ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria ed Architettura
Corso di Laurea Magistrale in Ingegneria Elettronica e
Telecomunicazioni per l'Energia

SIMULTANEOUS LOCALIZATION AND MAPPING USING RADAR IMAGES

Tesi di Laurea Magistrale in
Sistemi di telecomunicazioni LM

Presentata da:
LUCA TIRANTI

Relatore:
Chiar.mo Prof. Ing.
DAVIDE DARDARI

Correlatori:
Dott. Ing.
GIANNI PASOLINI
Dott. Ing.
FRANCESCO GUIDI
Dott. Ing.
ANNA GUERRA

SESSIONE III
ANNO ACCADEMICO 2019-20

Acronyms

SLAM Simulataneous Localization and Mapping

DBN Dynamic Bayesian Network

KF Kalman Filter

EKF Extended Kalman Filter

UKF Unscented Kalman Filter

PF Particle Filter

FMCW Frequency Modulated Continuos Wave

RADAR RAdio Detection And Ranging

RCS Radar Cross Section

FOV Field of View

MAP Maximum A Posteriori

MMSE Maximum Mean Square Error

ICP Iterative Closest Point

PLICP Point to Line Iterative Closest Point

NDT Normal Distribution Transform

CSM Correlative Scan Matching

DFT Discrete Fourier Transform

FFT Fast Fourier Transform

IFFT Inverse Fast Fourier Transform

IMU Inertial Measurement Unit

VNA Vector Network Analyzer

Introduzione

Il concetto di localizzazione e mappatura simultanea (SLAM) nasce in ambito robotico dove, tradizionalmente, i robot sono dotati di un sensore laser e sono in grado di navigare e mappare l'ambiente circostante sconosciuto. Le tecnologie SLAM ad alta accuratezza, basate su laser e steering meccanico, forniscono stime della distanza ad alta definizione e fasci diretti molto stretti. Sfortunatamente, queste tecnologie sono costose e ingombranti non potendo quindi essere integrate in dispositivi mobili.
Questo lavoro affronta il problema dello SLAM utilizzando la tecnologia radar a onde millimetriche (mmWave) testando le prestazioni di localizzazione e mappatura in ambienti interni reali conducendo campagne di misura. Lo scenario considerato e le tecnologie impiegate sono in linea con l'idea del *personal mobile radar* [1],[2],[3],[4] e questo lavoro può essere considerato come un proof-of-concept per lo sviluppo di questa idea.
Un esempio di radar personale in uno scenario interno è mostrato in Fig. 1.
Secondo la visione di radar personale, un array di antenne a onde millimetriche può essere integrato all'interno di dispositivi mobili come smartphone o tablet, che possono scansionare automaticamente l'ambiente e mapparlo attraverso un fascio diretto molto stretto e puntamento a guida elettronica. Queste caratteristiche lo rendono una possibile soluzione per lo SLAM radio (R-SLAM) ad alta precisione che rappresenta un'alternativa alla tecnologia laser. Per questi motivi, il radar personale evita lo l'impiego di infrastrutture ad-hoc per il posizionamento e la mappatura indoor. Inoltre, un vantaggio dell'utilizzo delle onde millimetriche in confronto ai sistemi lidar o vision-based, è il loro costo inferiore e la loro capacità di operare in condizioni di scarsa visibilità, come con fumo o bassa illuminazione. Il radar personale potrebbe, quindi, trovare applicazione in contesti come la navigazione autonoma nel settore industriale o per guidare le persone attraverso ambienti sconosciuti e pericolosi ma, anche, come guida per i non vedenti nella loro vita quotidiana.
Nonostante le considerazioni di cui sopra, l'applicazione dei sensori a onde millimetriche al contesto SLAM è difficoltosa a causa della loro scarsa risoluzione

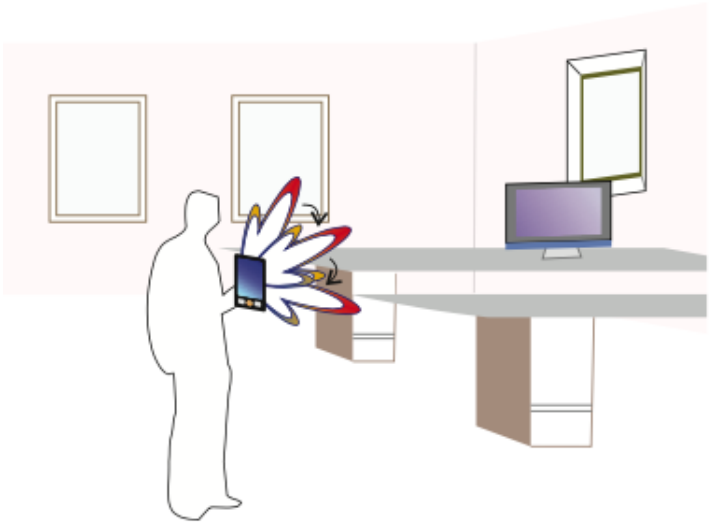


Figure 1: Esempio di uno scenario SLAM dove un dispositivo mobile, equipaggiato con un array di antenne, scansiona l'ambiente [4]

angolare e delle caratteristiche di riflessione dei materiali a queste frequenze. Per mostrare la fattibilità delle tecnologie mmWaves per SLAM, questo lavoro è stato organizzato come segue.
Nel Capitolo 1, viene presentato il problema dello SLAM e alcune strategie classiche per affrontarlo come il filtraggio Bayesiano ma anche approcci basati sulla teoria dei grafi.
Nel Capitolo 2, viene posta l'attenzione sugli algoritmi di registrazione di nuvole di punti; essi sono sfruttati allo scopo di stimare la posa relativa tra due misurazioni di sensori consecutive, consentendo la stima dell'intera traiettoria seguita dal radar mobile.
Nel Capitolo 3 sono riportati i fondamenti della tecnologia radar. In particolare, vengono descritti i principi della tecnologia radar mmWaves su cui si basa il radar mmWaves IWR1443 di Texas Instrument. Questo dispositivo è stato utilizzato per raccogliere dati dalle campagne di misura e, successivamente, per valutare le prestazioni degli algoritmi SLAM proposti. In questo capitolo, viene descritta anche una tecnologia di array di antenne a 300 GHz adottata presso l'istituto di ricerca CEA-Leti (Grenoble, Francia) per misurazioni in locali indoor. Questo sistema di misura è stato allestito in occasione della collaborazione tra CEA-Leti e l'Università di Bologna nell'ambito del progetto europeo PRIMELOC (Personal Radars for Radio Imaging and Infrastructure-less Localization). Il set di dati raccolti presso CEA-Leti viene

anch'esso sfruttato per valutare le prestazioni degli algoritmi SLAM proposti in questo lavoro.
Nel Capitolo 4 vengono descritti gli algoritmi SLAM proposti. Prima di tutto, viene presentato un algoritmo SLAM dove la stima della traiettoria è eseguita in modo incrementale e indipendente dalla fase di mappatura. Ciò significa che la mappa viene aggiornata a partire dalla traiettoria finale stimata ma quest'ultima non è influenzata dalla prima. In questo approccio, le pose relative vengono calcolate tra due scansioni radar consecutive. Successivamente, viene presentato un algoritmo basato sulla strategia di registrazione delle immagini con trasformata di Fourier-Mellin. Infine, per superare il problema dell'accumulo di errori degli approcci incrementali, è stata progettata una strategia scan-to-map per le applicazioni SLAM. L'idea chiave è sfruttare la mappa per eseguire il matching fra due scan trovando una traiettoria più precisa. Qui, la mappatura e la stima della traiettoria vengono eseguite congiuntamente e sono influenzate l'una dall'altra.
A questo punto, nel Capitolo 5 viene riportata una descrizione delle campagne di misura e dei risultati ottenuti. Nello specifico, nella prima sezione, vengono descritte le campagne di misura effettuate con il radar TI mmWave a 77 GHz in tre diversi scenari indoor e vengono mostrati di seguito i risultati ottenuti con le diverse strategie SLAM proposte. Nella seconda sezione di questo capitolo vengono descritte due campagne di misura realizzate presso CEA-Leti, con tecnologia a 300 GHz. Quindi, vengono visualizzati i risultati di SLAM ottenuti con i diversi set di dati.
Infine, nel Capitolo 6, vengono tratte alcune conclusioni e vengono evidenziate le direzioni future della ricerca.

Introduction

The concept of simultaneous localization and mapping (SLAM) is born in the robotic field where, traditionally, robots are equipped with a laser sensor and they are able to navigate and map an unknown surrounding environment. High accuracy SLAM technologies, based on laser and mechanical steering device, provide high-definition distance estimates and very narrow steering beams. Unfortunately, these technologies are expensive and bulky and they cannot be integrate into mobile devices.
This work deals with a SLAM problem using a millimeter wave (mmWave) radar technology where the localization and mapping performance has been tested in real indoor environments by conducting measurements campaigns. The considered scenario and the employed technologies are in line with the idea of the *personal mobile radar* [1],[2],[3],[4] and this work can also be considered as a proof-of-concept for the development of this idea.
An example of personal radar in an indoor scenario is shown in Fig. 2.
According to the personal radar vision, a mmWave antenna array can be integrated inside mobile devices like smartphones or tablets, which can automatically scan the environment and mapping it with a narrow beam and electronic-driving steering. These characteristics make it as a possible high accuracy radio SLAM (R-SLAM) solution that represents an alternative to laser technology. For these reasons, the personal mobile radar avoids the exploitation of ad-hoc infrastructures for indoor positioning and mapping. Moreover, an advantage of using mmWave frequencies instead of lidar or vision-based systems is their lower cost and their capability to operate in scarce visibility conditions, as with smoke or poor lighting. For these reasons, personal mobile radars could find application in contexts such as autonomous navigation of vehicles in the industry field or to drive people through unknown and dangerous environments but, also, such as guidance for visually impaired people in their daily life.
Despite the above considerations, the application of mmWave sensors to the SLAM context is challenging because of their poor angle resolution and the reflection characteristics of materials at these frequencies.

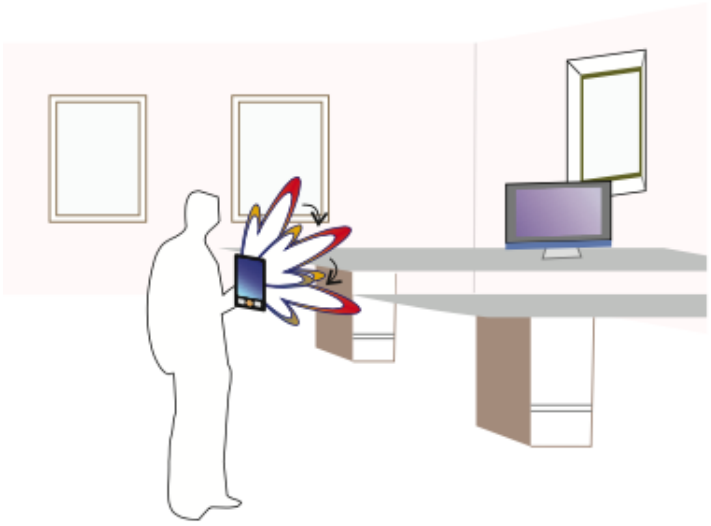


Figure 2: Example of a SLAM scenario where a mobile device, equipped with an antenna array, scans an environment [4]

To show the feasibility of mmWaves technologies for SLAM, we have organized this work as follows.
In Chapter 1, we present the SLAM problem and some classical strategies to tackle it with stochastic filtering approaches, like Bayesian filtering, and graph-based theory.
In Chapter 2, we put the attention on points clouds registration algorithms; they are exploited with the purpose to estimate the relative pose between two consecutive sensor measurements, allowing the estimation of the full trajectory followed by the mobile radar.
In Chapter 3, fundamentals of radar technology are reported. In particular, we describe the principles of mmWaves radar technology on which the mmWaves radar IWR1443 of Texas Instrument is based. This device has been used to collect data in measurement campaigns and, then, to assess the performance of the proposed SLAM algorithms. In this chapter, we also describe a 300 GHz antenna array technology adopted at CEA-Leti research institute (Grenoble, France) for indoor measurements. This measurement system was set up within a collaboration between CEA-Leti and the University of Bologna in the context of the European project PRIMELOC (Personal Radars for Radio Imaging and Infrastructure-less Localization). The data set collected at CEA-Leti is exploited in this work to assess the performance of the proposed SLAM algorithms.

In Chapter 4, the proposed SLAM algorithms are described. First of all, we present a SLAM algorithm where the estimation of the trajectory is performed incrementally and independently of the mapping phase. This means that the map is updated starting from the final estimated trajectory but the latter is not influenced by the former. In this approach, the relative poses are computed between two consecutive radar scans. Successively, an algorithm based on image registration strategy with the Fourier-Mellin transform is presented. Finally, to overcome the error accumulation issue of incremental approaches, a scan-to-map strategy for SLAM applications is designed. The key idea is to exploit the map to perform the scan matching and to find a more accurate trajectory. Here, the mapping and trajectory estimation are performed jointly and they are influenced by each other.
At this point, in Chapter 5 a description of the measurement campaigns and the results obtained are reported. Specifically, in the first section, the measurement campaigns made with the TI mmWave radar at 77 GHz in three different indoor scenarios are described. The results obtained with the different proposed SLAM strategies are shown. In the second section of this chapter, two measurement campaigns made at CEA-Leti, with a 300 GHz technology are described. Then, the SLAM results obtained with different datasets are shown.
Finally, in Chapter 6, some conclusions are drawn and future directions of research are highlighted.

Contents

Chapter 1

SLAM Theory

The goal of the simultaneous localization and mapping (SLAM) problem is to estimate the mobile node (often referred to as agent or mobile user) trajectory and, simultaneously, the map of the environment where the node is located without the need of any a priori knowledge.
In the literature, a large variety of solutions to SLAM problem is available but there is not an approach to fix the problem better than other. The best solution considers the agent type and sensors on it, the surrounding enviroment and the system performance requirements [5].
In the next sections, different ways to address the SLAM problem are presented.

1.1 Probabilistic SLAM

As mentioned before, the goal of the SLAM problem is to infer the state of the environment, where the state is composed of the radar state (i.e., its position, orientation and velocity) and of a map of the environment. The state is usually modeled as a random vector because of uncertainties on its dynamics and on the measurements used for the estimation process. For these reasons, SLAM problem is described by means of probabilistic tools.
To solve such a problem, different strategies can be adopted [6]: *filtering approaches* which model the problem as an online state estimation where the state consists in the current agent position and the map or *smoothing approaches* which estimate the full trajectory of the agent from the full set of measurements. The latter address to full SLAM problem and typically rely on least square error minimization techniques
To describe these approaches, the following quantities must be defined :

- $\boldsymbol{x}_k$: the state vector describing the location and orientation of the agent

at instant k

- $\boldsymbol{u}_k$: the control vector, applied at time $k-1$ to drive the mobile node to a state $\mathbf{x}_k$ at time k
- $\boldsymbol{z}_k$: the vector of the observations collected by the mobile node at the instant k
- $\boldsymbol{m}_k$: the vector representing the map of the environment at time k.

It is also necessary to define :

- $\boldsymbol{x}_{1:K} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_K\}$: the agent's trajectory
- $\boldsymbol{u}_{1:K} = \{\boldsymbol{u}_1, ..., \boldsymbol{u}_K\}$: the history of control inputs
- $\boldsymbol{z}_{1:K} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_K\}$: the whole set of measurements

Solving the full SLAM problem consists of estimating the posterior probability of the agent's trajectory $\mathbf{x}_{1:K}$ and the map $\mathbf{m}$, that is considered stationary, i.e., $\mathbf{m}_k = \mathbf{m}\ \forall\ k$, of the environment given all the measurments plus an arbitrarily initial position $\mathbf{x}_0$:

$$p(\mathbf{x}_{1:K}, \mathbf{m} | \mathbf{z}_{1:K}, \mathbf{u}_{1:K}, \mathbf{x}_0) \tag{1.1}$$

This probability distribution describes the joint posterior density of the full map and agent state, given the whole recorded observations and whole control inputs together with the initial state of the agent.
In many practical settings, we are not interested that at a given time instant k, we get the full joint posterior distribution of the sequence of states $\mathbf{x}_{0:K}$ instead, the posterior distribution of current state $\mathbf{x}_k$ given all the past measurment $\mathbf{z}_{1:k}$ is sufficient [7]. To compute this distribution, we rely on the Bayesian filtering theory.

1.2 Filtering Approaches

1.2.1 Bayesian Filter

To introduce Bayesian filtering, a state-space model can be defined by the use of the following two statistical models:

- ***Observation model***: describes the probability of making an observation $\mathbf{z}_k$ given that the agent is at location $\mathbf{x}_k$ in the map

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \tag{1.2}$$

- ***Transition model***: represents the probability that the agent at time k is in $\mathbf{x}_k$ given that, at time $k-1$, it was in $\mathbf{x}_k$ and its control input was $\mathbf{u}_k$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \tag{1.3}$$

Often in theory and practice, one adopts the following general description of the state-space model:

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{v}_k \tag{1.4}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{m}) + \mathbf{n}_k \tag{1.5}$$

where *g()* and *h()* are in generally non-linear functions and $\mathbf{v}_k$ and $\mathbf{n}_k$ are the process and measurement additive random noise, respectively [7].
Given the probabilistic form of SLAM problem, it requires a well defined structure under static world assumption for which the current observation $\mathbf{z}_k$ is independent of previous states, observations and control inputs depending only on the current state $\mathbf{x}_k$ and Markov assumption. In fact, the state transition is assumed to be a Markov process where the next state $\mathbf{x}_k$ depends only on the state $\mathbf{x}_{k-1}$ and the applied control $\mathbf{u}_k$ and not on the observations or the map.
Thus, the SLAM algorithm is implemented in a standard two recursive steps: time-update (prediction) and measurement-update (correction) [9].
The state estimation with Bayesian filter starts by the computation of the a-priori probability $p(\mathbf{x}_0) = p(\mathbf{x}_0|\mathbf{z}_0)$ and proceeds with the following steps iteratively:

- ***Prediction step***:

$$\begin{aligned} &p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}, \mathbf{x}_0) \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) p(\mathbf{x}_{k-1}, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \end{aligned} \tag{1.6}$$

- ***Correction step***:

$$p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_{1:k}, \mathbf{u}_{1:k-1}, \mathbf{x}_0) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}, \mathbf{x}_0)}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k})} \tag{1.7}$$

  where the normalizing constant is:

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k}) = \int p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{u}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k}, \mathbf{u}_{1:k}) d\mathbf{x}_k \tag{1.8}$$

Note that in (1.6), the *transition model* is exploited to compute predicted distribution then in (1.7), measurement $\mathbf{z}_k$ at time k is available and is used

to modify the predicted prior from the previous time step to obtain posterior distribution of the state.
Once a posterior of the current state $\mathbf{x}_k$ is computed, the state estimate $\hat{\mathbf{x}}_k$ can be defined using minimum mean square error (MMSE) or maximum a posteriori (MAP) criteria [7].
According to them, the state estimate are defined by:

$$\hat{\mathbf{x}}_k^{MMSE} = \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k \tag{1.9}$$

$$\hat{\mathbf{x}}_k^{MAP} = \arg\max_{x_k} p(\mathbf{x}_k|\mathbf{z}_{1:k}) \tag{1.10}$$

If the posterior distributions are Gaussian, the MAP and MMSE estimates coincide.
In Fig. 1.1 the recursive Bayesian filter steps are shown:

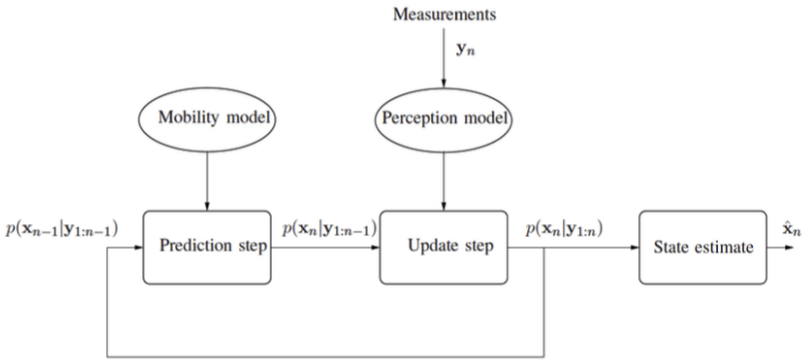


Figure 1.1: Block diagram of a recursive Bayesian filter [8]

This recursive Bayesian filter is a generic approach for filtering but the computation of the different terms of the posterior probability is difficult and closed-form analytic solutions are only available for very specific problems, e.g. for a linear system with Gaussian random variables. The filtering strategies can be exploited in SLAM context to merge data provided by different types of sensors, for example inertial measurement unit (IMU) or odometry, to improve the localization and mapping results.
A convenient way to describe this structure is via the dynamic Bayesian network (DBN), a graphical model that describes a stochastic process as a directed graph as in Fig. 1.2. Here, the arrows entering in $\mathbf{z}_k$ represent the observation model while the two arrows leading $\mathbf{x}_t$ represent the transition model. This way is well suited to describe filtering processes and it is used to tackle the on-line SLAM problem because DBN highlights temporal structure.

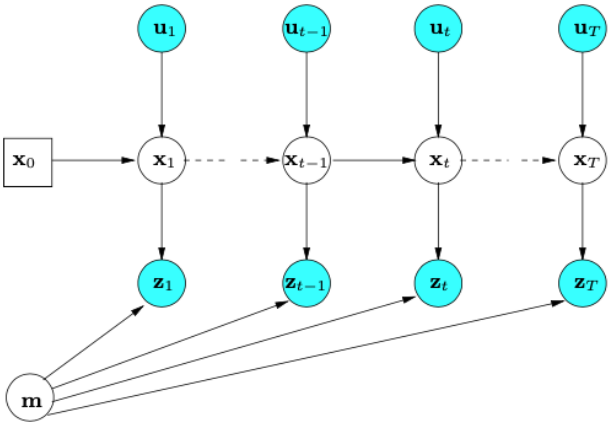


Figure 1.2: Dynamic Bayesian Network of the SLAM process [13]

1.2.2 Kalman Filter

Kalman filter comes from Bayesian filter and it is used when the observation model and transition model are linear and both process and measurement noise are independent Gaussian variables with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively.
From Bayesian equations, the models at time k can be rewritten according to (1.4) and (1.5) to the form [10]:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{v}_k \tag{1.11}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k \tag{1.12}$$

where $\mathbf{F}$ and $\mathbf{H}$ are matrices defining the linear functions.
As the classical Bayesian filter, it provides two recursive steps :

- ***Prediction step***: the state at time k, given the observation up to time k, and covariance matrix $\mathbf{P}_{k|k}$ of $\mathbf{v}_k$ are computed.

$$\begin{aligned} \mathbf{x}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{x}_{k-1|k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{1.13}$$

  where $\mathbf{x}_{k|k-1}$ indicates the predicted state for time instant k given the measurements until the previous time step $k-1$.

- ***Correction step***: the updated state $\mathbf{x}_{k|k}$ and updated covariance $\mathbf{P}_{k|k}$ are computed.

$$\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k\widetilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \end{aligned} \tag{1.14}$$

where

$$\widetilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k|k-1} \tag{1.15}$$

is the "innovation" term that can be interpreted as the deviation of the actual measurement from the predicted one and

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{1.16}$$

is Kalman gain matrix.

However, if (1.4) and (1.5) deviate from linearity other solutions can be exploited in SLAM context:

- *Extended Kalman Filter* (EKF) differs from the general case of Kalman filter because *g()* and *h()* are non-linear functions. The transition and observation matrices are approximate with first time Taylor expansion [10]:

$$\begin{aligned} \mathbf{F}_k &= \left.\frac{\partial f}{\partial x}\right|_{\mathbf{x}_{k-1|k-1},\mathbf{u}_k} \\ \mathbf{H}_k &= \left.\frac{\partial h}{\partial x}\right|_{\mathbf{x}_{k|k-1}} \end{aligned} \tag{1.17}$$

  However, the first order approximation can introduce errors in the posterior mean and covariance which may lead the performance.

- *Unscented Kalman Filter* (UKF) is based on a deterministic sampling technique known as *unscented transformation* (UT) which uses a set of $2n+1$ appropriately chosen weighted points, called *sigma-points*, to parametrize the mean and covariance of probability distributions [11] that undergoes a non linear transformation. The computational cost is the same as EKF but the performance is superior and accuracy of 3rd order can be achieved avoiding linearization.
  Given an n-dimesional random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ is approximated with *sigma-points* to obtain a matrix $\mathbf{X}$ with *sigma-vectors* $\mathbf{X}_i$ with $i = 0, ..., 2n+1$:

$$\begin{aligned} X_0 &= \bar{\mathbf{x}} \\ X_i &= \bar{\mathbf{x}} + \left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i \\ X_{i+n} &= \bar{\mathbf{x}} - \left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i \end{aligned} \tag{1.18}$$

with the weights :

$$\begin{aligned} W_0 &= \frac{\kappa}{(n+\kappa)} \\ W_i &= \frac{1}{2(n+\kappa)} \\ W_{i+n} &= \frac{1}{2(n+\kappa)} \end{aligned} \tag{1.19}$$

where κ is a scaling parameter and $\left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i$ is the i-th row of matrix square root.
Successively, the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_{yy}$ of transformed points are obtained as follows:

1. The sigma-vectors are propagated through the non-linear fucntion:

$$\mathbf{Y}_i = g\left[\mathbf{X}_i\right] \tag{1.20}$$

2. The mean and covariance are computed:

$$\begin{aligned} \bar{\mathbf{y}} &= \sum_{i=0}^{2n} W_i \mathbf{Y}_i \\ \mathbf{P}_{yy} &= \sum_{i=0}^{2n} W_i \left\{\mathbf{Y}_i - \bar{\mathbf{y}}\right\} \left\{\mathbf{Y}_i - \bar{\mathbf{y}}\right\}^T \end{aligned} \tag{1.21}$$

The unscented Kalman filter is a straightforward extension of the unscented transformation.
In prediction step, after the *sigma-points* x_j have been propagated through the non-linear transition model

$$y_j = g(x_j), \qquad j = 1, ..., 2n \tag{1.22}$$

they are weighted and the predicted mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_{yy}$ are estimated with eq. 1.21.
In the correction step, the predictions are updated thanks to the current measurement $\mathbf{z}$. Given the predicted mean and covariance estimates, a new set of *sigma-points* are computed and they are propagated through non-linear measurement model:

$$z_j = h(x_j), \qquad j = 1, ..., 2n \tag{1.23}$$

Successively, the empirical mean $\bar{\mathbf{z}}$ and covariance $\mathbf{S}_z$ of these transformed points are computed. After the cross-covariance matrix $\mathbf{C}_{xz}$

$$\mathbf{C}_{xz} = \sum_{j=0}^{2n} W_j \left(\mathbf{x}_j - \bar{\mathbf{y}}\right) \left(\mathbf{z}_j - \bar{\mathbf{z}}\right)^T \tag{1.24}$$

and Kalman gain $K = \mathbf{C}_{xz}\mathbf{S}_z^{-1}$ have been computed, finally the corrected mean and covariance can be estimated.

1.2.3 Particle Filter

The last filtering approach we present is the particle filter (PF) that relaxes some assumption of the Kalman filters. In particular, this filter is suited in situations when the system deviates from the linearity conditions and also from Gaussian world assumptions [7].
The main idea is to approximate the all posterior probability density functions at time step k, $p(\mathbf{x}_k)$, with a set of M independent weighted samples or particles as follows:

$$p(\mathbf{x}_k) \approx \sum_{m=1}^{M} \omega_{k,m} \delta(\mathbf{x}_{k,m} - \mathbf{x}_k) \tag{1.25}$$

where δ is the Dirac function centred in the location of the sample, $\mathbf{x}_{m,k}$ contains samples (i.e., particles) of the unknown state and the $\omega_{k,m}$ is called *importance weight* and they are normalized.
However, it is difficult or, sometimes, impossible to obtain samples directly from $p(\mathbf{x}_k)$. Consequently, it is important to select a different proposal distribution π from which the samples can be easily drawn. This method is called *importance sampling* [6] and an example is shown in Fig. 1.3. About this, it can be considered to be able to evaluate each distribution at given $\mathbf{x}_m$ and compute the weight as $\omega_m = p(\mathbf{x}_m)/\pi(\mathbf{x}_m)$. This method permits to obtain samples that compensate for the difference between the two distributions. It is important that in all locations where p is larger than zero also π must be larger than zero.
The particle filter implements a recursive filter and therefore includes a prediction and a correction step.
An example is the bootstrap filter. At every time instant, it performs the propagation of the particles, the computation of particles weights and resampling. In the prediction step, the samples are propagated through the motion model and the particles at $\mathbf{x}_{k+1}$ are generated from the priors $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$. In the correction step, when a new measurement $\mathbf{y}_k$ arrives, the weights of the

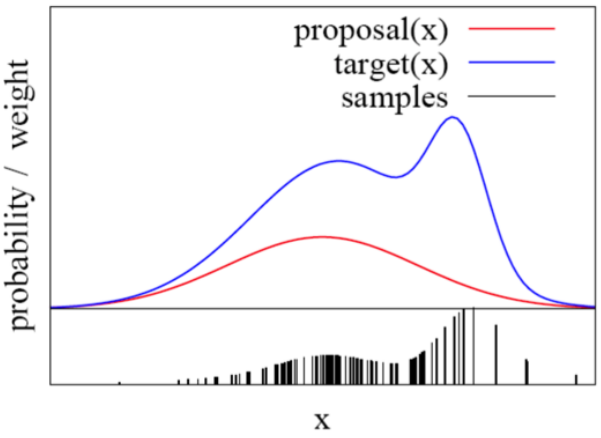


Figure 1.3: Importance sampling principle [12]

particles are computed as $\tilde{\omega}_{k,m} = p(\mathbf{y}_k|\mathbf{x}_{k,m})$ and then they are normalized. Then, the point estimate of the unknown states $\hat{\mathbf{x}}_k$ is also computed. A resampling strategy is used to overcome the problem of particles degradation due to the increase of the variance of the weights over time, which occurs at each iteration.

1.3 Graph-Based SLAM

There is an alternative representation to DBN called "graph based" or "network based". This formulation, unlike DBN, highlights the underlying spatial structure and constructs a graph out of the raw sensor measurement [13]. This graph is called *pose graph*.
In graph-based SLAM the agent's poses are modeled by nodes in a graph and the edge between them represents the spatial constraints, resulting from observations $\mathbf{z}_k$ or from odometry measurements $\mathbf{u}_k$, between two poses. A constraint consists in a probability distribution over the relative transformations between the two poses. As the Fig. 1.4 shows, an edge between two nodes can be created when the agent moves between two consecutive poses, for example from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$, corresponding to odometry measurement or when the agent observes the same part of the environment from $\mathbf{x}_i$ and $\mathbf{x}_j$.
Once the graph is constructed one seek to find the configuration of nodes' position that is consistent with the measurement constraints solving an error minimization problem.

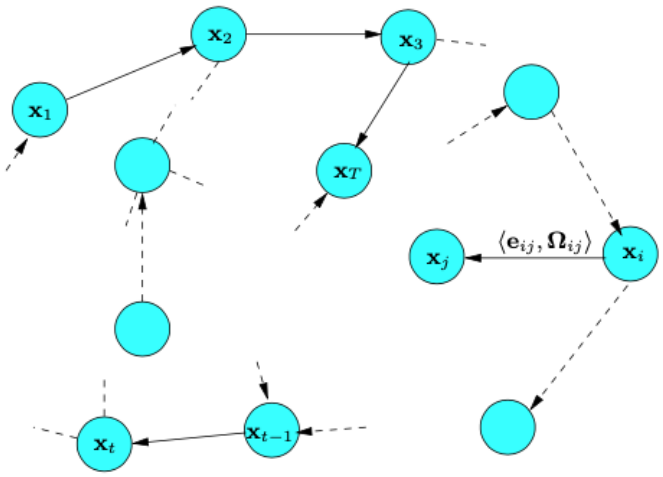


Figure 1.4: A pose graph representation of the SLAM problem [13]

Thus, the problem is decoupled in two tasks as shown in Fig. 1.5: constructing the graph from the raw measurements (graph construction or *front-end*), determining the most likely configuration of the poses given the edges of the graph (graph optimization or *back-end*). The first one is heavily sensor-dependent, while the second one relies on an abstract representation of the data which is sensor independent. Front-end and back-end must be interleaved to design a good SLAM system.

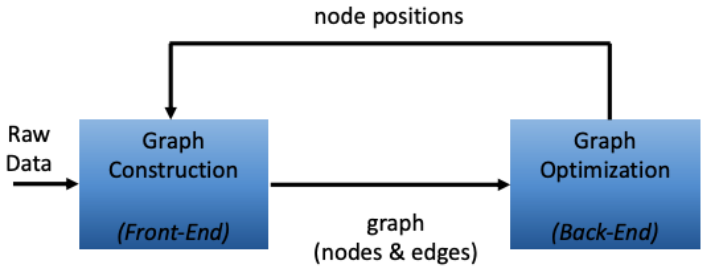


Figure 1.5: The interplay between Front-end and Back-end

1.3.1 Front-End

A more detailed explanation of graph construction is described [13] :

1. Whenever the agent moves or rotates more than an arbitrary threshold,

a new node is added to the graph and it is labeled with the current sensor observation.

2. The SLAM algorithm can use scan matching between two scans to determine a transformation. The current scan is matched with the previously to improve odometry estimate and the corresponding edge between the two nodes is added. In this step, a covariance estimate of the relative pose between neighbouring nodes is also required attaching it to the edges of the graph.

3. An edge is added to the graph also when the agent revisits an already visited area and the algorithm finds a matching between the current and past measurement. This step is known as *loop detection* and these edges are called *loop closure* edges because literally close the loop of the agent's movement in the graph. Also here, a covariance needs to be provided from the alignment process. Furthermore, the accumulated error over the loop is computed. This is crucial information for back-end.

4. The graph optimization procedure is performed whenever a loop closure is detected.

1.3.2 Back-End

If the observations are affected by Gaussian noise, the standard optimization procedure is a least square minimization because optimizing the graph with this method is equivalent of computing a Gaussian estimation of the posterior distribution over the agent trajectory.
Let $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_K)^T$ be a vector in which each element represents the pose of node i and let $\mathbf{z}_{ij}$ and $\mathbf{\Omega}_{ij}$ be respectively the mean and the information matrix from the ith pose to the jth pose. The information matrix is the inverse of the covariance of the pose and represents the uncertainty of the measurement. If the information matrix contains large values, it means that the collected data are informative with small uncertainty. Moreover, this means that the bigger $\mathbf{\Omega}_{ij}$, the higher confidence measurement and the more the edge matters in the optimization process.
Let $\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ be the prediction of measurements given a configuration of the nodes $\mathbf{x}_i$ and $\mathbf{x}_j$; this is usually the transformation between two nodes.
Now, we define the error function $\mathbf{e}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij})$ as the difference between the expected observation $\hat{\mathbf{z}}_{ij}$ and real observation $\mathbf{z}_{ij}$ provided by the agent:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \tag{1.26}$$

The log-likelihood l_{ij} of a measurement $\mathbf{z}_{ij}$ is proportional to

$$l_{ij} \propto \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j)^T \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j) \tag{1.27}$$

Fig. 1.6 illustrates the functions and the quantities that define an edge of the graph.

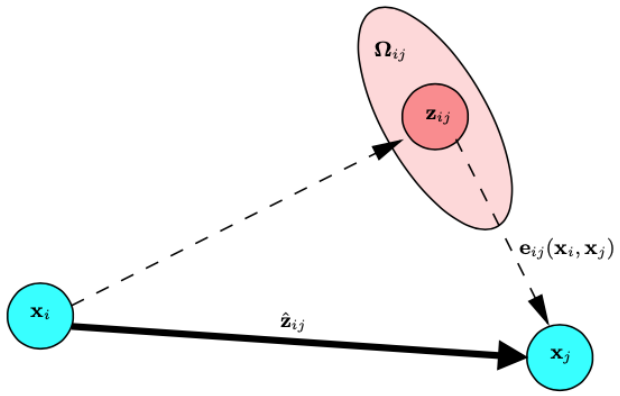


Figure 1.6: Aspects of an edge connecting the vertex $\mathbf{x}_i$ and the vertex $\mathbf{x}_j$ [13]

The goal of the maximum likelihood approach is to find the configuration of nodes $\mathbf{x}^*$ that minimizes the negative log-likelihood for all observations :

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \tag{1.28}$$

with

$$\mathbf{F}(x) = \sum_{(i,j) \in C} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij} \tag{1.29}$$

and where C be the set of pairs of indices for which an observation $\mathbf{z}$ exists. Note that the error of a constraint $\mathbf{e}_{ij}$ depends only on the relative position of the connected poses $\mathbf{x}_i$ and $\mathbf{x}_j$. Accordingly, the error $\mathbf{F}(\mathbf{x})$ of a particular configuration of the poses $\mathbf{x}$ is invariant under a rigid transformation of all the poses.
If the initial guess $\tilde{\mathbf{x}}$ of the agent's poses is known, a numerical solution of the Eq. (1.28) can be obtained using, for example, Gauss-Newton or Levenberg-Marquardt algorithms.

1.4 Mapping

The map can be represented in different ways depending on the sensors used, on the environment's characteristics and on the estimation algorithm. They can be subdivided into *metric map* or *semantic map*; the main differences between them is that the first one consists of a symbolic structure that encodes the geometry of the environment while the second consists of associating semantic concepts to geometric entities in an agent's surrounding.
Regarding metric maps, in the following, two predominant paradigms are introduced [5]:

- *Landmark-based maps*: models the environment as a sparse set of landmark; this solution is often preferred in environments where locally distinguishable features can be identified and especially when sensors like cameras or laser are used.

- *Occupancy grid maps*: discretizes the environment in cells and assigns a probability of occupancy to each cell; this solution is usually used in conjunction with some less accurate sensor data, for example with radar, or when no well-defined landmarks are identifiable.

1.4.1 Occupancy Grid Mapping

To address the problem of the occupancy grid mapping update, the Bayesian filtering theory can be exploited.
Let N be the number of targets detected by the sensor with the measurement $z_t = \cup_k^N \left(\rho_t^k, \phi_t^k\right)$ at time t where $\left(\rho_t^k, \phi_t^k\right)$ are range and bearing angle of kth target respectively.
The posterior probability $p(\mathbf{m}|\mathbf{x}_{1:t}, \mathbf{z}_{1:t})$ represents the occupancy probability of the map $\mathbf{m}$ considering all the measurement $\mathbf{z}_{1:t}$ and agent pose $\mathbf{x}_{1:t}$ up to time t [14]. If cells m_i of the map are independent, the posterior becomes as follows:

$$p(\mathbf{m}|\mathbf{x}_{1:t}, \mathbf{z}_{1:t}) = \prod_i p(m_i|\mathbf{x}_{1:t}, \mathbf{z}_{1:t}) \tag{1.30}$$

To avoid numerical instability, the log-odd form is used to represent the probability for each cell as shown in 1.31:

$$l_{t,i} = log \frac{p(m_i|\mathbf{x}_{1:t}, \mathbf{z}_{1:t})}{1 - p(m_i|\mathbf{x}_{1:t}, \mathbf{z}_{1:t})} \tag{1.31}$$

The log-odd probability for each cell, recursively, is updated using Bayes filter with equation 1.32:

$$l_{t,i} = l_{t-1,i} + log\frac{p(m_i|\mathbf{x}_t, \mathbf{z}_t)}{1 - p(m_i|\mathbf{x}_t, \mathbf{z}_t)} - l_{0,i} \tag{1.32}$$

where $l_{0,i}$ is the log-odd at $t = 0$.

Chapter 2

Registration Algorithms

In this chapter, we describe the registration problem that allows to estimate the trajectory of the agent by matching different measurements collected by sensors during its movement. There are different types of sensors that can be used to perform SLAM. Opposite to cameras, most of them are range sensors e.g. radar, lidar, sonar that provide a cloud of points from a detected object. In this chapter, different registration techniques are proposed to estimate the relative pose of sensors through i.e. translation and rotation, between different point clouds. We will present both local matching methods like iterative closest point (ICP) or normal distribution transform (NDT) matching and global methods like correlative scan matching (CSM).
We will also present a Fourier-based method that provides a way to estimate all rigid parameter transformations, i.e. translation, rotation and scale between different images. Although it is a method born for image registration, if we process raw sensor data, it can be used to estimate how the sensor moves during the motion. In particular, when the radar is considered this method works on all radar data avoiding the problem of landmarks extraction that are very difficult with sparse and noisy data.
Finally, we will present the Matlab functions used to estimate the relative pose between two consecutive sensor's measurement from scan matching and to map the environment. They will be used in this thesis for the development of the proposed SLAM algorithm.

2.1 Registration Problem

Registration is the problem of matching two scans (or points clouds) when the relative pose difference between the scans is unknown. In the follow-

ing, *scan* is referred to a sensors observations and it is denoted by a vector $\mathbf{h} = \{\mathbf{h}_i\}_{i=1,...,n}$ where each element $\mathbf{h}_i$ can represent the coordinate of 2D points with $\mathbf{h}_i \in \mathbb{R}^2$ or 3D space with $\mathbf{h}_i \in \mathbb{R}^3$. Moreover, we will refer to *frame* the coordinates system used to describe the observations gathered by the device sensor.
Given two scans, the output of registration is an estimate of the transformation needed to transform one scan (called *current scan*) into the correct pose in the coordinate system, or *frame*, of the other scan (called *reference scan*). In a 2D scenario, a transformation involving a translation and a rotation can be defined as a function T given by:

$$T : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2.1}$$

or in compact form by:

$$T : \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{R}(\theta) \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{t} \tag{2.2}$$

where $\mathbf{t} = (t_x, t_y)^T$ describes the translation and

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{2.3}$$

is the rotational matrix where θ is the rotation between the two different frames.
The goal of the scan alignment is to recover the parameters t and θ using the scans taken at two positions.

2.2 ICP: Iterative Closest Point Algorithm

The iterative closest point was introduced in [15]. It is still widely used for point clouds registration in robotic fields. The ICP algorithm refines the relative pose between two point sets from scanned surfaces by iteratively minimizing the sum of squared distances between pairs of closest points.
Let $P = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$ and $Q = \{\mathbf{q}_1, ..., \mathbf{q}_m\}$ be two points clouds that have to be aligned, and $\mathbf{t}_0$ and θ_0 an initial guess for the translation and rotation. The algorithm, at each iteration k, transforms P according to the current estimates of translation $\mathbf{t}_k$ and rotation θ_k.
Let $P'_k = \{\mathbf{p}'_1, ..., \mathbf{p}'_n\}$ be the transformed data set at iteration k with $\mathbf{p}'_i =$

$\mathbf{R}(\theta_k)\mathbf{p}_i + \mathbf{t}_k$ from (2.1), now the algorithm forms n pairs $\left(\mathbf{p}'_i, \mathbf{q}_j\right)$ by associating each point in P'_k with the closest point in Q.
The goal is to minimize the sum of the squared Euclidean distances until the convergence:

$$(\mathbf{t}_{k+1}, \theta_{k+1}) = \arg\min_{(\mathbf{t},\theta)} \sum_{(i,j)} ||\mathbf{R}(\theta)\,\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j||^2 \tag{2.4}$$

However, there are some limitations regarding the ICP algorithm. For example, it only considers a "point-to-point" registration therefore the nearest neighbour point does not, in general, correspond to the same point on the scanned surface in challenging environments. Moreover, if the scan includes a large number of points, a point-to-point search can lead to high computational costs and execution time. In the field of lidar sensors where each data set has many points, this algorithm is well suited but with radars, it is not true because the data sets are very poor and a direct match can lead to inaccurate results.

2.2.1 PLICP: Point-to-Line Iterative Closest Point Algorithm

In [16] the authors have proposed an improved ICP algorithm such as iterative closest point (PLICP) for "point-to-line" matching. It can converge quadratically in a finite number of steps.
Different from the classic ICP algorithm, after the transformed points set P'_k have been found, for each point $\mathbf{p}'_i$ the algorithm finds the two closest points $\mathbf{q}_{j_1}$ and $\mathbf{q}_{j_2}$ in the data point set Q. Let $\mathbf{C}_k$ be a set of tuples $\langle i, j_1, j_2 \rangle$ at step k that represents all the point-to-line correspondences. The goal is to minimize the sum of the squares of the distances from point i to the line containing the segment $j_1 - j_2$ [16] to obtain the optimal transformation parameters $\mathbf{t}$ and θ:

$$J(\mathbf{t}_{k+1}, \theta_{k+1}, \mathbf{C}_k) = \sum_{(i)} \left(\mathbf{n}_i^T \left[\mathbf{R}(\theta_{k+1})\,\mathbf{p}_i + \mathbf{t}_{k+1} - \mathbf{q}_{j_1}\right]\right)^2 \tag{2.5}$$

where $\mathbf{n}_i$ is the normal to the surface at the projected points.

2.3 NDT: Normal Distribution Transform

The normal distributions transform (NDT) can be described as a method for compactly representing a surface and it was introduced in [17] for registration of 2D points.

2.3.1 NDT Grid for Representing Surfaces

The NDT maps a point cloud to a smooth surface representation, described as a set of local probability density functions, each of which describes the shape of a section of the surface [18].
First, the space around the agent is subdivided regularly into cells with constant size. Then, from points $\mathbf{x}_{i=1\ldots n}$ in a single cell, a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is associated to each cell by calculating the empirical mean (2.6) and empirical covariance matrix (2.7):

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{2.6}$$

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{2.7}$$

where a single point $\mathbf{x}_i$ is identified with the 2D or 3D coordinates.
Now, the probability of measuring a point in a certain position within each cell is modeled by a normal distribution.

$$p(\mathbf{x}) \approx \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}{2}\right) \tag{2.8}$$

It represents a piecewise continuous and differentiable description of the point cloud in the form of a probability density.
In 2D and 3D cases, the surface orientation and smoothness can be assessed from the eigenvectors and eigenvalues of the covariance matrix within which the diagonal elements denote the variance of each variable, and the off-diagonal elements denote the covariance of the variables. Depending on the proportions of the variances, a 2D normal distribution can be either point-shaped or line-shaped while in 3D case can describe a line (one eigenvalue much larger than the other two), a plane (one eigenvalue much smaller than the others) or a sphere (all eigenvalues approximately equal).

2.3.2 NDT for Scan Registration

When using NDT for scan registration, the first step is to build the NDT of reference scan by the technique mentioned in 2.3.1. As the local registration methods such as those that use NDT or ICP require initial estimates, they must be initialized by zero or using odometry data or another sensor such as an inertial unit (IMU). A good initial guess can boost the algorithm convergence. Then, each point of the second scan is mapped into the coordinate

frame of the first scan using T in 2.1 finding the correspondent normal distribution.
The flow of NDT scan matching is shown in Fig. 2.1.

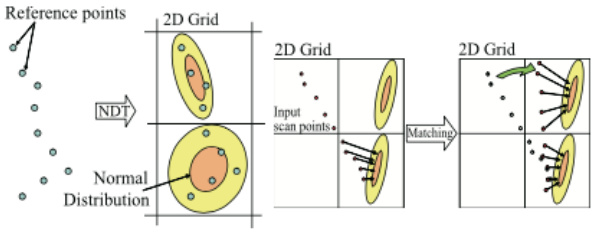


Figure 2.1: NDT converts for reference scan and NDT scan matching flow [19]

.

Let consider the following notation:

- $\mathbf{p} = (t_x, t_y, \theta)$: The vector of the transformation parameters, between reference and current scan, to estimate.
- $\mathbf{x}_i = (x_i, y_i)$: The coordinate of ith 2D point of the current scan in its coordinate frame.
- $\mathbf{x}_i^{'}$: The point $\mathbf{x}_i$ mapped into the coordinate frame of the reference scan according to the parameters $\mathbf{p}$, that is $\mathbf{x}_i^{'} = T(\mathbf{x}_i, \mathbf{p})$
- $\boldsymbol{\Sigma}_i, \boldsymbol{\mu}_i$: The covariance matrix and the mean of the corresponding normal distribution to point $\mathbf{x}_i^{'}$ looked up in the NDT of the reference scan.

The goal of the NDT scan-matching is to find parameters of $\mathbf{p}$ which maximize the sum of the normal distribution of all points $\mathbf{x}_i^{'} = T(\mathbf{x}_i, \mathbf{p})$ with parameters $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\mu}_i$, that is the function:

$$score(\mathbf{p}) = \sum_i \exp\left(-\frac{(\mathbf{x}_i^{'} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_i^{'} - \boldsymbol{\mu}_i)}{2}\right) \tag{2.9}$$

Since optimization problems normally are described as minimization problems, Newton's algorithm can be used to minimize the function $-score$ in $\mathbf{p}$ by iteratively solving linear system [17].
A recent work [20] proposes a new fast points clouds registration algorithm

that can realize fast and accurate localization and mapping through combination of NDT scan matching and PLICP described in subsection 2.2.1. First, the NDT registration algorithm between two adjacent point clouds is applied. Then, PLICP algorithm is adopted to correct the previous rough registration.

2.4 CSM: Correlative Scan Matching

In contrast to algorithms shown in (2.2) and (2.3), correlative scan matching (CSM) [21] exploits a global search method which avoids the local search problem of falling in a local minima failing to identify the global maximum. This method performs a search over the entire area of transformations without requiring an initial guess.
The principle behind the algorithm is the realization of a look-up rasterized table that approximates the distribution shape due to noisy sensors measurements as radially-symmetric distribution [21] making the scan matching easier. The look-up table is built whenever the *current scan* arrives according to the *reference scan*, in fact, it is considered a scan-to-scan matching strategy.
When the reference scan is converted into a probability grid, the scan points $\{\mathbf{h}_i\}_{i=1,\dots,n} \in \mathbb{R}^2$ are separated into cells with size r. If there are some points inside the single cell, it is considered occupied assigned to 1 while other cells are considered free assigned to 0. Now, each cell of the table is smoothed with a radially-simmetric kernel.
The goal is to estimate the relative pose, corresponding to a vector $\mathbf{p} = (t_x, t_y, \theta)$, of the agent at kth iteration respect to the $(k-1)$th iteration. To achieve the most probable transformation, the best *score* must be found. Three solutions are presented in [21]:

1. *Brute Force:* This method exploits three nested loops, one for each transformation parameter. Every point of the transformed current scan is mapped over the rasterized lookup table and the total score is computed as the sum of all mapped cells scores. This algorithm tries all possible translations and rotations from the selected search window W around a given pose $\mathbf{p}_0$ placed in its center. For example, the angular step size can be chosen so that the scan point at the maximum range d_{max} does not move more than r [22]. Then from the law of cosine:

$$d_{max} = \max_{i=1,\dots,n} ||\mathbf{h}_i|| \tag{2.10}$$

$$\delta_\theta = \arccos\left(1 - \frac{r^2}{2d_{max}^2}\right) \tag{2.11}$$

Given the linear and angular size of search window W, the integral number of steps to cover them can be computed. For each iteration the *score* is computed as:

$$score = \sum_{i=1}^{n} \mathbf{p}_0 + (\mathbf{R}_{\delta_\theta}(\theta)\mathbf{h}_i + \mathbf{t}_r) \tag{2.12}$$

where each scan point $\mathbf{h}_i$ is transformed according to the combination of translation and rotation parameters at a given step, around the given pose $\mathbf{p}_0$, by $\mathbf{t}_r$ and $\mathbf{R}_{\delta_\theta}(\theta)$. For these reasons the algorithm is slow.

2. *Computing 2D Slices:* A lot of calculation time can be spared by pre-generating θ values and iterating over them in the outer-most loop. This method is faster than brute force because the trigonometric functions are not computed.

3. *Multi-Level Resolution:* This method constructs two raster look-up tables with different resolutions. A low-resolution grid is used to discover the areas that might contain the global maximum then a high-resolution grid is used to search the global maximum in the restricted area. This strategy is fast making real-time scan matching possible.

To improve the speed of search over the large search window, the branch and bound strategy can be used to compute $\mathbf{p}$ by considering subsets of possibilities as nodes in a tree where the root node represents all possible solutions i.e. search window W [22].

2.4.1 Improved CSM

Since the presented method assigns the same occupancy probability for each point when constructing a rasterized look-up table, there may be different kinds of matching relations which lead to a wrongly estimated pose. Moreover, if the data volume provided from the sensor is small and sparse for example using mmWave radar, the original CSM is not well suited. Considering the above reasons, an improved CSM method is proposed in [23]. Here, the IMU and odometry are used to merge continuous radar scan in a "Multi-Scan" to increase density of point clouds but it is not enough to get a good estimate of the pose. The improvement consists in the exploitation of the radar cross section (RCS) feature of targets to weight the probability distribution of each point in reference scan in addition to the realization of scan-to-map matching algorithm which allows to match current scan with a local submap of the grid global map.

More in detail, the authors in [23] also do raw data preprocessing to achieve better SLAM algorithm accuracy. In this sense, first, they filter out the low precision points due to measurements of sensor and once the multi-scan is formed, the outlier points are deleted from the sensor scan. The outliers are generated by the multiple reflections of signals due to the environment.
The strategy is to iteratively alternate the pose estimation with an update map phase where the map contributes to compute that pose.
Before performing points clouds matching, the global map M was constructed so the environment is subdivided into a grid whose cells are considered independent. The first scan of multi-scan, called query scan Q, will be matched with a local submap called reference scan F, that is, a portion of the global grid map. In order to do that, the submap size must be computed considering Q that must be registered using the absolute pose at the previous time step.
A fundamental step is to compute the look-up table of F with a Gaussian kernel:

$$P(F^*) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(d-\mu)^2}{2\sigma^2}} R(M_{x,y})(t) \tag{2.13}$$

where d is the space between two adjacent cells F e F^* of the grid map, μ and σ are the expectation and the variance of Gaussian kernel which is used to calculate the occupancy probability of the cell F^* and $R(M_{x,y})(t)$ indicates the RCS value of a given cell in position (x,y) at time t.
At this point, the matching between Q and F is performed. The estimated pose at time t is obtained by rotating and translating Q over F with the brute force method of correlative scan matching. Thus, Q is projected over the rasterized look up table F and the RCS of Q is subtracted from the RCS of the grid. Finally, the result is summed. The smaller is the final sum, the best is the pose estimation.
Finally, the map is updated with:

$$R(M_{x,y})(t) = \frac{\sum_{k=0}^{t} \left(\frac{1}{r_{x,y}(k)} R(M_{x,y}(k) \right)}{\sum_{k=0}^{t} \left(\frac{1}{r_{x,y}(k)} \right)}, 0 \leq k \leq t \tag{2.14}$$

where $r_{x,y}(k)$ is the distance from the sensor to the target at time step k.

2.5 Fourier-Mellin Transform

The Fourier-Mellin transform is an FFT-based method which is used to register two different images searching for the optimal match in the frequency

domain. To recover translation, rotation and scale between two different images, this method makes use of the Fourier transform properties [24].
Let $F_1(\xi,\eta)$ and $F_2(\xi,\eta)$ be the Fourier transform corresponding to images $I_1(x,y)$ and $I_2(x,y)$.
Now, if two images differ only by a shift (x_0, y_0) then

$$I_2(x,y) = I_1(x - x_0, y - y_0) \tag{2.15}$$

and their Fourier transform are related by

$$F_2(\xi,\eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} \cdot F_1(\xi,\eta) \tag{2.16}$$

The translation parameters can be recovered by calculating cross-power spectrum between two images defined as :

$$\frac{F_2(\xi,\eta)F_1^*(\xi,\eta)}{|F_2(\xi,\eta)F_1(\xi,\eta)|} = e^{j2\pi(\xi x_0 + \eta y_0)} \tag{2.17}$$

where $F^*(\xi,\eta)$ is the complex conjugate of the Fourier spectrum $F(\xi,\eta)$. It is clear that the inverse Fourier transform of (2.17) is a $\delta-$Dirac function like $\delta(x - x_0, y - y_0)$ centered in (x_0, y_0).
If the two images also differ by the rotation angle (θ_0), then:

$$I_2(x,y) = I_1\left(x\cos(\theta_0) + y\sin(\theta_0) - x_0, -x\sin(\theta_0) + y\cos(\theta_0) - y_0\right) \tag{2.18}$$

and their Fourier transforms are related by:

$$F_2(\xi,\eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} \cdot F_1(\xi\cos(\theta_0) + \eta\sin(\theta_0), -\xi\sin(\theta_0) + \eta\cos(\theta_0)) \tag{2.19}$$

according to Fourier translation and rotation properties.
Let $M_1(\xi,\eta)$ and $M_2(\xi,\eta)$ be the magnitudes respectively of $F_1(\xi,\eta)$ and $F_2(\xi,\eta)$ which are related by:

$$M_2(\xi,\eta) = M_1(\xi\cos(\theta_0) + \eta\sin(\theta_0), -\xi\sin(\theta_0) + \eta\cos(\theta_0)) \tag{2.20}$$

Converting the magnitudes from cartesian to polar coordinates allows to represent rotation as a shift and the relation between them becomes as follows:

$$M_1(\rho,\theta) = M_2(\rho, \theta - \theta_0) \tag{2.21}$$

2.6 Matlab SLAM Functions

2.6.1 Pose Estimation

Before explaining other Matlab functions [25], we describe the *lidarScan* Matlab object. The *lidarScan* contains data referring to a single sensor mesurement. In particular, it is a scan in the 2D plane with distances measured

from the sensor to the targets in the environment at specific angles. The syntax is reported:

$$\text{scan} = \boldsymbol{lidarScan}\,(\text{ranges}, \text{angles})$$

where *ranges* and *angles* are the two vectors of the same length. The pair formed by the ith element of each vector identifies a point of the scan. Among the properties, there are the two vectors of equal length, for ranges and for angles but also there is an associated n-by-2 matrix with the target's cartesian coordinates. Finally, also the number of targets in the scan can be provided. To estimate the relative 2D pose between two sensor scans, Matlab provides matching functions that are built for lidar applications but, here, we adapt them for the mmWave radar sensor measurement. The functions are:

- *matchScans*: finds the relative pose between two consecutive *lidarScans* using normal distribution transform described in section 2.3. Among the properties provided as output by this function, there is the *score*. To improve the matching, the initial guess of the relative pose between current and reference scan can be supplied in input specifying the 'InitialPose' argument. The syntax of this function is shown:

$$\text{pose} = \boldsymbol{matchScans}\,(\text{currScan}, \text{refScan})$$

- *matchScansGrid*: finds the relative pose between two consecutive *lidarScans* using grid-based search. The function transforms two scans in probabilistic grids and it finds the relative pose as described in section 2.4. The users can set the initial guess of the relative pose between the two scans specifying in input the 'InitialPose' and can indicate the search windows in meters and also in radians around the initial pose estimate specifying the 'TranslationSearchRange' and the 'RotationSearchRange' arguments. These properties increase the speed of the algorithm. Also, this function outputs the *score*. The syntax for this function is also shown:

$$\text{pose} = \boldsymbol{matchScansGrid}\,(\text{currScan}, \text{refScan})$$

For each mentioned function, the relative pose between the current scan and reference scan in a vector $[x, y, \theta]$ where $[x, y]$ are in meters and θ in radians while the currScan and the refScan are *lidarScan* objects.

2.6.2 Mapping

Matlab provides algorithms in term of occupancy grid map to represent the sensor workspace. The map updating policy is provided by *insertRay* Matlab function depending on the types of the map is used. This function inserts rays corresponding to sensor observations i.e. above mentioned *lidarScan* object, starting from the sensor pose to the position of detected targets in the environment. The syntax of *insertRay* is reported:

$$\textbf{\textit{insertRay}}(\text{map}, \text{pose}, \text{scan}, \text{maxRange})$$

where the input arguments represent the occupancy grid map object, the pose where the measurement is taken, the corresponding *lidarScan* object and the maximum range of sensor respectively.
There are two types of occupancy map in the Matlab toolbox:

- *binaryOccupancyMap*: the probability of cells being occupied is a logical value that can be *true* (1) for occupied cells or *false* (0) for free cells. This type is preferred if one needs to preserve the memory in the application. The syntax to build a binary occupancy grid map is :

$$\text{map} = \textbf{\textit{binaryOccupancyMap}}(\text{width}, \text{height}, \text{mapResolution})$$

- *occupancyMap*: it is more detailed than the previous one because the probability inside the cells can assume all the values included between 0 and 1. The value inside the cell is close to 1 when it is likely an obstacle is present in that cell. On the other hand, the values close to 0 indicate probably free cells i.e. an empty cell. For this map, the syntax is:

$$\text{map} = \textbf{\textit{occupancyMap}}(\text{width}, \text{height}, \text{mapResolution})$$

For both functions, the *width* and *height* arguments represent the size of the map, in meters, and *mapResolution* indicates the grid discretization resolution.
For the binary occupancy map, the *insertRay* function updates the map by setting *true* the cell corresponding to the location of scan endpoints that is where the obstacle is presumed to be. If the ranges of endpoints are higher than the *maxrange* input argument, these are considered empty cells. Also, all the other points along the ray are treated as obstacle-free and the corresponding cells are set to *false*.
For the occupancy map instead, the update follows the inverse sensor model for range sensors in Fig. 2.2.

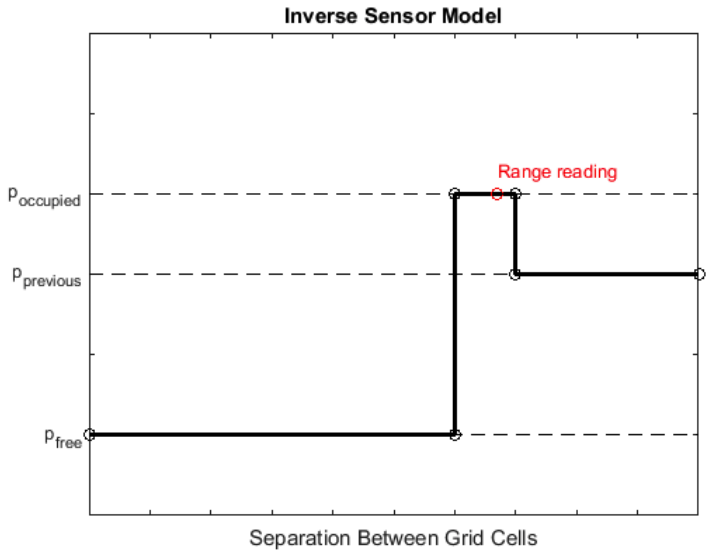


Figure 2.2: Inverse sensor model [25]

According to the model, grid locations before the range reading are set as free, the cell where is the sensor reading is set as occupied and all distances after the reading are not updated. The probability values of occupied ($p_{occupied}$) and free (p_{free}) locations can be customized by the user.

Chapter 3

Radar Technologies

RADAR whose stands for radio detection and ranging [26] is an electrical system capable of detecting the presence of an object, namely *target*, in the surrounding space using radiofrequency electromagnetic waves. Although the radars were born to detect objects and determine their distance for military applications due to their bulky size and high cost, nowadays thanks to modern high frequency radar systems, it is possible to exploit them also in the civilian domain to track, identify, image, and classify targets. For this reason, the fields of application of these systems are constantly expanding. Among these applications, there are vehicle tracking and self-driving, collision avoidance, 2D and 3D mapping of indoor and outdoor environments, Earth resource and precipitations monitoring.
Nowadays, sensors like lidar and optical camera are widely used in applications that implement SLAM algorithms because lidar sensors have very high precision and they are able to accurately model surrounding 3D environment and optical cameras provide a wealth of detailed information. From this point of view, the modern low cost and high frequency radars may be not suitable for this application because due to small and sparse volume of information and sometimes they may contain some errors because of noise and multiple paths of the backscattered signal. The reflection of backscattered radar signals depends on the surface and on the incidence angle β (Fig. 3.1). At mmWave radar frequencies, the diffuse component of reflected signals are very low compared to the specular reflection component. In particular, if the incidence angle β is large, the wave can be mirrored and it may be not detected by the radar if the transmitting antenna is close to receiving antenna.

Due to non-ideal radiation pattern of radar, we may also have the contributions of "ghost object" i.e. reflections that appear to come from objects that are not there. As it is intuitive, these problems are more present in indoor

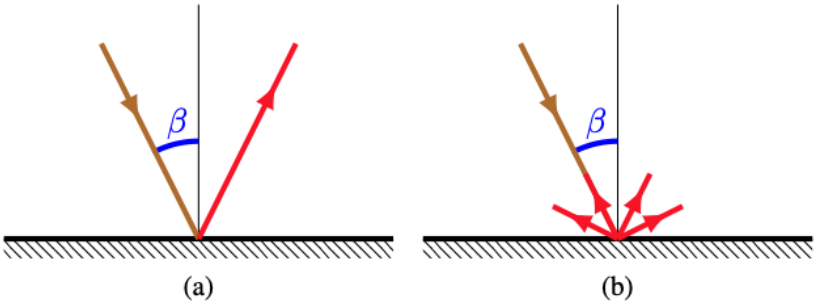


Figure 3.1: (a)specular reflection, (b)diffuse reflection [14]

environments that have a large number of reflective objects rather than in outdoor environments.
However, modern mmWave radar sensors may be exploited in challenging environment conditions and due to their small size, they can be suitable for portable indoor systems. Propagating radio waves in the environment in the order of tens GHz, radars can work within certain reliability even with low-visibility condition due to rain, smoke, fog or for example poor lighting in contrast to other competitive technologies like Lidar or optical cameras.
Generally, the main elements involved in the Radar detecting process are transmitter (TX), receiver (RX), antenna and signal processor as shown in Fig. 3.2.
In this chapter, first, fundamentals of radar systems are presented. Second, we present a type of mmWaves radar systems used to measurement campaign makes in this work.

3.1 Principles of Radar

Fig. 3.2 shows the main functional blocks composing a radar. The signal is generated by the transmitter and it is irradiated in the atmosphere by the antenna. The electromagnetic wave after propagating through the enviroment is reflected back by the target (when present) and it is processed by receiver circuits. The backscattered signal in general experiences interference [26] like noise due to electronic components and environment, electromagnetic interference from the external signal generated by other users, clutter due to the reflection due to objects not of interest and, possibly, jamming from electronic countermeasures systems. The signal processor situated after the receiver block has the function to detect the signal in the presence of the above mentioned interference sources.

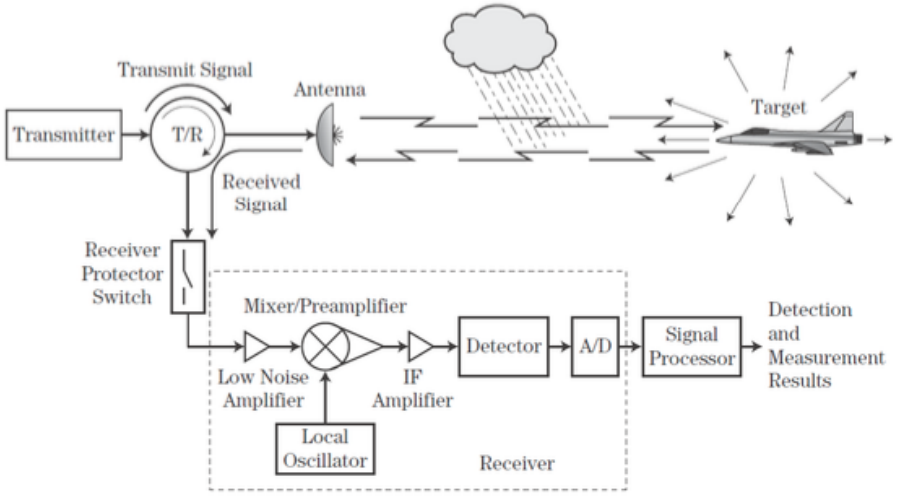


Figure 3.2: A schematic view of the TX/RX processing chain of a radar [26]

The radar systems may be different according to the type of signal generated by the transmitter: *Continuous wave* (CW) radars continuously transmit signal in the propagation medium and *pulsed wave* radars transmit finite duration short impulses to avoid the problem of time overlapping between TX and RX signals.
Radars can be also classified according to the configuration of antennas in *monostatic* or *bistatic* radar. In a monostatic configuration, there is only one antenna that serves both the transmitter and receiver. On the other hand, the radar is said in the bistatic configuration if there are two different antennas, one for the transmitter and one for the receiver, sufficiently separated and isolated.
Now, we present some equations that describe the radar principles and show its dependencies on different parameters.
Let P_T be the transmitted power and let G_T be the transmitter antenna gain, the power density at distance R is:

$$S_T = \frac{P_T G_T}{4\pi R^2} \quad \left[\frac{W}{m^2}\right] \tag{3.1}$$

Equation 3.1 shows how the power decreases with the quadratic value of the distance between radar and target. The receiving power density from the

target can be expressed as:

$$S_R = \frac{P_T G_T}{4\pi R^2} \frac{\sigma}{4\pi R^2} \quad \left[\frac{W}{m^2}\right] \tag{3.2}$$

where σ is radar cross section (RCS) that is a parameter measuring the ability of a target to reflect electromagnetic waves so that the greater is RCS the more the object is detectable by the radar. Note that in general, the RCS is the frequency and angle dependent.
To compute the received power, we need to multiply the receiver power density by $A_R = \frac{G_R \lambda^2}{4\pi}$ that is the effective antenna area where λ is the signal wavelength. Therefore, the received power is:

$$P_R = \frac{P_T G_T \sigma}{(4\pi R^2)^2} A_R \quad [W] \tag{3.3}$$

Finally, the received power for a monostatic radar can be expressed as:

$$P_R = \frac{P_T G_T G_R \sigma \lambda^2}{(4\pi)^3 R^4} \quad [W] \tag{3.4}$$

From 3.4 which is the well-known radar equation, it is possible to design the parameters such as P_T or λ to achieve desired power value according to the sensibility of the receiver.

3.1.1 FMCW Radars

Among the family of CW radars, there are frequency modulation continuous wave (FMCW) radars. These radar systems are able to detect the target and to estimate target-related parameters like range, angle of arrivals and velocity [27]. The FMCW technique is based on *chirp*, a sinusoid signal shown in Fig. 3.3 whose frequency increases linearly with time. It is characterized by a bandwidth B, start frequency f_c and duration T_c. A convenient way to represent a chirp is a graph with frequency as a function of time in which the slope $S = B/T_c$ is highlighted. It represents the rate at which the chirp ramps up.

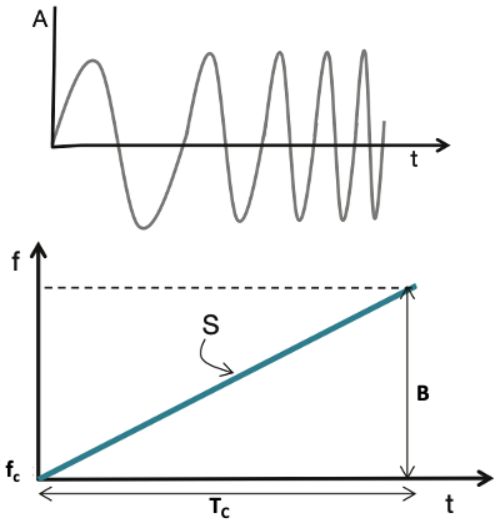


Figure 3.3: Top: Example of a chirp signal; Bottom: Slope as a function of time [27]

Now, the block diagram of a FMCW radar is reported in Fig. 3.4.

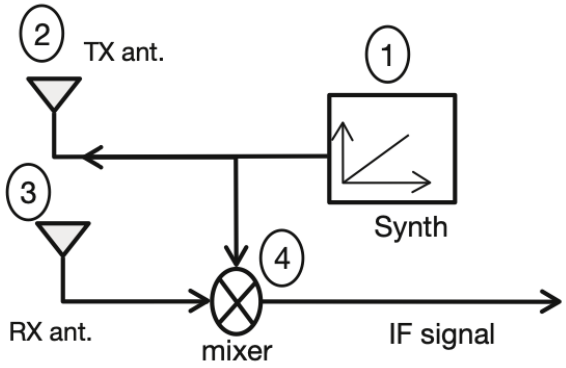


Figure 3.4: A block diagram of a FMCW radar [27]

The synthesizer (1) generates a chirp which is transmitted in the propagation channel by the TX antenna (2).

$$s_T(t) = A\sin(2\pi f_{TX}(t) \cdot t) \tag{3.5}$$

where $f_{TX}(t) = (f_c + St)$. The chirp is reflected back by the target and it is received at RX antenna (3).

$$s_R(t) = A\sin(2\pi f_{RX}(t) \cdot (t - \tau)) \tag{3.6}$$

where $f_{RX}(t) = (f_c + S(t - \tau))$. An electronic circuit called "mixer" (4), combines the TX and RX signals generating an intermediate frequency $f_{IF} = f_{TX} - f_{RX}$ signal whose frequency allows to calculate the distance of the target.
The RX chirp is a time-delay version of TX chirp with round-trip delay defined in 3.7:

$$\tau = \frac{2d}{c} \tag{3.7}$$

where d is the distance to the object and c is the speed of light.
If we consider to have only one target in the field of view (FoV) of the radar, then the output signals will be a sinusoid $A\sin(2\pi f_{IF}t + \phi_o)$ with constant intermediate frequency $f_{IF} = S\tau = S2d/c$ (Fig. 3.5) and phase $\phi_o = 2\pi f_{IF}\tau = 4\pi d/\lambda$. Thus, the distance of the target can be defined as:

$$d = \frac{f_{IF}c}{2S} \tag{3.8}$$

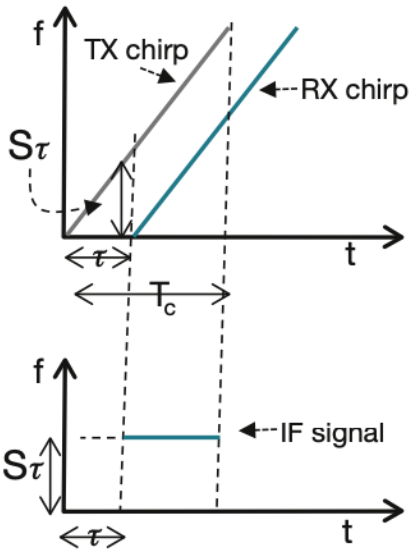


Figure 3.5: Top: Slope of the TX and RX chirp, τ is the delay of the signal coming from a target. Bottom: IF frequency related to the detected target [27]

On the other hand, if we detect some different objects, the received signal will be composed of different tones. In this case, we will use the Fourier transform in order to separate different tones in the frequency domain. The ability to

distinguish between different objects is called *range resolution*. To improve this, it is necessary to increase the duration of the observation window T_c of the chirp until tones are sufficiently separated at the expense of increasing bandwidth.
Since $\Delta f = \frac{2S\Delta d}{c} > \frac{1}{T_c} \Rightarrow \Delta d > \frac{c}{2B}$ with $B = ST_c$, the range resolution equation is given by:

$$d_{res} = \frac{c}{2B} \tag{3.9}$$

We note that it depends only on bandwidth swept by the chirp, the wider bandwidth the higher range resolution thus smaller targets can be identified.
As above mentioned, FMCW radars allow also to measure the velocity of the target. In a single target case, two TX chirps separated by T_c are transmitted. At the receiver, we will obtain two sinusoids with the same IF but different phases. The different $\Delta\phi$ can be exploited to measure the target velocity v. Starting from

$$\Delta\phi = \frac{4\pi\Delta d}{\lambda} \tag{3.10}$$

with $\Delta d = vT_c$, we obtain velocity:

$$v = \frac{\lambda\Delta\phi}{4\pi T_c} \tag{3.11}$$

Regarding equation 3.11, we note that it depends on the phase difference. For this reason, if $|\Delta\phi| < \pi$ there will be ambiguity in the measurement due to the phase periodicity 2π. Thus, the maximum target velocity that the radar can manage is:

$$v_{max} = \frac{\lambda}{4T_c} \tag{3.12}$$

The velocity resolution, defined as the minimum velocity difference perceptible of two objects equidistant to radar, can be determined transmitting N equi-spaced chirps called *frame* and it is defined as:

$$v_{res} = \frac{\lambda}{2T_s} \tag{3.13}$$

where $T_s = NT_c$ is frame time. The higher is the frame time, the higher is the velocity resolution.
The radar can also estimate the angle of arrival (AoA) that is the angle that the received signal generates with the RX antenna. To estimate it, at least two RX antennas are required. A possible solution is shown in Fig. 3.6.

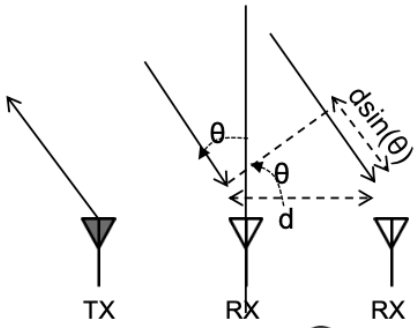


Figure 3.6: AoA estimation with one target [27]

We leverage the phase variation 3.14 at two receiving antennas due to differential distance between each antenna and target:

$$\Delta\phi = \frac{2\pi\Delta d}{\lambda} \tag{3.14}$$

with $\Delta d = d\sin(\theta)$. From 3.14, we can estimate the angle of arrival (AoA) of the object:

$$\theta = \sin^{-1}\left(\frac{\lambda\omega}{2\pi d}\right) \tag{3.15}$$

Angle resolution is defined in [27]:

$$\theta_{res} = \frac{\lambda}{Nd\cos(\theta)} \tag{3.16}$$

The higher is the number of antennas available, the higher will be the achievable resolution.

3.2 The Texas Instruments Radar IWR1443

In this section, we will illustrate the main features of the mmWave TI radar used for measurement campaigns with the goal to exploit the data obtained for the performance assessment of the SLAM algorithm developed in this work.
The board IWR1443 Booster Pack is shown in Fig. 3.7. It is composed of IWR1443 single-chip that is an ideal sensing solution for low power, self-monitored, ultra-accurate radar systems in the industrial field [28].
More in details, IWR1443 Booster pack contains mmWave radar transceiver IWR1443, which is based on the FMCW technology working in a frequency

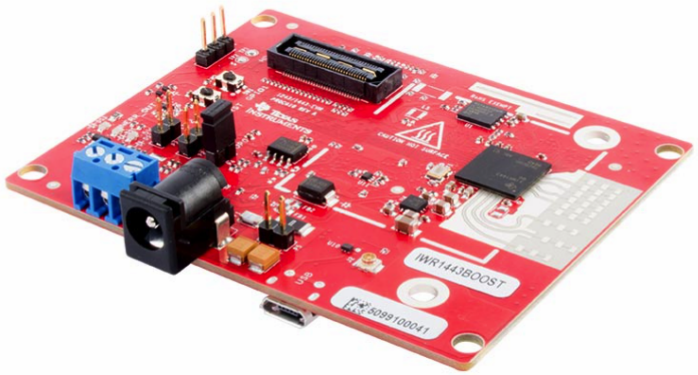


Figure 3.7: The mmWave sensing solution - IWR1443 Booster pack

range from 76 to 81 GHz with 4 GHz continuous bandwidth. In addition, the device is equipped also with 4 receiving antennas and 3 transmitting antennas mounted on PCB as the Radar front-end board with built-in PLL and A/D converters. Thanks to the time division MIMO technology exploited by the radar, it allows to consider 1 TX and an array with 12 RX virtual antennas. Antenna on PCB has an Azimuth FoV which goes from $-60°$ to $60°$ with $15°$ of resolution. However, being 3D radar, it has also $30°$ field of views in an elevation plane with a resolution of $57°$.
This device also includes a DSP subsystem for the signal processing and an ARM R4F-based radio control system for front-end configuration, control and calibration.
To configure mmWave radar and connect it with PC, we use the web application *mmWave Demo Visualizer* [29] of the Texas Instrument. Through this application we can change, in real-time, the configuration of radar parameters and then visualize and record output data on the PC. Furthermore, it is possible to configure the radar with 2 TX and 4 RX antennas or with 3 TX and 4 RX antennas. The data will be processed and used as input for the SLAM algorithm as we will explain in the following chapters.

3.3 300 GHz Measurement Set-Up

In this section, we describe a measurement technology that exploits signals at 300 GHz. Measurements have taken place at CEA-Leti, Grenoble (France), within a collaboration with the University of Bologna in the European project PRIMELOC.
The system is configurated as shown in Fig. 3.8. This measurement system is composed of one transmitting horn antenna and one receiving horn antenna.

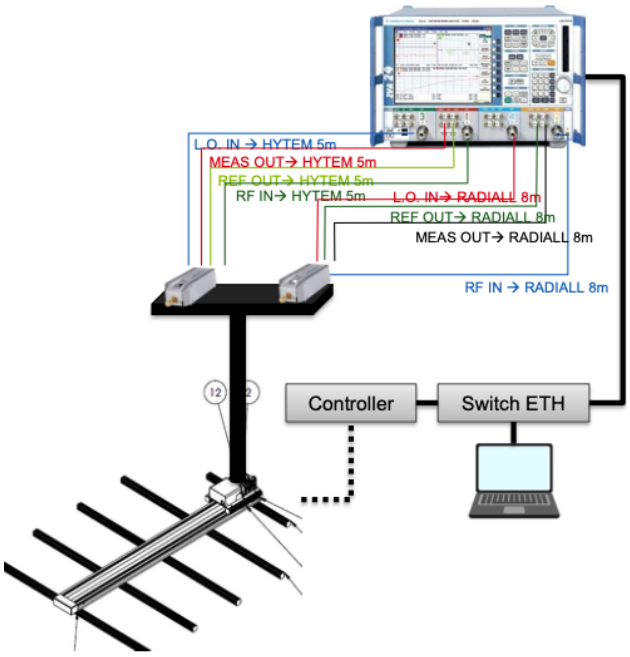


Figure 3.8: CEA-Leti measurement system at 300GHz

Here, the measurements are collected in the frequency domain through a sweep possible thanks to a Vector Network Analyzer (VNA). The purpose is to emulate the behaviour of the quasi-monostatic radar by collecting the data from measurements and subsequently processing them by FFT to characterize the impulse response of the backscattered channel.
The measurement campaigns carried out at CEA-Leti will be described in more details in the next chapter.

Chapter 4

Radio SLAM with Radar Signals: Proposed Algorithms

In this chapter, we will describe the proposed radar-based SLAM algorithms that exploit radar signals to estimate the agent's trajectory and map the surrounding environment. Thanks to above mentioned FMCW radar, raw data can be extracted and processed to obtain angle-range matrix called *radar images* from the measurements acquired during campaigns which can be used as input of algorithms.
The proposed algorithms will be also tested with data sets of measurement obtained with radar at 300 GHz. In this case, the data collected through measurement campaigns provides directly the *radar images*.

4.1 Radar Images from Raw Data

The raw data processing described in this section is summarized by the block diagram in Fig. 4.1.

The TI radar provides raw data measurements as matrices containing the complex baseband samples of the received signal. In particular, for each frame the data format is a complex matrix where the number of columns corresponds to the number of virtual antennas and the number of rows corresponds to the FFT order (N_{FFT}) made by radar in its internal data processing. Each row index is referred to the reflected signal's time of arrival, so the distance from the obstacle to the radar obtained with equation (3.7). We will indicate each element of the matrix with a_{rn} where $r = 1, 2, ..., N_{FFT}$ and $n = 0, 1, ..., N_a - 1$, with N_a the number of virtual antennas.
Before going on, we point out that in this work we consider a 2D case without the Doppler effect but it is possible to create a matrix also with doppler

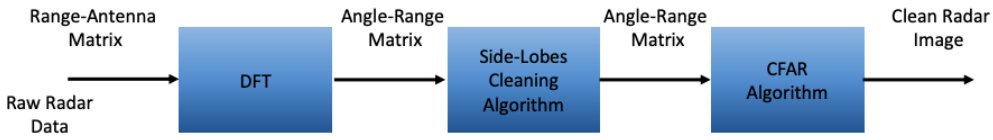


Figure 4.1: Radar pre-processing chain: raw radar measurements are translated into an Angle-Range Matrix thorugh a DFT operation, and, then, a CLEAN algorithm (two step approach) provides the final clean Radar Image.

information.
Let θ and ϕ be the azimuth and elevation angle of radar respectively. So, we consider $\phi = 0$ and we divide θ into M different portions which represent steering angles in the azimuth plane.
The range-antenna matrix is processed using the Discrete Fourier Transform (DFT) in the antenna domain to extract the angular information and to obtain the angle-range matrix. Specifically, the DFT processing is:

$$e_{mr} = \sum_{n=0}^{N_a-1} a_{rn} e^{-j\pi n \sin\theta_m \cos\phi} \tag{4.1}$$

where $r = 1, 2..., N_{FFT}$ with N_{FFT} order of FFT internally made by the radar, $\theta_m = -\frac{\pi}{2} + \frac{(m-1)\pi}{M}$ is the angle domain with $m = 1, ..., M$ and N_a is the number of virtual antennas.
In other words, a radar image represents the environment that lies in the radar's FoV in the azimuth plane. It is obtained from the reflected signals from the surrounding environment returning to the different radar virtual antennas with different angles and intensities.
In particular, the radar image is a $M \times N_{FFT}$ angle-range matrix in which the columns are referred to range samples, with step d_r that is the range resolution of the radar, while the rows are referred to the azimuth angles from $-\pi/2$ to $\pi/2$ with step $d_\theta = 1$ degree i.e., $M = 180$. If the environment presents an obstacle at a given angle-range position, the corresponding cell in the matrix will contain values significantly different from zero.
Due to side lobes in the antenna radiation pattern and multipath of the backscattered signal, the angle-range matrix may contain noise and "ghost" object contributions. They could be a limit for the SLAM algorithm accuracy.

Moreover, due to the wide main lobe of the antenna, objects close to those steered by the radar could be considered. For these reasons, in order to obtain a "clean" and more accurate radar image, we post-process the matrix with cleaning algorithms with which we try to limit these contributions through the setting of suitable thresholds between 0 and 1:

- *Side lobes threshold*: For each angle-range matrix column, we find the cell containing maximum energy value and we discard all the contributions coming from the same range that are likely due to side lobes below the product between this value and the threshold. Fig. 4.2 shows the cleaning process where the maximum M for the rth column is found then only the energy contributions higher than $M \cdot$ *Side lobes threshold* are selected (for example in Fig. 4.2 the red values).

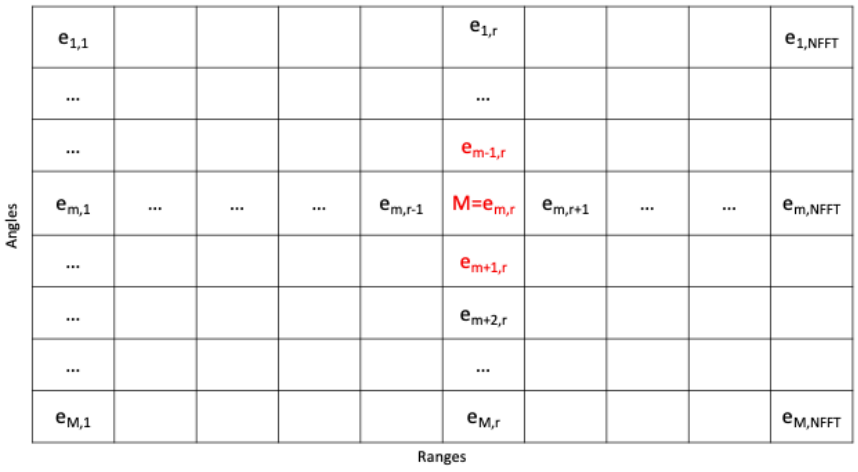
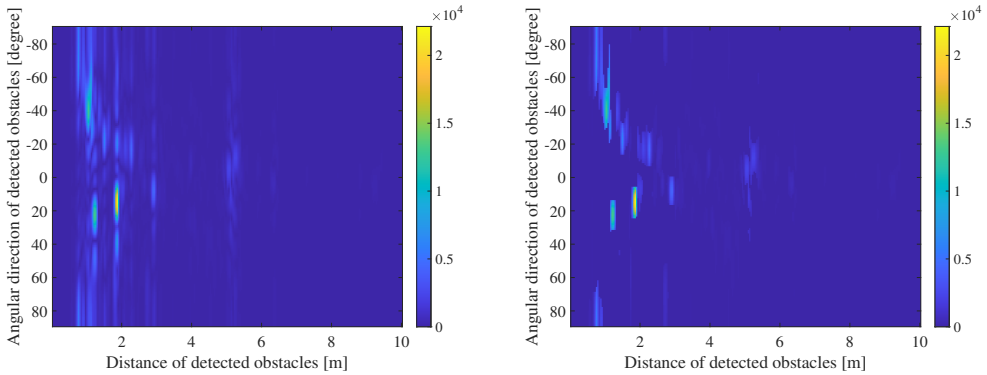


Figure 4.2: Example of the cleaning process

- *Noise threshold*: Considering the entire angle-range matrix, we find the cell the maximum energy value and we discard all energy contributions due to noise which are lower than the product between it and the threshold.

In Fig. 4.3 we show an example of radar image before (Fig. 4.3(a)) and after (Fig. 4.3(b)) the application of cleaning algorithms Here, the radar is placed in $(\rho, \theta) = (0, 0)$ and all contributions below the thresholds have been discarded.



(a) Radar Image (b) Clean Radar Image

Figure 4.3: Raw radar image (left) and clean radar image (right) obtained with *Side lobes threshold* = 0.5 and *Noise threshold* = 0.001

4.2 Scan-to-Scan SLAM Algorithm

In this section, we try to estimate the radar trajectory and map with the Matlab functions, described in section 2.6, provided by the Navigation Toolbox [25] that are born for the lidar's applications. The proposed iterative algorithm is summarized in Fig. 4.4.

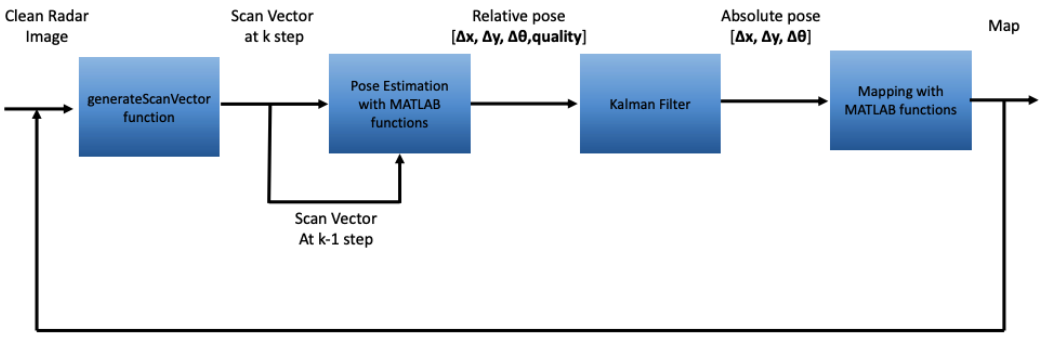


Figure 4.4: Scan-to-scan SLAM algorithm: block diagram

For this purpose, for a given radar measurement, we extract the corresponding radar image as described in section 4.1 and we generate a 1-by-M scan vector to emulate a lidar measurement. First of all, we set a threshold and for each angle i.e., ith row of the range-angle matrix, we take the first element of the radar image which overcomes $s = scanThreshold \cdot e_{max}$ where e_{max} is the maximum energy value of the considered row, and $scanThreshold \in [0, 1]$

is a suitable threshold. Each element of the scan vector is identified by r_m that is the range sample corresponding to the column's index of the identified element. In case all the elements of the row are zero, r_m is set to zero which means no obstacle detected. The scan vector has the form:

$$scan = [r_1, r_2, ..., r_m, ..., r_M] \tag{4.2}$$

So, we have a range vector in which the mth elements corresponds to the angle θ_m.
Now, we generate a *lidarScan* object which can be used with the scan matching and mapping functions presented in sections 2.6.1 and 2.6.2. The *lidarScan* object is built starting from range-angle pairs of targets detected by the radar. An example of a scan vector is shown in Fig. 4.5.

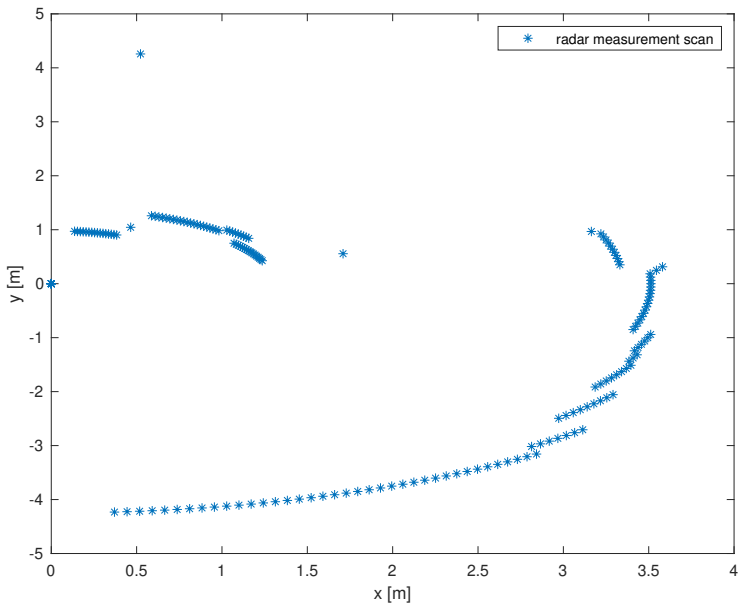


Figure 4.5: An example of radar scan vector plotted in Cartesian coordinates.

The key idea is to find the relative pose $[\Delta x, \Delta y, \Delta\theta]$ between consecutive radar measurements with a scan-to-scan matching strategy at each iteration and successively to estimate the absolute pose and finally to update the map with the *insertRay* function described in section 2.6.2. In addition to the relative pose, we take into account also the quality of the estimated pose as $quality = score/N$ where *score* is a property of Matlab matching functions and N is the number of points in the current scan measurement.

The relative pose resulting from the matching is considered only if the scan at kth step does not differ much from the scan at $(k-1)$th step. In this case, we consider the last estimated pose as an outlier and we force it to be $[\Delta x, \Delta y, \Delta\theta] = (0, 0, 0)$.
Regarding the absolute pose, it is computed in two different ways: first by adding step by step the estimated relative poses and then also with the Kalman filter strategy to improve the estimate. For the latter, the state of the system, $\mathbf{x}$, is represented by the two components of absolute position, by the two components of speed, by orientation and by the speed of rotation of the radar. Therefore, the state is identified by a vector:

$$\mathbf{x} = [x, y, \dot{x}, \dot{y}, \theta, \dot{\theta}]^T \tag{4.3}$$

We have considered a linear motion model of the radar at step k with the constant speed with additive Gaussian noise $\mathbf{w}_k \backsim \mathcal{N}(0, \mathbf{Q})$ with covariance matrix $\mathbf{Q}$:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_k \tag{4.4}$$

where the deterministic terms of 4.4 are given by:

$$\begin{aligned} x_{k+1} &= x_k + \dot{x}_k \cdot \Delta t \\ y_{k+1} &= y_k + \dot{y}_k \cdot \Delta t \\ \dot{x}_{k+1} &= \dot{x}_k \\ \dot{y}_{k+1} &= \dot{y}_k \\ \theta_{k+1} &= \theta_k + \dot{\theta}_k \cdot \Delta t \\ \dot{\theta}_{k+1} &= \dot{\theta}_k \end{aligned} \tag{4.5}$$

with Δt being the time between two consecutive estimates. The state transition matrix is constant over time then $\mathbf{F}_k \equiv \mathbf{F}$. Also $\mathbf{Q}$ is constant over time:

$$\mathbf{Q} = \begin{bmatrix} w_0\frac{\Delta t^3}{3} & 0 & w_0\frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & w_0\frac{\Delta t^3}{3} & 0 & w_0\frac{\Delta t^2}{2} & 0 & 0 \\ w_0\frac{\Delta t^2}{2} & 0 & w_0\Delta t & 0 & 0 & 0 \\ 0 & w_0\frac{\Delta t^2}{2} & 0 & w_0\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{t0}\frac{\Delta t^3}{3} & w_{t0}\frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & w_{t0}\frac{\Delta t^2}{2} & w_{t0}\Delta t \end{bmatrix}$$

where $w_0 = 0.01^2$ and $w_{t0} = 0.01^2$ are empirical values that depend on the expected mobility of the radar.

Let $\mathbf{z}$ be the observation vector:

$$\mathbf{z} = [\dot{x}, \dot{y}, \dot{\theta}]^T \tag{4.6}$$

The observation model with additive Gaussian noise $\mathbf{n}_k \backsim \mathcal{N}(0, \mathbf{R}_k)$ with covariance matrix $\mathbf{R}_k$ is identified by :

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k \tag{4.7}$$

where the observation matrix is constant over time then we can neglect the temporal index ($\mathbf{H}_k \equiv \mathbf{H}$):

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Given a prior information about the state (i.e., $\mathbf{m}_0 = \mathbf{0}$ and $\mathbf{P}_0 = \mathbf{I}$), the algorithms alternate between the following two steps:

- ***Prediction step*** :

$$\begin{aligned} \mathbf{x}_{k|k-1} &= \mathbf{F} \cdot \mathbf{x}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F} \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}^T + \mathbf{Q} \end{aligned} \tag{4.8}$$

  where $\mathbf{P}$ is the state covariance matrix and the notation $\mathbf{x}_{k|k-1}$ indicates the prediction of the state vector $\mathbf{x}_k$ made at time instant $k-1$ by considering only the transition model in 4.4. Such a prediction will be corrected in the correction step once a new measurement is collected at k.

- ***Correction step*** :

$$\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k \cdot \mathbf{y}_k \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \cdot \mathbf{S}_k \cdot \mathbf{K}_k^T \end{aligned} \tag{4.9}$$

  with

$$\begin{aligned} \mathbf{y}_k &= \tilde{\mathbf{z}}_k - \mathbf{H} \cdot \mathbf{x}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H} \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \cdot \mathbf{H}^T \cdot \mathbf{S}_k^{-1} \end{aligned} \tag{4.10}$$

  where $\mathbf{y}$ is the innovation term, $\mathbf{S}$ is the innovation covariance matrix, $\mathbf{K}$ is Kalman gain and $\mathbf{R}$ is the covariance matrix of the measurement noise depending on the *quality* of estimated relative pose between two

consecutive measurements:

$$\mathbf{R}_k = \begin{bmatrix} (\frac{\sigma_r}{quality(k)})^2 & 0 & 0 \\ 0 & (\frac{\sigma_r}{quality(k)})^2 & 0 \\ 0 & 0 & (\frac{\sigma_\theta}{quality(k)})^2 \end{bmatrix}$$

where $\sigma_r = d_r/5$ and $\sigma_\theta = d_\theta/5$.
The term $\tilde{\mathbf{z}}$, represents the relative translations transformed according to the orientation estimate:

$$\tilde{\mathbf{z}} = [z_x, z_y, z_\theta] \tag{4.11}$$

where

$$\begin{aligned} z_x &= \Delta x \cdot \cos(\theta) + \Delta y \cdot \sin(-\theta) \\ z_y &= \Delta y \cdot \cos(\theta) - \Delta x \cdot \sin(-\theta) \\ z_\theta &= \theta \end{aligned} \tag{4.12}$$

Here, $\Delta x, \Delta y, \Delta\theta$ are the estimated parameters of relative pose between k and $k-1$ step.

After the absolute pose has been determined, the map is updated with the last measurement. The environment is represented by a grid whose cell contains the probability of the cell being occupied and it is treated with the Matlab functions in 2.6.2.
We point out that it is an incremental method and it computes the relative pose by matching two consecutive images. For these reasons, this class of algorithms may present an accumulation of errors which propagating may cause a drift in the estimated trajectory.

4.3 Consecutive Images Registration SLAM Algorithm

This approach differs from the previous for the strategy to recover the relative pose between two consecutive radar measurements.
Here, the strategy is to estimate the translation and rotation parameters between two consecutive radar images directly with Fourier-Mellin transform described in section 2.5.
For this purpose, we present the block diagram in Fig. 4.6 which describes the registration algorithms.
We consider two consecutive polar radar images, I_k at the current frame and

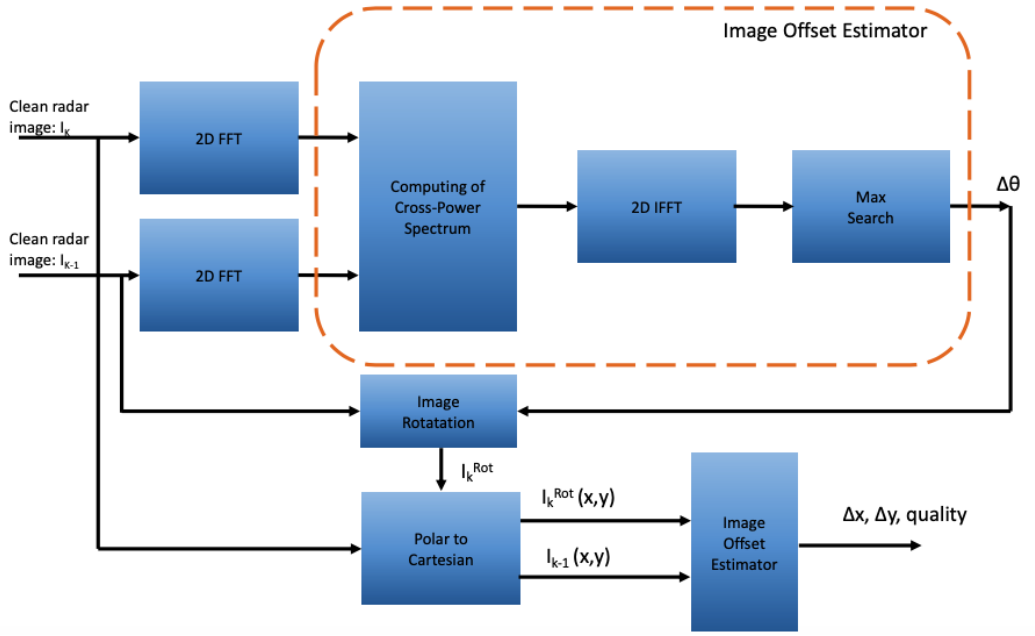


Figure 4.6: Relative pose estimation between two consecutive radar images

I_{k-1} at the previous frame. Here, the radar images are of the same size.
First of all, we retrieve the rotation $\Delta\theta$ of I_k respect to I_{k-1}. As described in section 2.5, we apply the 2D-FFT to both current and previous images and we compute the cross-power spectrum C using equation (2.17). After that, we process C with an ideal 2D low pass filter with 0.4 empirical bandwidth. After that, we apply the 2D-IFFT to the absolute value of cross-power spectrum and we find the maximum corresponding to the rotation parameter $\Delta\theta$. Now, the translation parameters $(\Delta x, \Delta y)$ must be found. For this purpose, both the rotated current and previous image must be converted into an image in cartesian coordinates and the parameters are computed with the same steps used for the rotational parameter. This time the low pass filter empirical bandwidth is set to 0.1. The quality of registration is also taken into account considering the maximum value assumed by cross-correlation peak in correspondence with the translation parameters.
After we have computed the relative pose, to find the absolute pose and successively to update the map, the algorithm follows the steps described in section 4.2 according to the block diagram in Fig. 4.7.

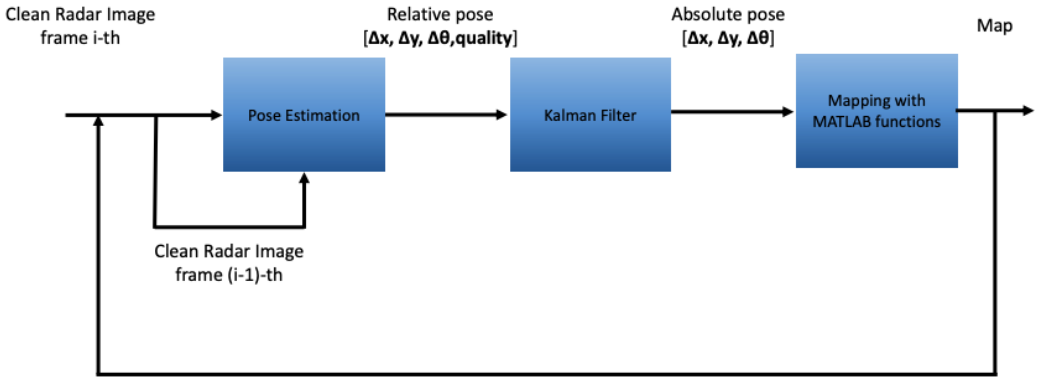


Figure 4.7: Overview of the SLAM algorithm using radar images

4.4 Scan-to-Map SLAM Algorithm: First Version

We now describe a new strategy to solve the SLAM problem partially avoiding the problem of trajectory drifting. The key idea is to exploit the map which is updated at each iteration to estimate the radar relative pose between two consecutive measurements. Here, the map is a binary occupancy grid map. The expected advantage of this strategy is the mitigation of the drift in case of outlier images, and the possibility to get an absolute position estimation. The strategy is shown in Fig. 4.8.

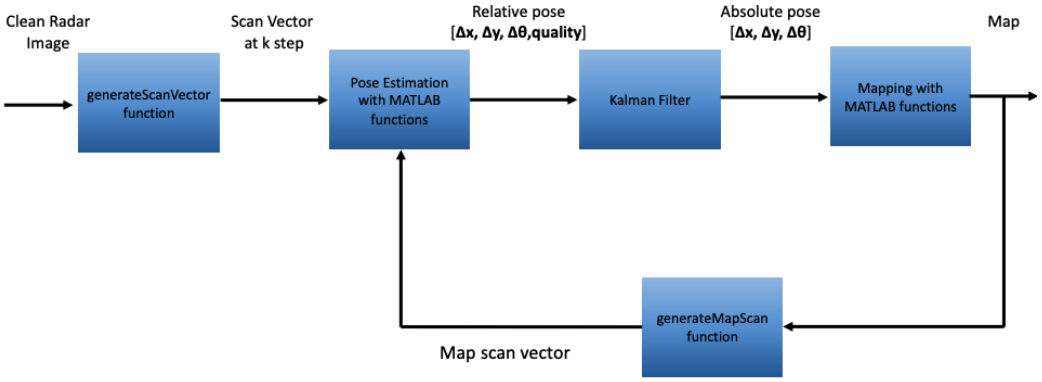


Figure 4.8: Scan-to-map SLAM algorithm: block diagram

First, we set an arbitrary threshold and we refer to it with *frameThreshold*.

This threshold sets the number of measurements after which the algorithm starts to match the current scan with the map built up to that point at each iteration. For this reason, the different results can be obtained with different values of this threshold even if the data set of measurement is the same. As long as the number of processed frames is less than the threshold value, the algorithm behaviour is the same as section 4.2 and the matching is made between consecutive scan measurement. To compare the map with the current scan using the Matlab functions, it is necessary to retrieve the scan vector relative to the map. For this purpose, we have implemented an ad-hoc function described in the next section 4.4.1.
Once the map scan vector is generated, we use the Matlab function for 2D scan matching to estimate the relative pose of the radar. Successively, the absolute pose is found as in 4.2 and the algorithm proceeds as described in the previous section.

4.4.1 Map Scan Vector Generation

The purpose of this function is to obtain a scan vector starting from a partial map of the global grid map by accounting for the current estimated position and orientation of the radar. The function takes in input the binary occupancy map and the absolute pose of radar. The full process is described by the block diagram in Fig. 4.9.
Let $\{\mathbf{h}_i\}_{i=1,\dots,n}$ be all the points set to 1 by the occupancy map algorithm at the kth step. Each of these points has spatial coordinates $\mathbf{h}_i = (x_i, y_i)$.
The first time that this function is called, the absolute radar pose $\mathbf{p}^{\mathrm{A}} = [x^{\mathrm{A}}, y^{\mathrm{A}}, \theta^{\mathrm{A}}]$ at the previous step $k-1$ is known so we shift all the targets coordinate by the absolute radar position $[x^{\mathrm{A}}, y^{\mathrm{A}}]$ as follows:

$$\tilde{\mathbf{h}}_i = \mathbf{h}_i - [x^{\mathrm{A}}, y^{\mathrm{A}}] \tag{4.13}$$

In this way, the previous radar position now corresponds to the origin $[x^{\mathrm{A}}, y^{\mathrm{A}}] = (0, 0)$ of the map.
For each translated point, we obtain the polar coordinates (ρ, θ) :

$$\begin{aligned} \rho_i &= \sqrt{\tilde{x}_i^2 + \tilde{y}_i^2} \\ \theta_i &= \tan^{-1}\left(\frac{\tilde{y}_i^2}{\tilde{x}_i^2}\right) \end{aligned} \tag{4.14}$$

Now, we have two 1-by-n vectors, one for angles measured counter-clockwise around the positive z-axis, and the second for ranges. We will refer to them as *Angles vector* and *Ranges vector* respectively. Successively, we consider

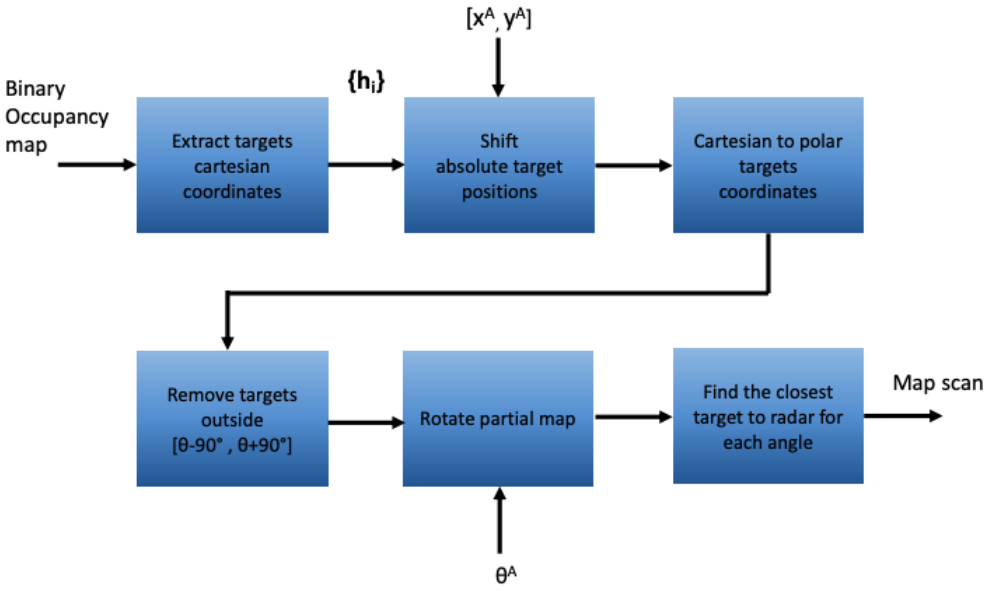
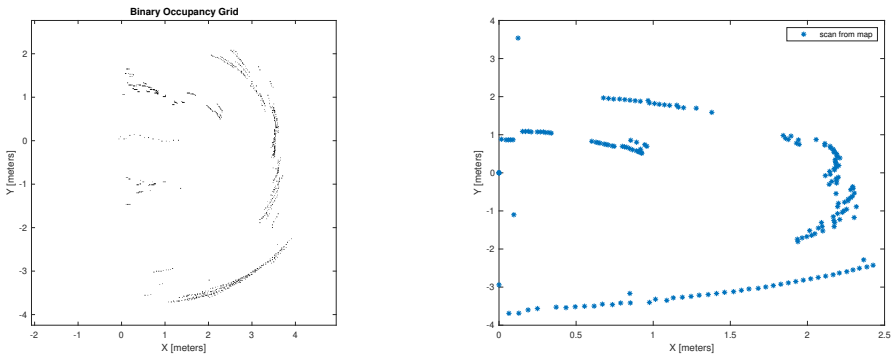


Figure 4.9: Scan vector from the map: block diagram

the radar bearing angle θ^{A} at previous step $k-1$ and we rotate the map by this angle. In polar coordinates, the rotation is possible by subtracting the quantity θ^{A} from each element θ_i of *Angles vector* as shown by the following equation:

$$\tilde{\theta}_i = \theta_i - \theta^{\mathrm{A}} \tag{4.15}$$

After we have roto-traslated the map, we neglect all the elements of the *Angles vector* which are outside the angular range $[\theta^{\mathrm{A}} - 90°, \theta^{\mathrm{A}} + 90°]$ and consequently their associated elements in the *Ranges vector* to obtain a partial map of the global map. This partial map corresponds to the portion of the global map seen by the radar at that moment.
Now, we need to get a 1-by-M, with $M = 180$, scan vector from the map with the same number of elements of the current scan. For this purpose, we approximate all the elements of the *Angles vector* to the smallest integer then for each angle from $-90°$ to $90°$, we consider all the associated points in the *Ranges vector* and we account for the closest point to radar obtaining a scan vector with the same length as the current scan.
In Fig. 4.10 we show an example of *lidarScan* generated starting from the binary occupancy map as above described:



(a) (b)

Figure 4.10: *lidarScan* from the occupancy binary map mapped in Cartesian coordinates

4.5 Scan-to-Map SLAM Algorithm: Second Version

In this section, we present a different approach for the scan-to-map matching strategy described in section 4.4. The only difference is in the generation of the *lidarScan* objects to be fed to the matching functions of Matlab in order to compute the relative pose.
To account for more information, instead of processing a scan vector using a single range value, we consider multiple ranges for each given angle. This is possible by considering all the cells with energy comprised within a specific threshold. We will refer to it as *scanThreshold*.
We show in Fig. 4.11 an example of *lidarScan* object generated starting from a radar image in which, for a given angle, we take into account all energy contributions between $max \cdot scanThreshold < e < e_{max}$ where e_{max} is the maximum value of energy for that given angle. Following a similar approach also for the map, we considered more points instead of the single closest point to the radar as in section 4.4. In particular, given an updated map at kth time step, we roto-translate it as described in section 4.4.1 then we generate a *lidarScan* object starting from all the points that, for a given angle, lie in the space between the closest point to the radar and a specific threshold set by the user. We will refer it to as *mapThreshold*.

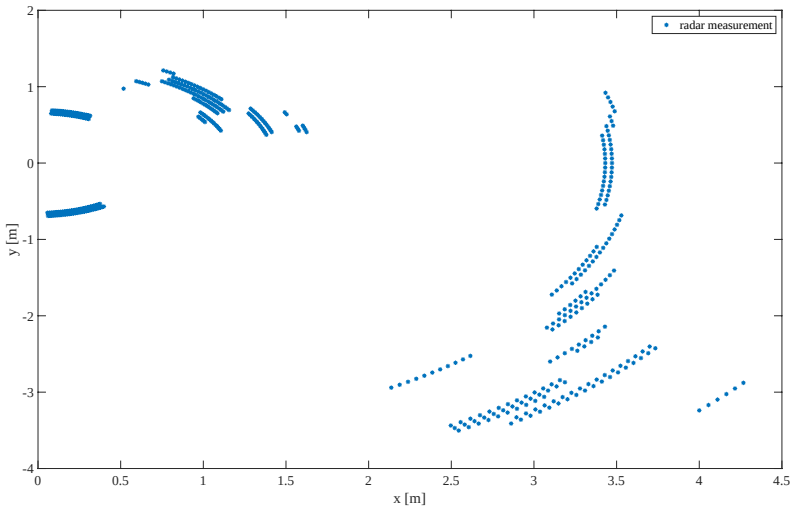
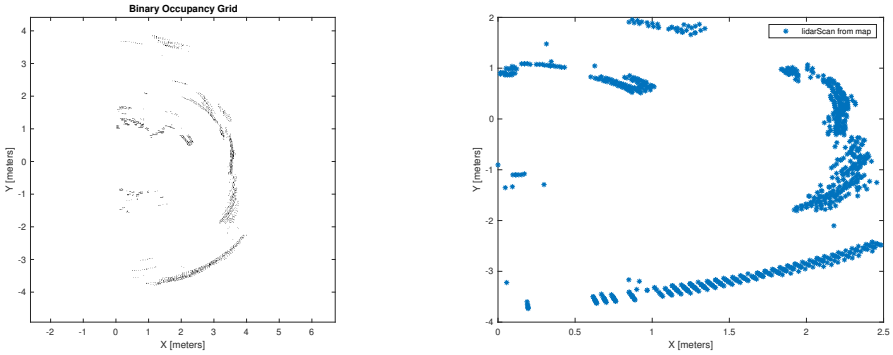


Figure 4.11: An example of Radar measurement with *scanThreshold* = 0.7

In Fig. 4.12, we show an example of *lidarScan* obtained from the updated binary map.



(a) (b)

Figure 4.12: *lidarScan* from the occupancy binary map mapped in Cartesian coordinates with *mapThreshold* = 0.2 m

With this approach, we do not emulate the lidar's measurements to exploit the Matlab pose estimation function but we try to estimate the relative pose between the current measurement and the current binary map by considering a cloud of points.

Chapter 5

Measurement Campaigns and Results

The goal of this chapter is to describe the different measurement campaigns made with different radar technologies and successively, to show the results obtained by the proposed SLAM algorithms. In addition to the results achieved with the mmWave radar at 77 GHz, we will also show the results obtained with the 300 GHz radar technology available at CEA-Leti.

5.1 Measurement at 77 GHz with TI mmWave Radar

For measurement campaigns at 77 GHz, the TI mmWaves radar described in section 3.2 has been used. It has been fixed over wheeled support and it has been driven by the user along different trajectories (Fig. 5.1). In particular, in this section, we will describe different measurement campaigns made in different indoor scenarios in order to assess the validity of the proposed SLAM algorithms and to understand which is the most accurate.

For each scenario, first, we will show the results obtained through the SLAM algorithm described in section 4.2, successively, we will illustrate the results with the image registration method of section 4.3 and, finally, with the enhanced algorithm introduced in section 4.4.

Figure 5.1: Set-up of 77GHz measurement campaigns with TI Radar

For all the campaigns, the radar has been configured with 2 TX antennas and 4 RX antennas and the following parameters have been set through the mmWave Demo Visualizer:

- *Frame Rate* (fps): 4
- *Range Resolution* (m): 0.044
- *Maximum Unambiguous Range* (m): 9.01
- *Maximum Radial Velocity* (m/s): 1
- *Radial Velocity Resolution* (m/s): 0.13

For the simulations, a parameter called *skipFrame* allows to consider only multiple radar frames. For all scenarios, we set *skipFrame*=4 which considers one measurement per second.
Regarding the properties of the signals emitted by the radar, other important parameters are listed in the following:

- *Start Frequency* (GHz): 77
- *Slope* (MHz/μs): 70
- *Samples for chirp*: 256
- *Chirp per Frame*: 32

- *Sampling Rate* (Msps): 4.88

Samples for chirp is the N_{FFT} parameter mentioned in section 4.1 that represents the order of the FFT made internally by the radar.
The results will show two estimated trajectories obtained through the absolute pose estimation as described in section 4.2. The red trajectory is obtained by the sum of relative poses estimated at each iteration while the blue trajectory with the Kalman filter. The map was updated starting from the absolute poses estimated with the Kalman filter. For this purpose, the map properties set in Matlab are:

- *mapResolution* = 60
- *maxRange* = 4

This means that the environment is subdivided in 3600 cells per square meters and range values greater than to *maxrange* are discarded.
We also report the values of the thresholds mentioned for the cleaning algorithms valid for all measurement campaigns if not otherwise specified:

- *Side lobes threshold* = 0.5
- *Noise threshold* = 0.001

The threshold for the scan image generation is set to *scanThreshold* = 0.9; it is valid for both the scan-to-scan and the first version of scan-to-map SLAM algorithms.
Finally, we report two setting thresholds for which if the estimated relative pose is higher than them then the corresponding estimation is considered as outlier:

- *angle outlier threshold* = 45°
- *offset outlier threshold* = 0.3 [m]

For all the scenarios, first, we will show the results obtained with the scan-to-scan SLAM algorithm so we will present the improvement with the scan-to-map strategy introduced to avoid the trajectory drifting problem.

5.1.1 First Scenario

In the first scenario represented in Fig. 5.2, the radar walks along a square trajectory inside a 4×4 empty room.

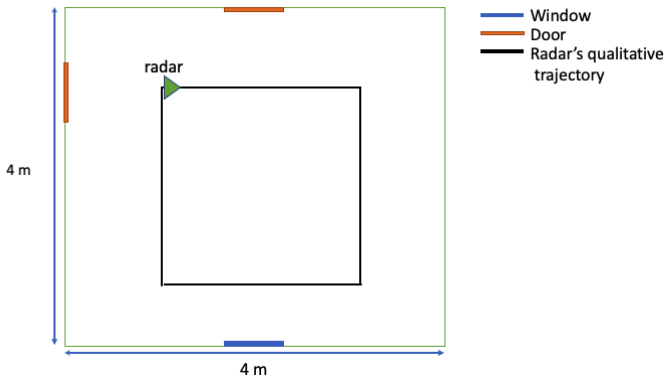


Figure 5.2: Qualitative trajectory of the radar and map for the first scenario

First, we show the results in Fig. 5.3 obtained with the scan-to-scan SLAM algorithm and the *matchScansGrid* function.

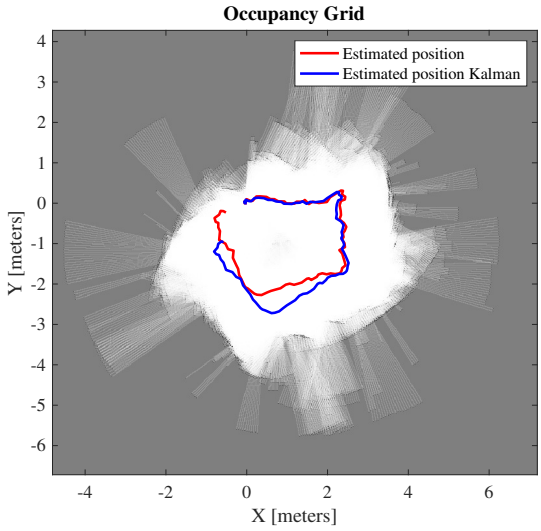


Figure 5.3: Results of the scan-to-scan SLAM algorithm at 77 GHz: 1st scenario

The trajectory has a drift in the final section. In fact, being a square trajectory the final position should be close to the initial position. This drift is due to the accumulated error during the pose estimation.

In Fig. 5.4, we present also the results obtained with the SLAM algorithm based on the registration between consecutive radar images with the Fourier-Mellin approach described in section 4.3:

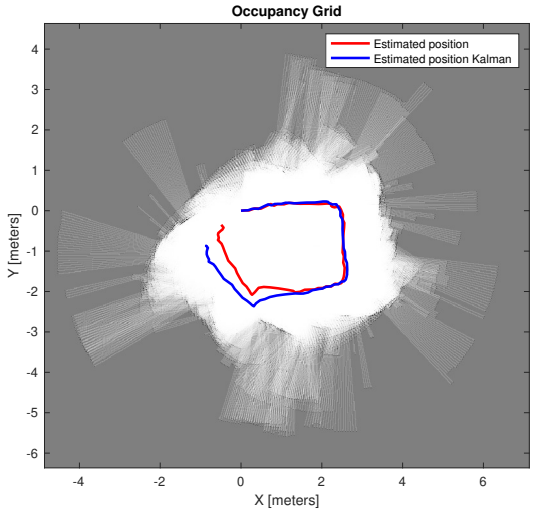
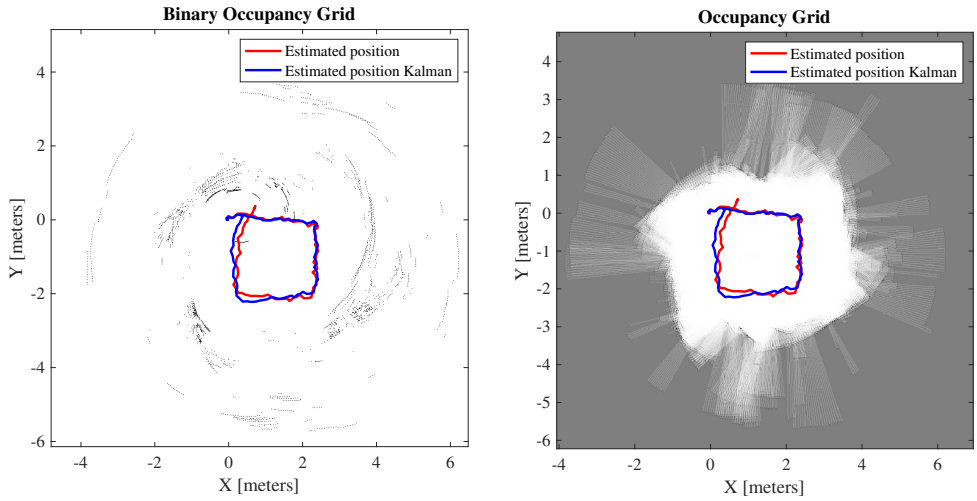


Figure 5.4: Results of Fourier-Mellin SLAM algorithm at 77 GHz: 1st scenario

Initially, the results show that this registration technique leads to a good estimation of the trajectory and consequently of the map. However, being the trajectory estimated incrementally starting from the previous image, this strategy can fall into the same drift issue for longer paths.
Finally, we show an improvement of the estimated trajectory according to the first version of the scan-to-map SLAM algorithm presented in section 4.4. The result is achieved with the *frameThreshold* parameter set to 20. This strategy allows to partially avoid the drift experienced by the estimated trajectory because the current scan is matched with the absolute map. The map is reinforced by previous measurements and it offers a fixed reference point to the current measurement to which is matched.
In this approach, both the binary occupancy map and the occupancy map are shown in Fig. 5.5:



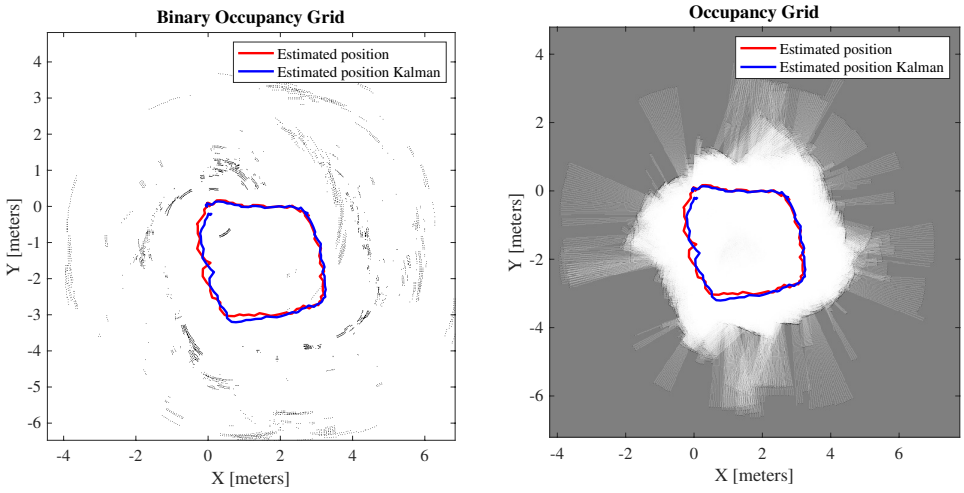
(a) (b)

Figure 5.5: Results of 1st version of the scan-to-map SLAM algorithm at 77 GHz: 1st scenario

As expected, Fig. 5.5 shows an improvement of the trajectory due to the adopted scan-to-map strategy. In particular, the final drift of the trajectory is avoided resulting in an improvement also for the map. Considering both the maps, the binary map in Fig. 5.5(a) has less information than the occupancy grid map in Fig. 5.5(b) and it has some details that are not present in the other map due to the updating policy adopted by the Matlab's algorithms. Successively, the results obtained with the second version of the scan-to-map SLAM algorithm are shows in Fig. 5.6.
Here, the thresholds are :

- $scanThreshold = 0.7$
- $mapThreshold = 0.2$ [m]
- $frameThreshold = 20$

As for the results shown in Fig. 5.5 also here, there is an improvement of both the trajectory and the map.



(a) (b)

Figure 5.6: Results of 2nd version of the scan-to-map SLAM algorithm at 77 GHz: 1st scenario

5.1.2 Second Scenario

In the second scenario, the radar walks through a 1 meters wide corridor for 5 meters then it turns a 90° degree angle and follows a straight path for the other 4 meters (Fig. 5.7).

Figure 5.7: Second scenario at 77 GHz

The qualitative trajectory followed by the radar is shown in Fig. 5.8.

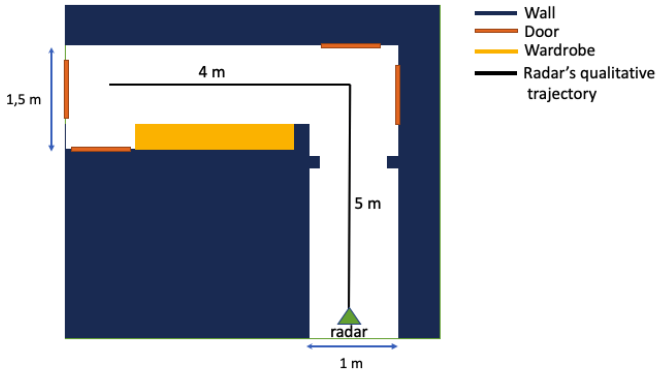


Figure 5.8: Qualitative trajectory of the radar and map for the second scenario

The result with the scan-to-scan matching algorithm with *matchScansGrid* are proposed in Fig. 5.9.

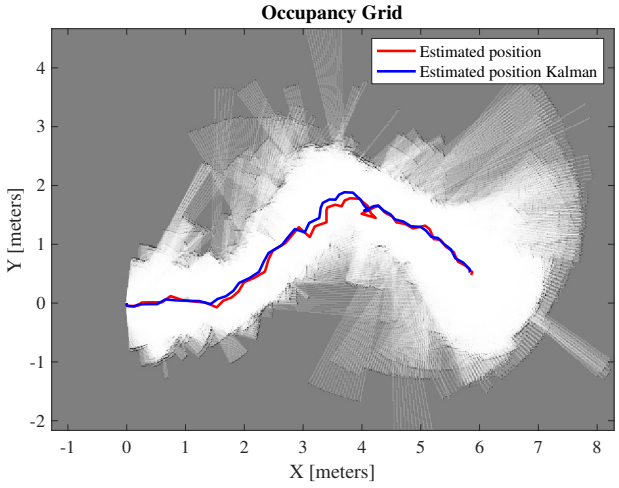


Figure 5.9: Results of the scan-to-scan SLAM algorithm at 77 GHz: 2nd scenario

Here, the trajectory deviates from the straight path after 1.5 meters before turning 90 degrees and follows a straight line for other 4 meters. This behaviour is likely caused by a bad measured image which occurs after 1.5 meters and the estimation error is accumulated over the entire trajectory thus putting in evidence the limits of scan-to-scan approaches. Although the

trajectory deviates from the expected path, the mapping shows the presence of side walls at a distance of 1 meter, identifying the movement through a corridor. Also the trajectory and map with SLAM algorithm based on Fourier-Mellin are shown:

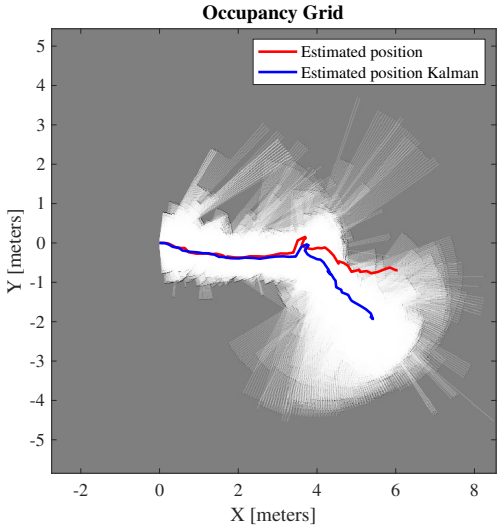
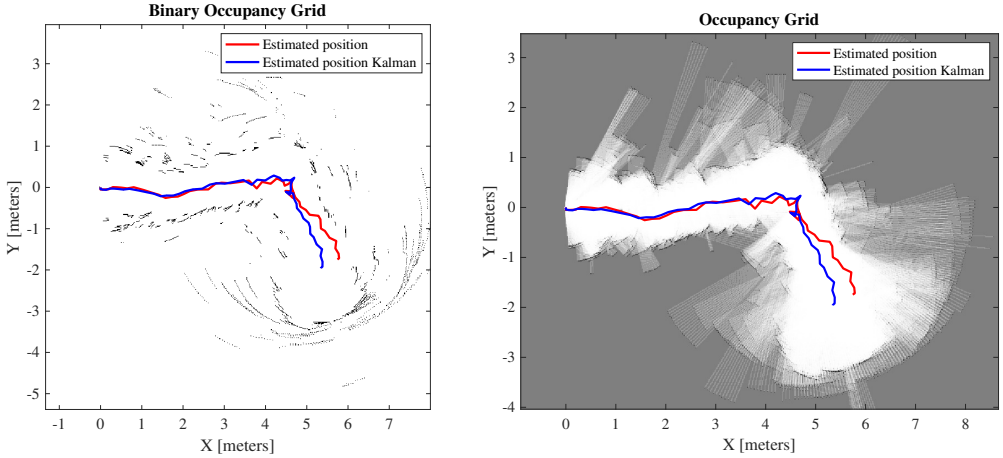


Figure 5.10: Results of Fourier-Mellin SLAM algorithm at 77 GHz: 2nd scenario

Here, the trajectory follows a straight path until the radar makes a turn of 90° angle. After this turn, we expect the radar moving again along a straight line but the trajectory gets a drift that becomes significant bwhen the relative poses are just accumulated and the Kalman filter is not used. Comparing the trajectories in Fig. 5.9 and in the Fig. 5.10, it can be seen that in the first section Fourier-Mellin strategy provides more accurate results than the scan-to-scan strategy. In fact, the trajectory is straight. On the other hand, in the second section, the trajectory is less accurate by directly using the radar images for registration causing also less accurate mapping.
The above mentioned drift issue can in part be mitigated using scan-to-map approaches, as depicted in Fig. 5.11.
In particular, the estimated trajectory is more accurate than the previous one and follows the true path. For this scenario, the result has been obtained with a *thresholdFrame* set to 5.

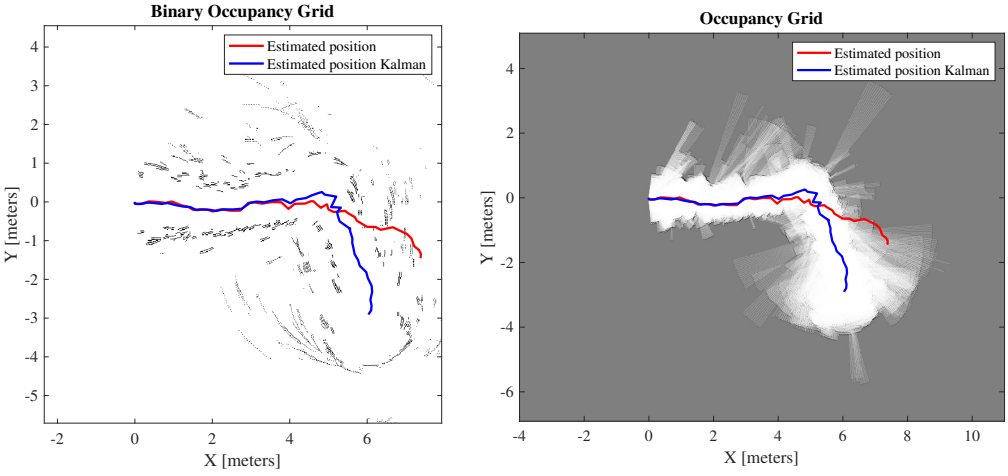


(a) (b)

Figure 5.11: Results of 1st version of the scan-to-map SLAM algorithm at 77 GHz: 2nd scenario

Now, we show the results obtained with the second version of the scan-to-map SLAM algorithm where the *frameThreshold* parameter is still set to 5 and the others are set as follows:

- $scanThreshold = 0.8$
- $mapThreshold = 0.2$ [m]



(a) (b)

Figure 5.12: Results of 2nd version of the scan-to-map SLAM algorithm at 77 GHz: 2nd scenario

Here, the estimation of the trajectory with Kalman is more accurate than the case in which the relative poses are recursively added.
Both the binary occupancy map and the occupancy map clearly show that the radar moved through the above described corridor.

5.1.3 Third Scenario

In the third scenario, the radar follows a 8-like trajectory inside a $10,5 \times 5$ meters rectangular room (Fig. 5.13). During the movement, the radar turns around two highly reflective objects.

Figure 5.13: Third scenario at 77 GHz

The qualitative trajectory followed by the radar is shown in Fig. 5.14.

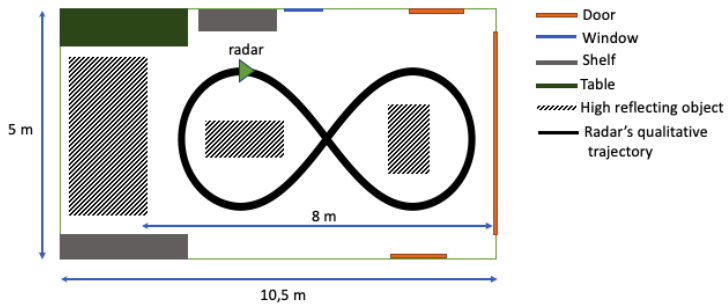


Figure 5.14: Qualitative trajectory of the radar and map for the third scenario

Figure 5.15 shows the result obtained with the scan-to-scan matching strategy with *matchScansGrid* function.

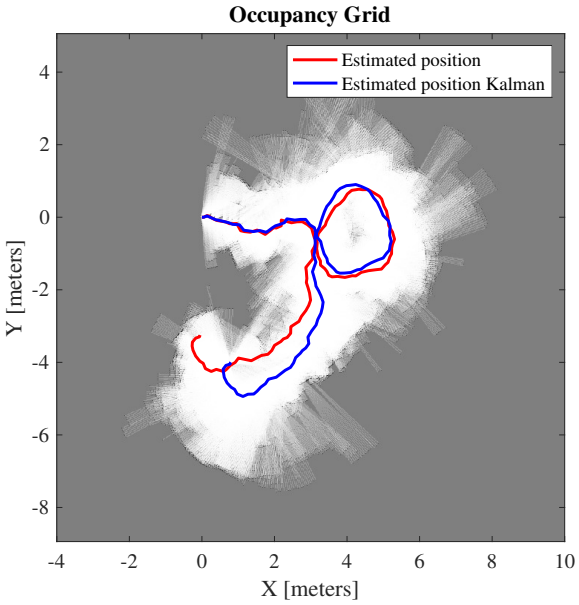


Figure 5.15: Results of the scan-to-scan SLAM algorithm at 77 GHz: 3rd scenario

Due to the complex environment, also for this scenario, some of the measurements are inaccurate leading to a deviation from the real trajectory.
The results obtained using the algorithm based on the Fourier-Mellin transform, presented in section 4.3, are shown in Fig. 5.16.

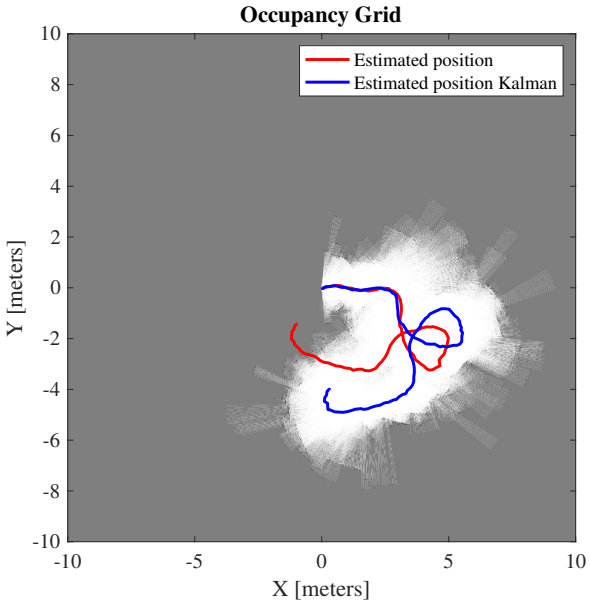
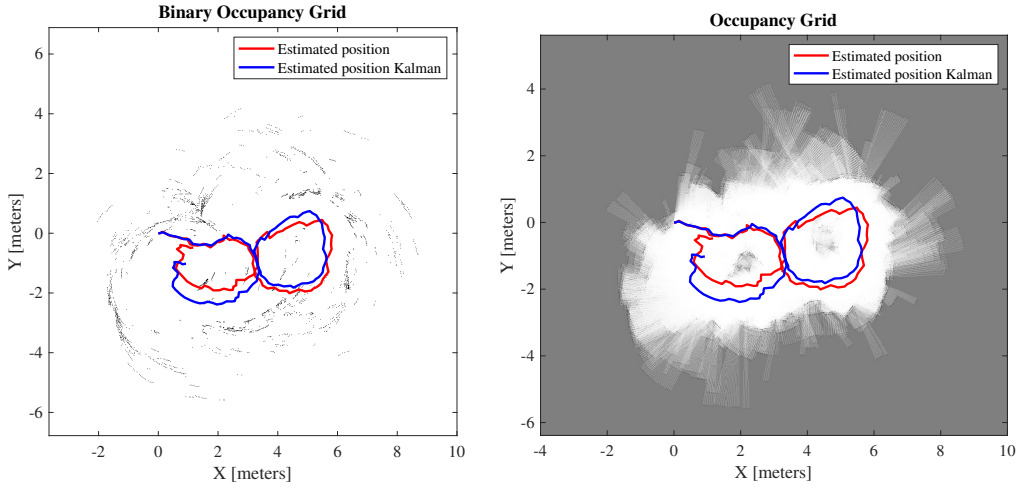


Figure 5.16: Results of Fourier-Mellin SLAM algorithm at 77 GHz: 3st scenario

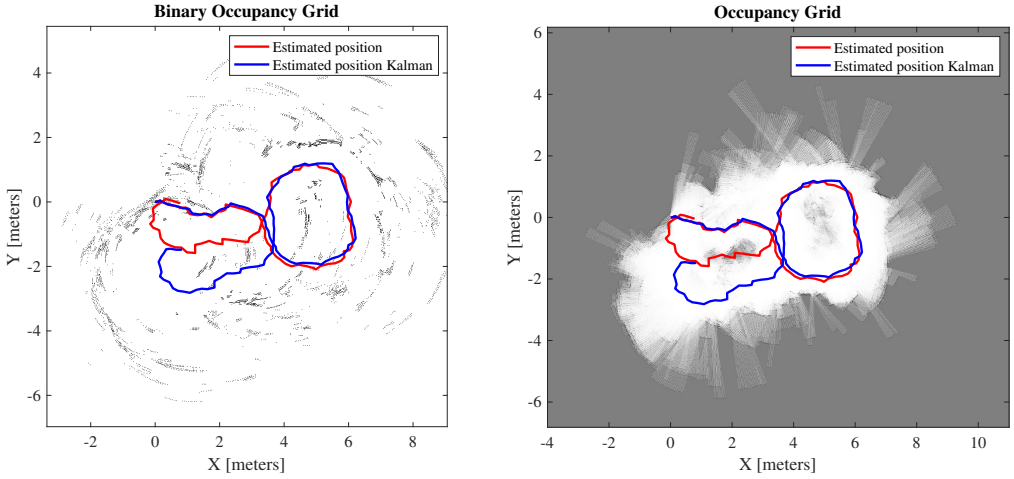
Here, the trajectory does not respect the pre-determined path because the registration of consecutive images is inaccurate. For this reason, the absolute pose estimation with Kalman is affected by errors leading also to poor mapping results.
Fig. 5.17 shows the improvement of the SLAM algorithm based on the scan-to-map strategy. Here, the *frameThreshold* is set to 20. When the radar approaches the already visited places, matching the scan with the map permits to avoid the drift of the trajectory.



(a) (b)

Figure 5.17: Results of 1st version of the scan-to-map SLAM algorithm at 77 GHz: 3rd scenario

Also for the last scenario, we show the results obtained with the second version of the scan-to-map SLAM algorithm reported in Fig. 5.18.



(a) (b)

Figure 5.18: Results of 2nd version of the scan-to-map SLAM algorithm at 77 GHz: 3rd scenario

As in Fig. 5.17, the drift of trajectory estimation has decreased compared to the results in Fig. 5.15 but the radar final position, in this case, does not overlap the initial position.

Regarding the occupancy map for each version of the scan-to-map algorithm, it is more accurate than the binary occupancy map. In particular, the presence of the two highly reflective objects can also be noted. Generally, referring to the scan-to-map strategy, we point out that the results are too sensitive to the set thresholds, in fact, their variations lead to different results. However, unlike the scan-to-scan algorithms, this method permits to avoid the drifting issue because it uses the map built up to that moment and therefore reinforced by the previous measures. This partially mitigates the problem of outlier measurements that often occur at 77GHz.

5.2 Measurement at 300 GHz

For the measurement campaign described in this chapter, a measurement system equipped with two 300 GHz horn antennas has been used, one for TX and one for RX. In particular, the measurement parameters have the following features:

- *Frequency min* (GHz): 235
- *Frequency max* (GHz): 320
- *Frequency step* (MHz): 10
- *Angular step* (°): 10
- *Horn antenna gain* (dBi): 20

The frequency step between each measure of 10 MHz allows to have maximum distance detectable of 15 meters for radar measurement.
Regarding to the map, the properties are:

- *mapResolution* = 40
- *maxRange* = 8

The thresholds set for the cleaning algorithms are:

- *Side lobes threshold* = 0.5
- *Noise threshold* = 0.01

For all the scenarios, regarding results from scan-to-scan and from the first version of the scan-to-map algorithm, the *scanThreshold* is set to 0.9 as in the 77 GHz measurement campaigns. According to the measurement campaigns above described, we set thresholds to consider outliers a deviation from two radar poses as:

- *angle outlier threshold* = 50[°]
- *offset outlier threshold*= 0.5 [m]

For these measurement campaigns all measurements are considered then *skipframe* = 1;

5.2.1 First Scenario

In the first scenario, the data set has been collected moving the platform, where the radar was fixed, straight along to rail placed in the centre of a 10.2×8.6 meters rectangular room in Fig. 5.19. The radar trajectory is of 2 meters forward with intermediate steps of 0.5 meters during which the radar scans the environment with mechanical steering from $-90°$ to $90°$ with a step of $10°$ (Fig. 5.20).

Figure 5.19: Real scenario at CEA-Leti

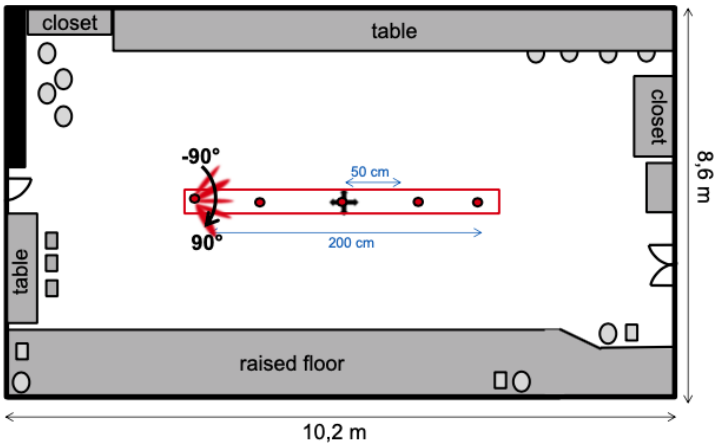


Figure 5.20: Actual trajectory for the first scenario at 300 GHz at CEA-Leti

First, we present the results obtained with the scan-to-scan SLAM algorithm described in section 4.2 and the *matchScansGrid* function. In Fig. 5.21 are reported the true trajectory of the radar in cyan, the estimated trajectory obtained summing the relative pose at each iteration in red and finally, in blue, the estimated trajectory with Kalman filter is shown:

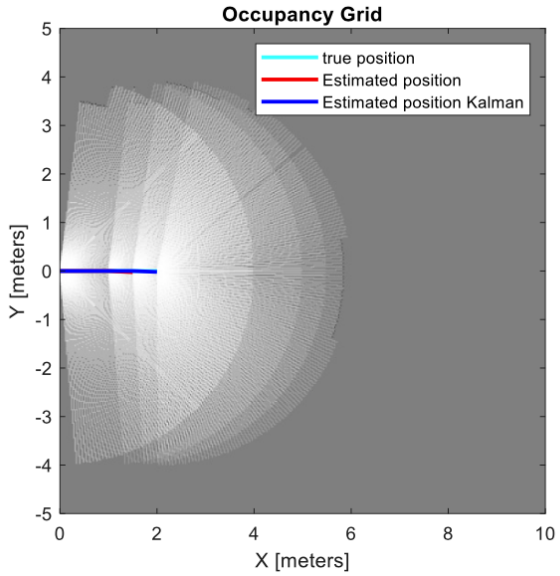


Figure 5.21: Results of the scan-to-scan SLAM algorithm at 300 GHz: 1st scenario

The estimated trajectory perfectly follows the 2 meters actual path but the limited number of measurements leads to a poor mapping performance. The trajectory and the map estimated with the first version of the scan-to-map algorithm described in section 4.4 are shown in Fig. 5.22. Here, the *frameThreshold* is set to 2. We also show the occupancy grid map built starting from the absolute estimated pose with Kalman and the measurement of radar at that position.

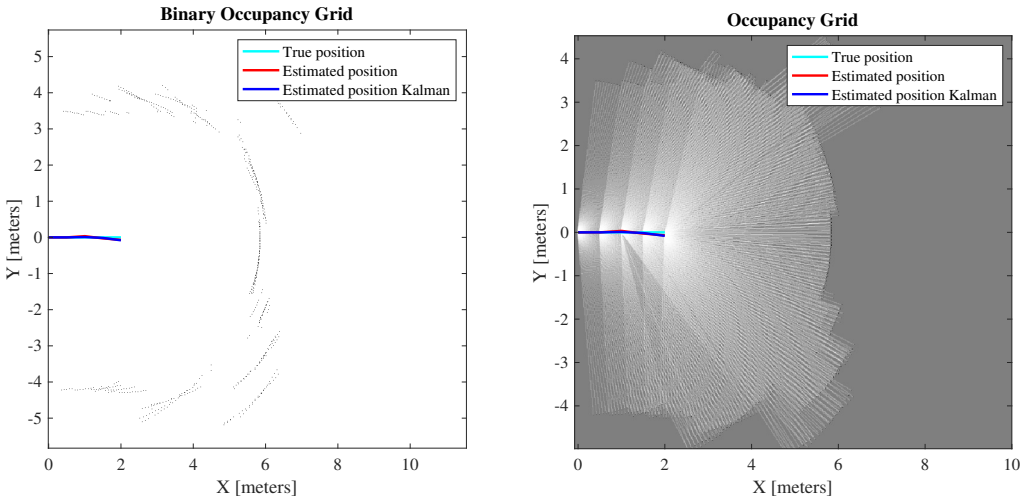


Figure 5.22: Results of 1st version of the scan-to-map SLAM algorithm 300 GHz: 1st scenario

For this scenario, we report also the results obtained with the second approach presented in section 4.5. As the previous results in Fig. 5.22, the results reported in Fig. 5.23 are slightly worse than the results from scan-to-scan algorithm in Fig. 5.21 in particular a little deviation from the real path at the end of the trajectory is visible. In both cases, the binary occupancy map makes clear the location of the walls of the room in the radar's FoV.
In order to achieve these results, we have set thresholds as follows:

- $scanThreshold = 0.7$
- $mapThreshold = 0.4$ [m]

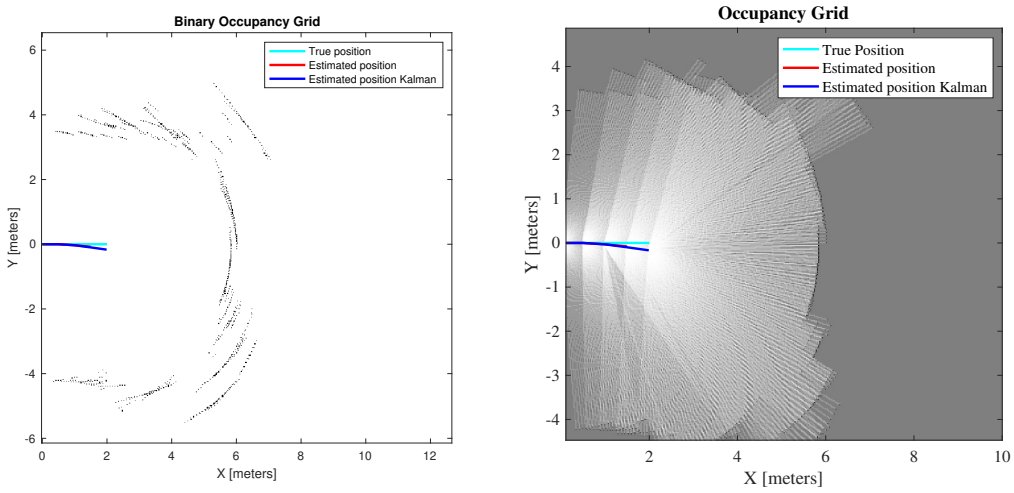


Figure 5.23: Results of 2nd version of the scan-to-map SLAM algorithm 300 GHz: 1st scenario

5.2.2 Second Scenario

In the second scenario, the data set has been collected moving the platform along an oval circuit in the centre of the same rectangular room considered for the first scenario. The radar travels 46 different positions where it takes a scan of the environment. In the straight sections, the radar visits six positions spaced 0.4 meters from each other. In all the positions the radar scans the environment with mechanical steering from $-90°$ to $90°$ with a step of $10°$. In Fig. 5.24 the real scenario above described is shown:

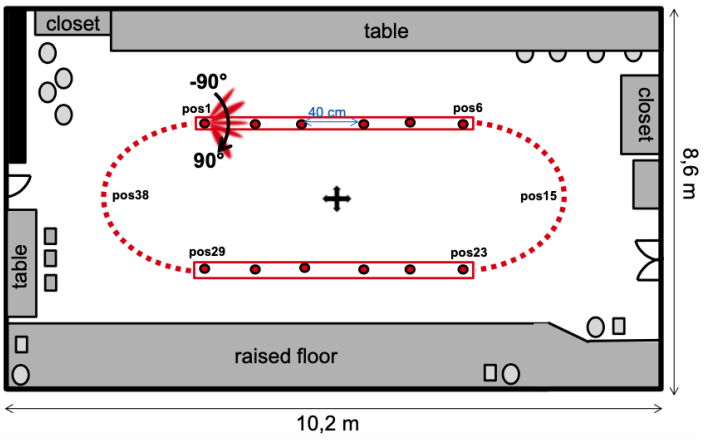


Figure 5.24: Actual trajectory for the second scenario at 300 GHz at CEA-Leti

In Fig. 5.25, we present the results obtained with the scan-to-scan SLAM algorithm described in section 4.2 and *matchScansGrid*.

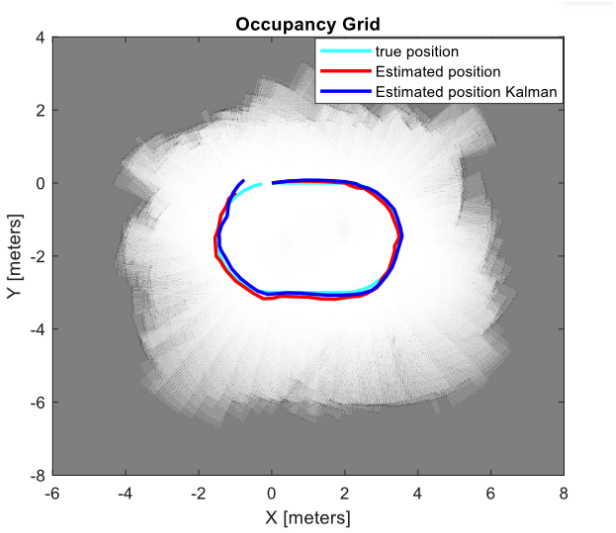


Figure 5.25: Results of the scan-to-scan SLAM algorithm at 300 GHz: 2nd scenario

Although this is an incremental method that considers the matching between consecutive scans, the high frequencies allow to obtain a good estimate of the translation and rotation parameters partially avoiding the problem of outliers frames due to the side lobes effect.

The results with the algorithm described in section 4.4 are shown in Fig. 5.26. Here, the *frameThreshold* is set to 5.

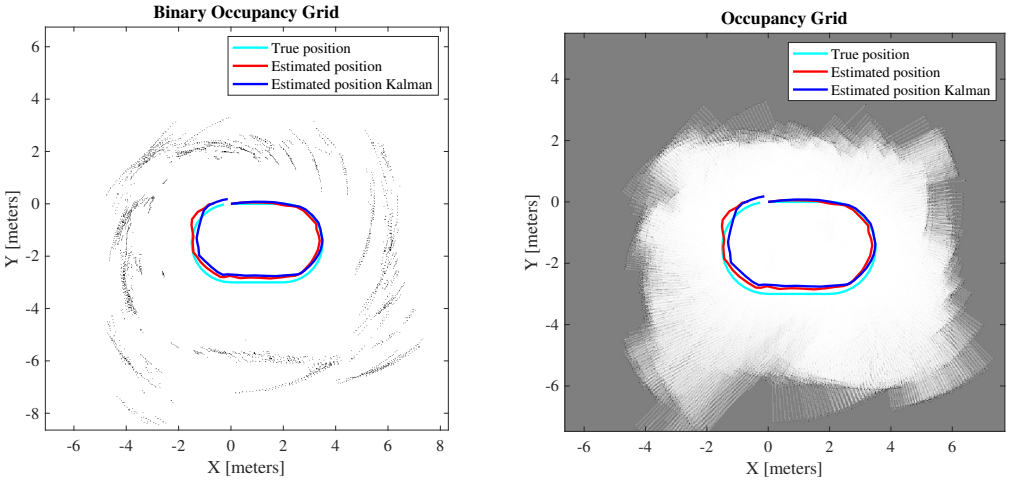


Figure 5.26: Results of 1st version of the scan-to-map algorithm 300 GHz: 2nd scenario

The estimated trajectory faithfully follows the planned path with a slight shift in the lower part of the trajectory.
Finally, in Fig. 5.27 we show the results obtained with the approach in section 4.5. Again, the *frameThreshold* is set to 5. In order to achieve these results, we have set the thresholds as follows:

- $scanThreshold = 0.7$
- $mapThreshold = 0.4$ [m]

The same *frameThreshold* provides results that are worse than the results obtained with the first version of the scan-to-map algorithm both in term of the trajectory and the map.

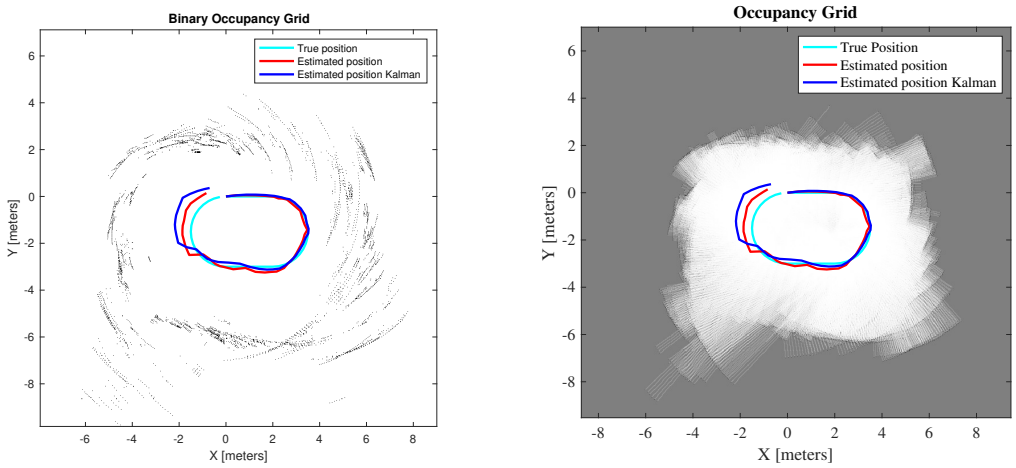


Figure 5.27: Results of 2nd version of the scan-to-map SLAM algorithm 300 GHz: 2nd scenario

The data sets of measurements at 300 GHz, unlike those at 77 GHz, thanks to the high frequencies allow to obtain accurate radar images and consequently precise trajectory estimates also with scan-to-scan algorithms. In fact, the results in Fig. 5.21 and Fig. 5.25 at 300 GHz are very accurate. Here, the outlier measurements that can generate errors in the estimation of the pose are limited so that the drift problem is also limited. On the other hand, also the results of the scan-to-map algorithms follow the true trajectory but with the second version of algorithm, the trajectory is less accurate probably due to imperfect representation of the map in that area.

Chapter 6

Conclusions

The purpose of this thesis was to tackle the simultaneous localization and mapping with millimeter wave technology in indoor scenarios. While SLAM algorithms were mainly developed for lidar and camera sensors providing excellent performance, their application to radar-based sensors is more challenging because of their lower angular resolution and the reflection characteristics of materials at mmWave. However, their use is considered appealing to complement other technologies in applications where these technologies may fail.
The use of data from mmWave radar required the introduction of "cleaning" algorithms to reduce the presence of artefacts in the measurements caused by the non-ideal radiation pattern. Successively, both incremental (scan-to-scan) and partial map-based (scan-to-map) trajectory estimation algorithms were studied in this work.
Analyzing the results obtained from 77 GHz Texas Instrument data, we can say that the scan-to-scan algorithm proposed in section 4.2 follows the expected trajectory until a drift occurs. This drift can be generated by the accumulation of error in pose estimation due to the scan matching function but also it can be due to noisy sensor's measurements. Since here the map is generated starting from the actual poses of the radar if the trajectory assumes a drift also the estimated map presents critical issues.
The strategy in section 4.3 concerns the estimation of the relative pose between two radar images without the need to go through the scan vector considering more information about the measurement and exploiting the Fourier's transform properties. In this case, the results show that even here the trajectory can cause a drift as in the first mentioned case. Furthermore, this strategy makes the algorithm slower from the computational point of view and less accurate in the presence of complex environments from which very noisy radar images arise.

With the scan-to-map approaches described in section 4.4 and 4.5, we managed to mitigate the drift problem of scan-to-scan algorithms in order to improve also the mapping process. In fact, despite this radar technology is not well suited for indoor applications, we have obtained an improvement on the trajectory estimation for all different presented scenarios. While in the scan-to-scan algorithms the two phases of localization and mapping are completely disjoint, with scan-to-map the quality of the map considerably influences the accuracy of the trajectory estimation. In particular, the resolution of the map plays a fundamental role for the accuracy of the results. It is important to underline that the results shown with this approach are sensitive also to the threshold value used to match the current measurement with the map as well as on the thresholds involved in the clean algorithms which preceed the SLAM algorithm. In this direction, future works could provide an objective criterion for parameter setting. Also the second version of the scan-to-map SLAM algorithm is sensitive to the thresholds involved.
These results described at 77 GHz depend heavily on the technology used during the campaigns. In fact, with the Texas Instrument radar, we have been limited by the number of antennas that has not allowed to obtain a narrow main lobe and reduced side lobe. In this sense, we have also presented results obtained through measurement campaigns at 300 GHz. This technology is certainly more performing. This is due to high frequencies that allow to obtain more selective main lobe and at the same time the diffusive component of the backscattered signal increases compared to mmWave which allow to detect the obstacle even with large signal's incident angles. From each scenario we were able to obtain satisfactory results in terms of trajectory estimation and mapping of the real environment.
Comparing the results between the two versions of scan-to-map algorithms, it can be seen that in the case of the 77GHz measurements the results are qualitatively similar but slightly worse than the case of the 300 GHz measurements. In particular, the first method obtains a scan vector from the map which is similar to a lidar measurement while in the second version a point cloud is used for the map and the *matchScansGrid* function may not work properly.
Finally, to improve the trajectory and consequentially the map estimation, additional data measurements, i.e., from IMU, could be merged with radar measurements by exploiting the filtering approaches described in chapter 1, as EKF or UKF. In addition, SLAM can be improved if Doppler information, gathered from 77 GHz radar, is accounted for. Future works about SLAM with mmWave radar technology could also address the use of the image processing theory, computer vision elements or neural networks theory to further improve the results.

List of Figures

Bibliography

[1] F. Guidi, A. Guerra and D. Dardari, "Personal Mobile Radars with Millimeter-Wave Massive Arrays for Indoor Mapping," in IEEE Transactions on Mobile Computing, vol. 15, no. 6, pp. 1471-1484, 1 June 2016.

[2] F. Guidi, A. Guerra and D. Dardari, "Millimeter-wave massive arrays for indoor SLAM," 2014 IEEE International Conference on Communications Workshops (ICC), Sydney, NSW, 2014, pp. 114-120.

[3] A. Guerra, F. Guidi, J. Dall'Ara and D. Dardari, "Occupancy Grid Mapping for Personal Radar Applications," 2018 IEEE Statistical Signal Processing Workshop (SSP), Freiburg, 2018, pp. 766-770.

[4] A. Guerra, F. Guidi and D. Dardari, "Millimeter-wave personal radars for 3D environment mapping," 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2014, pp. 701-705.

[5] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," in IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1309-1332, Dec. 2016.

[6] Simo Särkkä (2013). Bayesian Filtering and Smoothing. Cambridge University Press.

[7] D. Dardari, P. Closas and P. M. Djurić, "Indoor Tracking: Theory, Methods, and Technologies," in IEEE Transactions on Vehicular Technology, vol. 64, no. 4, pp. 1263-1278, April 2015.

[8] Davide Dardari, Bayesian filtering, University of Bologna, 2017

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I" in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006.

[10] J. Mochnac, S. Marchevsky and P. Kocan, "Bayesian filtering techniques: Kalman and extended Kalman filter basics," 2009 19th International Conference Radioelektronika, Bratislava, 2009, pp. 119-122.

[11] S. J. Julier and J. K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems." In Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls., 1997.

[12] Cyrill Stachniss and Wolfram Burgard (2014), "Particle Filters for Robot Navigation", Foundations and Trends in Robotics: Vol. 3: No. 4, pp 211-282.

[13] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "A Tutorial on Graph-Based SLAM" in IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31-43, winter 2010.

[14] S. Dogru and L. Marques, "Using Radar for Grid Based Indoor Mapping," 2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Porto, Portugal, 2019, pp. 1-6.

[15] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes" in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[16] A. Censi, "An ICP variant using a point-to-line metric" 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, 2008, pp. 19-25.

[17] Peter Biber and Wolfgang Straßer, "The normal distributions transform: A new approach to laser scan matching." In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pages 2743-2748, Las Vegas, USA, October 2003.

[18] Magnusson, M. "The Three-Dimensional Normal-Distributions Transform - an Efficient Representation for Registration, Surface Analysis, and Loop Detection." Ph.D. Thesis. Örebro University, Örebro, Sweden, 2013.

[19] T. Takubo, T. Kaminade, Y. Mae, K. Ohara and T. Arai, "NDT scan matching method for high resolution grid map," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009

[20] W. Qingshan and Z. Jun, “Point Cloud Registration Algorithm Based on Combination of NDT and PLICP,” 2019 15th International Conference on Computational Intelligence and Security (CIS), Macao, Macao, 2019, pp. 132-136.

[21] E. B. Olson, “Real-time correlative scan matching,” 2009 IEEE International Conference on Robotics and Automation, Kobe, 2009, pp. 4387-4393.

[22] W. Hess, D. Kohler, H. Rapp and D. Andor, “Real-time loop closure in 2D LIDAR SLAM,” 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278.

[23] Li Y, Liu Y, Wang Y, Lin Y, Shen W. “The Millimeter-Wave Radar SLAM Assisted by the RCS Feature of the Target and IMU,” Sensors (Basel). 2020 Sep 22.

[24] B. S. Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” in IEEE Transactions on Image Processing, vol. 5, no. 8, pp. 1266-1271, Aug. 1996, doi: 10.1109/83.506761.

[25] Mathworks, Navigation Toolbox, URL : https://www.mathworks.com/help/nav/index.html

[26] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, Principles of modern radar. Citeseer, 2010.

[27] Texas Instument. “The foundamentals of millimeter wave sensors,” URL: https://www.ti.com/lit/pdf/spyy005 , Aug.27,2020.

[28] Texas Instruments. IWR1443-Datasheet. URL: https://www.ti.com/lit/ds/symlink/iwr1443.pdf. Oct. 2018.

[29] Texas Instrument. mmWave Demo Visualizer ver. 2.1. URL: http://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/2.1.0.

Ringraziamenti

Voglio ringraziare sentitamente il Prof. Davide Dardari che con la sua cordialità, disponibilà e opportuni consigli mi ha dato la possibilità di lavorare a questo stimolante progetto e di redigere questa tesi. Ringrazio anche i Dott. Anna Guerra, Francesco Guidi e Gianni Pasolini, correlatori di tesi, sempre molto disponibili. È stato un piacere collaborare con voi.
Arrivato alla fine del mio percorso universitario, mi sento di ringraziare la mia famiglia che mi ha permesso di affrontarlo con serenità e mi ha supportato e sopportato nei momenti più delicati facendomi credere sempre in me stesso.
Un pensiero va anche ai miei più cari amici che hanno condiviso con me questo percorso fuori dalle mura universitarie ma anche ai miei compagni di corso, fondamentali per il raggiungimento dell'obiettivo.
Avrei voluto concludere il mio cammino nella normalità vivendo la vita universitaria fra i banchi del campus ma purtroppo, a causa di questa pandemia, la normalità non sappiamo nemmeno più cosa sia.