2\Delta x) - 2E(z) + E(z - 2\Delta x)}{4\Delta x^2} \tag{B.2}$$

Whereas, the mixed derivatives can be computed as

$$\frac{\partial^2 E}{\partial x \partial y} \approx \frac{\partial}{\partial y}\left(\frac{E(z + \Delta x) - E(z - \Delta x)}{2\Delta x}\right) \tag{B.3}$$

which results to be

$$\frac{E(z + \Delta x + \Delta y) - E(z - \Delta x + \Delta y)}{4\Delta x \Delta y} - \frac{E(z + \Delta x - \Delta y) - E(z - \Delta x - \Delta y)}{4\Delta x \Delta y} \tag{B.4}$$

To improve the estimation of the Hessian matrix the searching for corresponding pairs can be computed after each variation of the variable z. This procedure prevents false corresponding pairs which they would not be matched using the method for the search of the nearest neighbours (see also [5, 6]).

Appendix C

Kalman Filter Example of Implementation: Estimation of a Constant Vector

This appendix is dedicated to the implementation of a simple Kalman Filter in order to test the correctness of the software implementation. The aim is to estimate the value of a 2D vector subjected to noisy measurements.

Using the Extended Kalman Filter framework (See Algorithms 4.2.1 and 5.1.1), the model is expressed as follows.

The state is represented by x_t:

$$x_t = \begin{pmatrix} x \\ y \end{pmatrix} \tag{C.1}$$

where the real vector for this example is $x_{real} = (1, 2)$. The initial state vector is initialized by adding a white noise to the real state values:

$$x_0 = x_{real} + \delta \tag{C.2}$$

where δ is a WN$\sim (0, P_{noise})$. The initial covariance matrix $P_{0|0}$ is equal to P_{noise}.

The control vector u_t is set to zero while the covariance of the additional motion noise in the state space is set to a non-zero matrix Q.

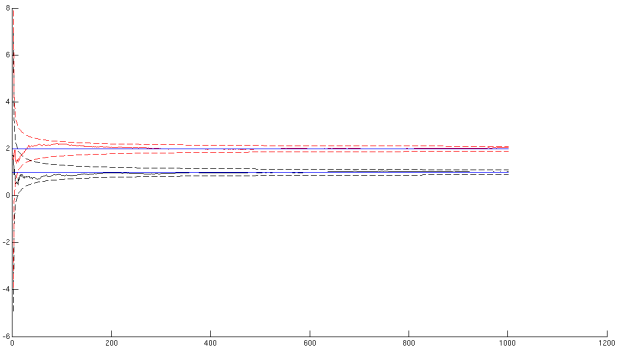
The measurement is expressed as a 2D vector representing directly (x, y) components to which a white noise is added.

$$z_t = x_{real} + \epsilon \tag{C.3}$$

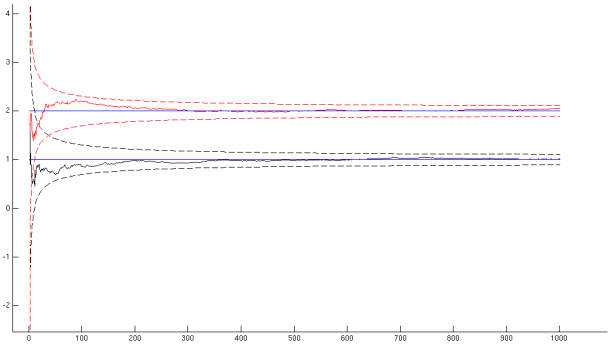
where ϵ is a WN$\sim (0, R_t)$ with $R_t = R$ constant. The results for the simulation are shown in Figure C.1, C.2, C.3, C.4. The different parameters for each dataset are summarized in the table.

Dataset	P_{noise}	Q	R	x_0	x_{end}
1	$\begin{pmatrix} 2^2 & 2 \\ 3 & 2^2 \end{pmatrix}$	$\begin{pmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.7270 \\ 1.5583 \end{pmatrix}$	$\begin{pmatrix} 1.0153 \\ 2.0485 \end{pmatrix}$
2	$\begin{pmatrix} 2^2 & 2 \\ 3 & 2^2 \end{pmatrix}$	$\begin{pmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{pmatrix}$	$\begin{pmatrix} 10^{-2} & 0 \\ 0 & 10^{-2} \end{pmatrix}$	$\begin{pmatrix} 1.7270 \\ 1.5583 \end{pmatrix}$	$\begin{pmatrix} 1.0014 \\ 2.0170 \end{pmatrix}$
3	$\begin{pmatrix} 10^{-2} & 0 \\ 0 & 0.5^2 \end{pmatrix}$	$\begin{pmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.0364 \\ 1.7675 \end{pmatrix}$	$\begin{pmatrix} 1.0169 \\ 2.0481 \end{pmatrix}$
4	$\begin{pmatrix} 10^{-2} & 0 \\ 0 & 0.5^2 \end{pmatrix}$	$\begin{pmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{pmatrix}$	$\begin{pmatrix} 0.2^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}$	$\begin{pmatrix} 1.0364 \\ 1.7675 \end{pmatrix}$	$\begin{pmatrix} 1.0028 \\ 2.0170 \end{pmatrix}$

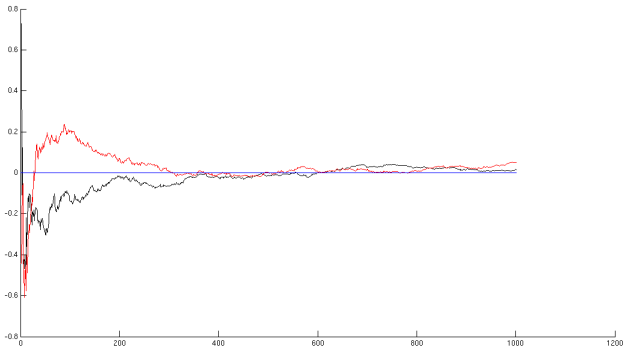
From the results, it can be concluded that even if the single measurement is very noisy the integration of successive measurements permits to find a good estimate of the real vector. It should be noted that in this case the Kalman Filter is reduced to an on-line average estimator. The plots confirm the correctness of the software implementation.



(a)

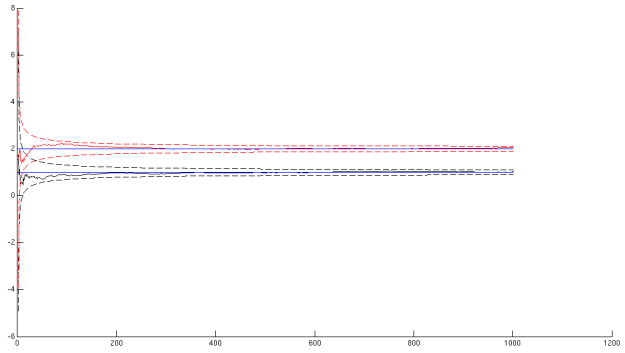


(b)

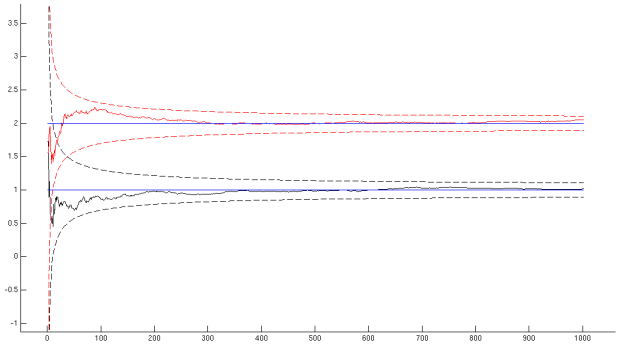


(c)

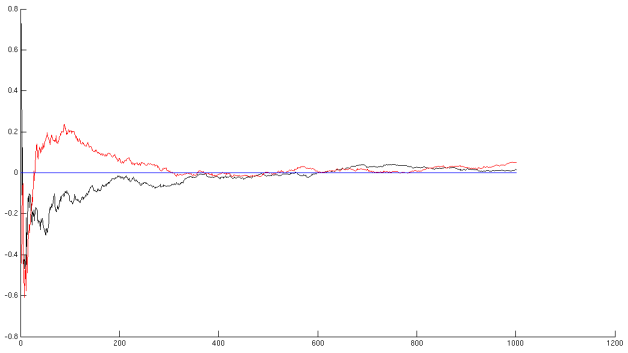
Figure C.1: Dataset 1. In this dataset, the uncertainty both in prediction and in measurement is high



(a)

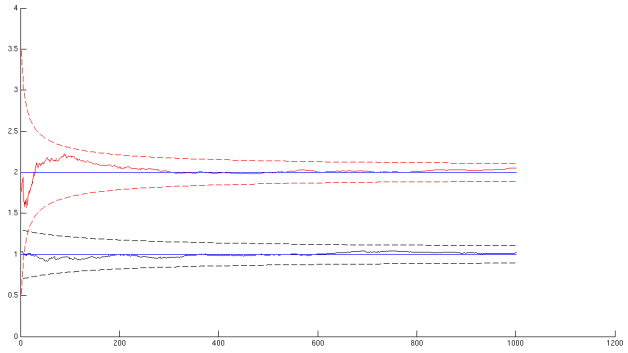


(b)

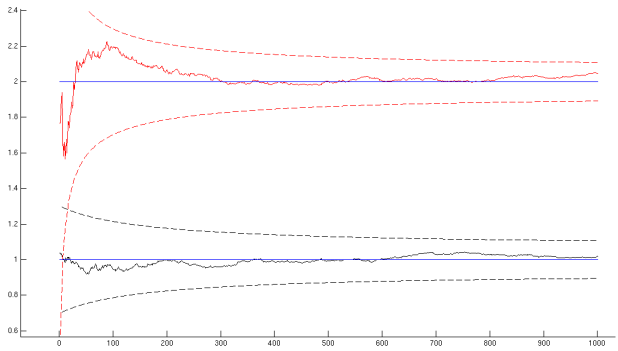


(c)

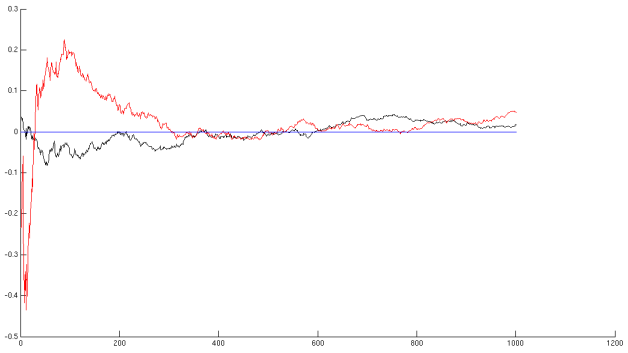
Figure C.2: Dataset 2. In this dataset, the uncertainty in prediction is high, while uncertainty in in measurement is low



(a)

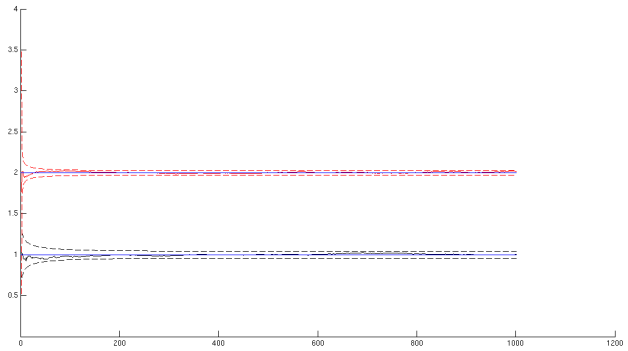


(b)

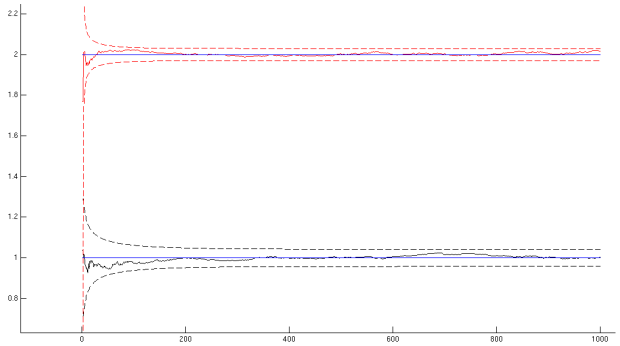


(c)

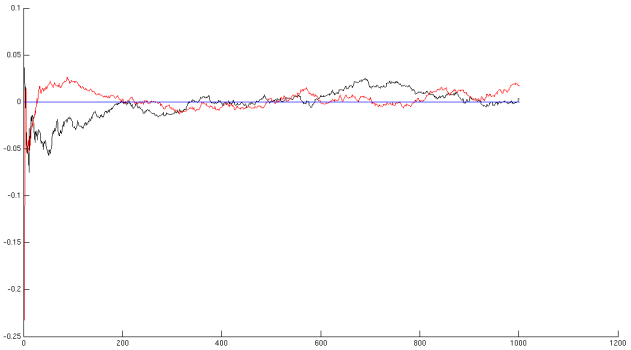
Figure C.3: Dataset 3. In this dataset, the uncertainty in prediction is low, while uncertainty in in measurement is high



(a)



(b)



(c)

Figure C.4: Dataset 4. In this dataset, the uncertainty both in prediction and in measurement is low

Appendix D

Mahalanobis Distance

Mahalanobis distance is a particular distance metric introduced by P. C. Mahalanobis in 1936. It is typically used to determine the similarity of unknown sample set with respect to a known one.

The mathematical definition can be expressed as follows.

Given a multivariate vector

$$x = \begin{pmatrix} x_1 & x_2 & \dots & x_i & \dots & x_N \end{pmatrix}^T \tag{D.1}$$

and a Gaussian with mean

$$\mu = \begin{pmatrix} \mu_1 & \mu_2 & \dots & \mu_i & \dots & \mu_N \end{pmatrix}^T \tag{D.2}$$

and covariance C, the Mahalanobis distance is given by:

$$dist(x) = \sqrt{(x-\mu)^T C^{-1} (x-\mu)} \tag{D.3}$$

This distance generally differs from the Euclidean one because it takes into account the uncertainty in the estimate of μ.

An example is given in Table D.1 for the points in Figure D.1 for the point:

$$\mu = \begin{pmatrix} 0 & 0 \end{pmatrix}^T \tag{D.4}$$

and covariance matrix C equals to:

$$C = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix}^T \tag{D.5}$$

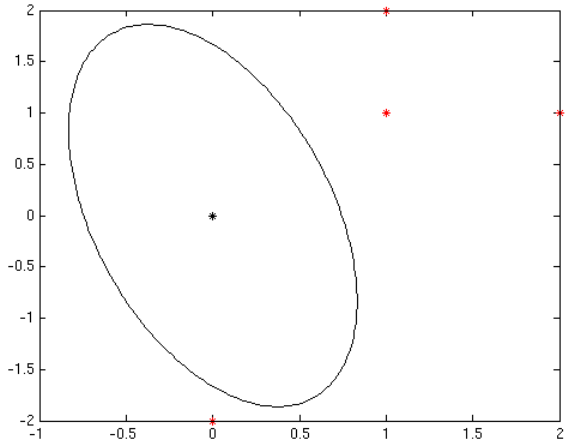


Figure D.1: Mahalanobis Distance. In black it is represented the point p with its covariance C whereas, in red points at different distances are represented

$Point$	$Euclidean^2$	$Mahalanobis^2$
$\begin{pmatrix}1\\1\end{pmatrix}$	2	1
$\begin{pmatrix}2\\1\end{pmatrix}$	5	3.5
$\begin{pmatrix}1\\2\end{pmatrix}$	5	2
$\begin{pmatrix}0\\-2\end{pmatrix}$	4	2

Table D.1: In the table are shown the point coordinates, the Euclidean distance and the Mahalanobis distance with respect to the point $(0, 0)$