Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA ELETTRONICA, INFORMATICA E DELLE TELECOMUNICAZIONI

Ciclo XXVII

Settore Concorsuale di afferenza: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico disciplinare: ING-INF/03 - TELECOMUNICAZIONI

TARGET TRACKING IN UWB MULTISTATIC RADARS

Presentata da: BITA SOBHANI

Coordinatore Dottorato

Prof. Dr.
ALESSANDRO
VANELLI-CORALLI

Relatore

Prof. Dr.
MARCO CHIANI

Co-Relatore

Dr.
ENRICO PAOLINI

Esame finale anno 2015

To the one who always keeps track of
everything everywhere.

Abstract

Detection, localization and tracking of non-collaborative objects moving inside an area is of great interest to many surveillance applications. An ultra-wideband (UWB) multistatic radar is considered as a good infrastructure for such anti-intruder systems, due to the high range resolution provided by the UWB impulse-radio and the spatial diversity achieved with a multistatic configuration.

Detection of targets, which are typically human beings, is a challenging task due to reflections from unwanted objects in the area, shadowing, antenna cross-talks, low transmit power, and the blind zones arised from intrinsic peculiarities of UWB multistatic radars.

Hence, we propose more effective detection, localization, as well as clutter removal techniques for these systems. However, the majority of the thesis effort is devoted to the tracking phase, which is an essential part for improving the localization accuracy, predicting the target position and filling out the missed detections.

Since UWB radars are not linear Gaussian systems, the widely used tracking filters, such as the Kalman filter, are not expected to provide a satisfactory performance. Thus, we propose the Bayesian filter as an appropriate candidate for UWB radars. In particular, we develop tracking algorithms based on particle filtering, which is the most common approximation of Bayesian filtering, for both single and multiple target scenarios. Also, we propose some effective detection and tracking algorithms based on image processing tools.

We evaluate the performance of our proposed approaches by numerical simulations. Moreover, we provide experimental results by channel measurements for tracking a person walking in an indoor area, with the presence of a significant clutter. We discuss the existing practical issues and address them by proposing more robust algorithms.

Acknowledgement

I would like to express my special appreciation and thanks to my supervisor Prof. Dr. Marco Chiani. Your unflinching support, encouragement, advice and constructive criticism have been tremendous in making me grow.

I would also like to thank my advisor Dr. Enrico Paolini, as well as Dr. Andrea Giorgetti and Dr. Matteo Mazzotti for their brilliant comments and suggestions on my research during this time.

I would especially like to thank Prof. Dr. Thomas Zwick for hosting me in the IHE institute, and for many inspiring discussions on my research and teaching me the essential experimental skills. Your help and advice have been priceless.

Finally, my deepest gratitude goes to my very dear family who has been always there to support me.

Publications

Parts of the work in this thesis have previously appeared in the following publications:

[1] B. Sobhani, E. Paolini, A. Giorgetti, M. Mazzotti, and M. Chiani, "Bayesian tracking in UWB radar sensor networks," in *IEEE Int. Conf. Commun. Workshop on Advances in Network Localization and Navigation (ANLN)*, June 2013, pp. 47–51.

[2] B. Sobhani, M. Mazzotti, E. Paolini, A. Giorgetti, and M. Chiani, "Effect of state space partitioning on bayesian tracking for UWB radar sensor networks," in *IEEE Int. Conf. on Ultra-Wideband (ICUWB)*, Sep. 2013, pp. 120–125.

[3] B. Sobhani, E. Paolini, A. Giorgetti, M. Mazzotti, and M. Chiani, "Target tracking for UWB multistatic radar sensor networks," *IEEE J. Sel. Topics Signal Processing*, vol. 8, no. 1, pp. 125–136, Feb. 2014.

[4] B. Sobhani, M. Mazzotti, A. Giorgetti, E. Paolini, and M. Chiani, "Channel gain estimation in UWB multistatic radars in the presence of multiple targets," in *European Conf. on Networks and Comm. (EuCNC)*, June 2014.

[5] B. Sobhani, M. Mazzotti, E. Paolini, A. Giorgetti, and M. Chiani, "Multiple target detection and localization in UWB multistatic radars," in *IEEE Int. Conf. on Ultra-WideBand (ICUWB)*, Sep. 2014, pp. 135–140.

[6] B. Sobhani, T. Zwick, and M. Chiani, "Target TOA association with the Hough transform in UWB radars," *submitted to IEEE Trans. on Aerospace and Electronic Systems (TAES)*, 2015.

[7] B. Sobhani, E. Paolini, M. Mazzotti, A. Giorgetti, and M. Chiani, "Multiple target tracking with particle filtering in UWB radar sensor networks," in *submitted to IEEE Int. Conf. Commun. Workshop on Advances in Network Localization and Navigation (ANLN)*, 2015.

Contents

Chapter 1

Introduction

Precise localization and tracking of moving targets inside an area is of great interest for several surveillance applications. Localization is usually intended as the capability to locate friendly collaborative objects. This is sometimes referred to as *active localization*, because the object to be localized collaborates to its localization process [8]. However, an increasing attention is recently being devoted to the capability of detecting and tracking unfriendly non-cooperative objects within a given area. This is referred to as *passive localization* and is typical of radar systems [9, 10]. In order to improve the localization performance of the system, tracking is an essential component to achieve high-level system reliability.

Unlike a monostatic radar, which uses the same antenna for both the transmitted and received signal, a multistatic radar uses at least three non co-located antennas for transmitting and receiving [11]. The major advantages of a multistatic radar over a conventional monostatic one include a wider area coverage and a higher amount of information available due to spatial diversity. Moreover, receivers in a multistatic radar system are not required to transmit any signal, which enables development of low power and low cost equipments. A multistatic radar is sometimes called a *radar sensor network*. When multiple antennas are used in each node, the resulting sensor network is also called multiple-input/multiple-output (MIMO) radar [12]. These features lead to improved performance and foster new applications such as anti-intruder surveillance, ambient monitoring in safety and healthcare applications or location-aware commercial services.

A promising wireless technique for such multistatic radar applications is the ultra-wideband (UWB) technology. By definition, a UWB radar has a large fractional bandwidth, whereas a narrowband radar has a very small fractional bandwidth. We will focus on impulse radio-UWB (IR-UWB) radars which are characterized by transmission of short duration pulses of

the order of a few nanoseconds [13, 14]. IR-UWB offers an extraordinary resolution and localization precision. Additional advantages include low power consumption, high spatial resolution (typically a few centimeters) even in indoor environments with dense multipath, high security and low probability to be intercepted, co-existence with a large number of devices operating in small areas, and robustness to narrowband jamming [15, 16]. These features make IR-UWB radars suitable for various recent applications such as ground penetration [17], human being detection [18, 19], and biomedical applications [20].

The above features and the fact that IR-UWB devices are usually lightweight, cost-effective, and characterized by low-power emissions, make UWB an ideal candidate for short-range radar applications [10, 15]. The radar sensor networks addressed in this thesis are UWB multistatic radar systems aimed at detecting and tracking non-cooperative targets (e.g., human subjects) moving inside a surveillance area.

Passive localization through multistatic UWB radars is the subject of several works such as [21, 22, 23, 24, 25, 26] a few of which, however, focus on tracking aspects. However, a few of them focus on tracking aspects. In UWB systems, the measurement equation is non-linear and also the measurement noise cannot be assumed as Gaussian. Therefore, the commonly used tracking filters based on linearity and Gaussianity assumptions, such as the Kalman filter, are not expected to exhibit a satisfactory performance, even if they may still represent a valuable choice due to their low complexities [27].

1.1 Thesis Outline

Chapter 2 explains IR-UWB technology. It describes the multistatic radar system setup used during this thesis for anti-intruder applications. The detection problem associated with *blind zones*, i.e., regions inside which the target cannot be detected by at least one of the receivers, is also presented.

Chapter 3 introduces Bayesian filtering as a good candidate for UWB radar systems. Such filtering techniques are not based on any linearity and Gaussianity assumption for the measurement equation and noise. The grid-based Bayesian approach and particle filtering are then explained as the two common implementation techniques for Bayesian filters.

In Chapter 4, the particle filtering is employed for UWB multistatic radars composed of one transmitter and multiple receivers. The corresponding particle weights are derived analytically for that particular system, based on observations from all receivers.

The conventional particle filtering may lead to track divergence due to

intrinsic peculiarities of UWB multistatic radars. Mainly, these divergences are caused by blind zones. To cope with this problem, a low-complexity modified particle filter is proposed which improves the target velocity estimation. The performance of the proposed approach is evaluated numerically accounting for the spatial configuration of the receivers, propagation effects, the presence of residual clutter, power constraints imposed on UWB transmitted signals, and noise. We show that the proposed algorithm provides a high estimation accuracy compared to the Kalman filter, even at low signal-to-noise ratios, in the presence of either static or dynamic clutter. Moreover, it can track complicated manoeuvering target trajectories. We remark that the proposed algorithm based on particle filtering can be implemented for real-time tracking.

In Chapter 5, the proposed particle filtering is compared numerically with the grid-based Bayesian approach in terms of both tracking performance and algorithm complexity. We show by numerical results that the grid-based tracking method outperforms the particle filtering, especially in proximity of blind zones, at the cost of a much higher computational complexity and memory requirements.

Chapter 6 focuses on detection and localization steps which have to be performed before tracking in multiple target scenarios. A constant false alarm rate (CFAR) detection approach based on median filtering is proposed which allows us to reduce the CFAR threshold to a very low value and hence to heavily reduce the number of missed detections. A pixel-based localization technique using CFAR detector is then proposed and compared with the conventional direct method of localization (trilateration). The performance of all the above techniques is evaluated numerically in a multiple target scenario.

The particle filtering algorithm proposed in Chapter 4 is suitable for tracking a single target. Chapter 7 extends this algorithm to the case of multiple targets. A closed-form equation for estimating the channel attenuations in UWB multistatic radars with one transmitter and several receivers is derived by means of the maximum-likelihood approach. The estimated channel gains are then used to calculate the particle weights. The performance of the proposed approach is compared to the Kalman filter through numerical simulations.

Chapter 8 employs the Hough transform to detect and track the target at the same time. Also, it tracks all of the targets simultaneously in a multiple target scenario. We propose two different approaches based on the Hough transform.

In one approach, the Hough transform finds the time of arrival (TOA) curves of the targets. In the other approach, the Hough transform extracts

the trajectory of the targets. In both techniques, a straight-line constant-velocity model was adopted for the target movement, representing the human walk.

Moreover, a new empty-room clutter removal technique is proposed in this chapter to combat the shadowing effect, particularly severe in indoor scenarios. We show that the proposed approach can considerably reduce the number of false alarms.

We evaluate the performance of all the proposed techniques by experimental results provided by channel measurements which were carried out in an indoor environment with significant clutter.

Chapter 9 concludes the algorithms presented during this thesis and provides directions for future works.

Chapter 2

Impulse-Radio UWB Multistatic Radar

A promising technology for anti-intruder radar sensor networks is ultra-wideband (UWB), where signals are characterised by a large fractional bandwidth. We will focus on the impulse radio-UWB (IR-UWB) which is based on transmitting ultra short (in the order of nanosecond) pulses, properly shaped to work in the selected frequency band [28, 29]. In this way, it is possible to avoid the use of a carrier signal, thus lowering the complexity of the radio transceiver.

The other possible way for implementing UWB communication is the multi-carrier or multi-band UWB (MB-UWB) [30]. The MB-UWB uses orthogonal frequency division multiplexing (OFDM) techniques to transmit the information on each of the sub-bands. The OFDM has several good properties, including high spectral efficiency, robustness to RF and multi-path interferences. However, up and down conversions are required, making the system very sensitive to frequency, clock, and phase inaccuracy. Moreover, nonlinear amplification may jeopardize OFDM orthogonality. For these reasons, MB-UWB is less appealing for low-power and low cost applications.

Carrierless transmission in IR-UWB has the benefit of requiring fewer RF components than carrier-based techniques, permitting the realization of simple and inexpensive architectures. For instance, there is no need for mixers and local oscillators to convert the carrier frequency to another required frequency. Consequently, there is no need for a carrier recovery stage at the receiver. This simplicity makes a low-cost implementation of IR-UWB transceivers possible.

Thanks to the ultra-short duration pulses, IR-UWB is very robust against multipath, as more multipath components (clutter) can be resolved at the receiver. Therefore, sub-centimeter ranging is (theoretically) possible with

IR-UWB. Moreover, eavesdropping the IR-UWB signal is a very hard task [16].

All the above features and the fact that IR-UWB devices are light-weight, cost-effective, and characterized by low-power emissions, make IR-UWB suitable for various recent short-range radar applications such as ground penetration [17], human being detection [18, 19], and biomedical applications [20].

In this chapter, we introduce the IR-UWB technology used in a multistatic radar system, aimed at detecting and tracking of non-cooperative targets (e.g., human subjects) moving inside a surveillance area. Section 2.1, explains the UWB multistatic radar system setup used during this thesis for anti-intruder applications. Section 2.2 discusses the detection problems intrinsically associated with such a system configuration. Conclusions are provided in Section 2.3.

2.1 The UWB Multistatic Radar Setup

We consider a UWB multistatic radar system composed of one transmitter (TX) and N_R receivers (RX). In order to provide a full area coverage, the transmitter and multiple receivers may, for example, be deployed on the perimeter of the area as depicted in Fig. 2.1. A central node collects the received signals from all of the receivers and performs the required data processing.

We consider an IR-UWB radar which transmits continuously a sequence of UWB pulses at time intervals T_{IP} (around tens of nanoseconds). The received signals are observed and processed in time intervals of duration T_{scan} (a fraction of a second), called *scan* time or *frame* time (T_F). If a number of N_s pulses are emitted by the transmitter during each scan time, then $T_{scan} = N_s T_{IP}$. The only process which is needed to be done on the corresponding N_s received signals in each scan time to form the observation signal, is averaging them along T_{IP}. This increases the signal-to-noise ratio (SNR) by the processing gain N_s. The system timings are designed in such a way that the channel response to a single pulse, when a moving target is present, does not change appreciably during a scan time so that averaging is feasible.

The transmitted pulse at time t is assumed to be the first derivative Gaussian monocycle

$$p(t) = A\, t \exp\left\{-\frac{t^2}{2{\tau_p}^2}\right\} \tag{2.1}$$

with duration parameter τ_p. The pulse is modulated with a sinusoidal carrier.

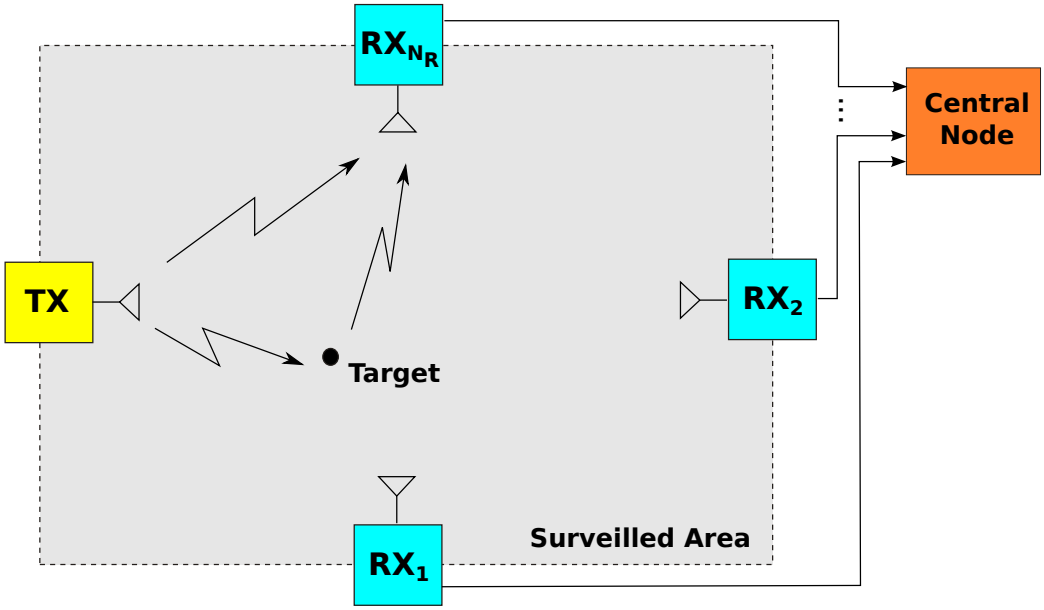


Figure 2.1: Pictorial representation of the UWB multistatic radar system considered in this thesis.

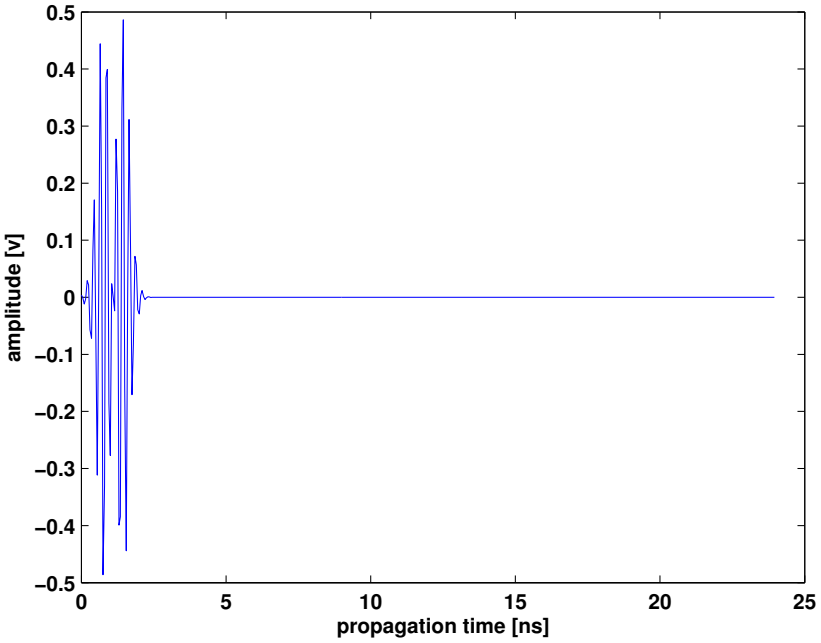
The modulation for the first derivative order is required to remain in FCC regulations [31]. Although in this thesis, we have chosen the first derivative order as an example, higher derivative orders may also be considered without any modulation. The main concept of the proposed tracking algorithms in the next chapters doesn't depend on the UWB pulse shape used.

Fig. 2.2 shows an example of the first derivative Gaussian monocycle with $\tau_p = 0.3$ns , modulated with a carrier frequency of 4.5GHz, in both time and frequency domains. The FCC mask is also shown in the figure. In this example, $T_{IP} = 24$ns and the amplitude of the pulse was set to fulfill the FCC mask.

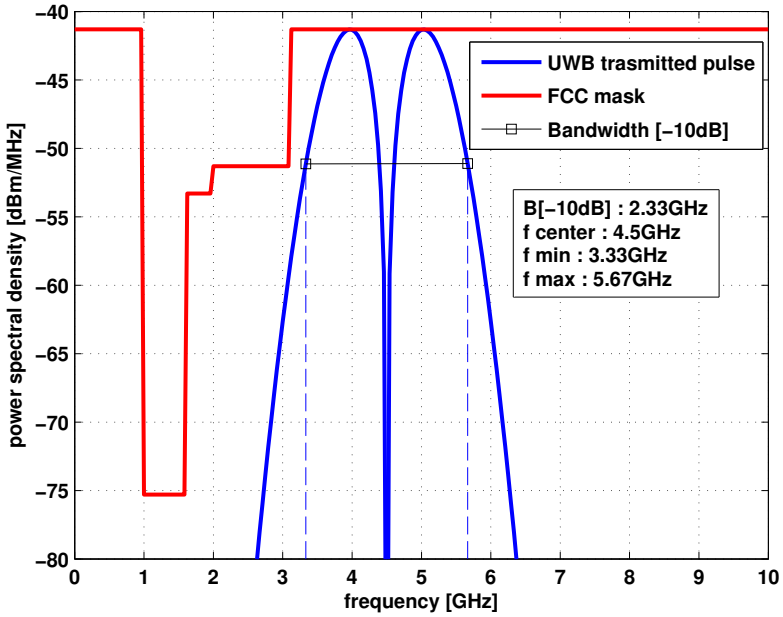
If a target is present inside the area, the received signal at each receiver corresponding to a UWB transmitted pulse consists of the direct path pulse followed by pulse replicas due to both the clutter and the target, and the noise.

2.2 The Blind Zone Problem

One of the problems in UWB multistatic radar systems are blind zones for each pair of TX and RX antennas. Blind zones are regions inside which



(a) The pulse in time domain.



(b) The pulse power spectral density.

Figure 2.2: A first derivative Gaussian monocycle pulse.

the difference between the delays corresponding to target-reflected path and direct path is lower than system time resolution. When the target is close to a line-of-sight (LOS), connecting the TX and any of the RX antennas, it may fall inside the corresponding blind zone, in which case it can not be detected by that RX antenna [10]. Blind zones are consequence of two main concurrent phenomena. Firstly, this problem always arises in presence of a non-ideal synchronization. In the considered system, in fact, synchronization is the first task to be performed, based on TOA estimation of the direct-path pulse. This operation is essential both to align the N_s pulse responses belonging to the same frame in order to achieve a process gain and to align successive frames to perform clutter removal. Synchronization based on TOA estimation is never ideal, thus leading to an imperfect frame alignment. A residue of direct-path pulse is then always present in real systems after clutter removal processing. The problem is that the direct path is usually much stronger than the target echo and the other clutter components, due to the fact that it does not experience reflections. Hence, the direct path residue after clutter removal typically masks the echo of a moving target that is close to the direct TX-RX path.

Another issue concurring to the blind zone effect, even in presence of a perfect TOA estimation, is related to sampling resolution which maps the target position onto *quantized* ellipses inside the area. For example, denote the propagation delay times corresponding to target and direct path by $\tau_{target} = (l_r + l_t)/c$ and $\tau_{LOS} = l/c$, respectively, where l_r and l_t are the target-RX and TX-target distances and l is the TX-RX distance. Moreover, denote by T_s the sampling period of the received signal, and by k_1 and k_0 the integers $\lceil \tau_{target}/T_s \rceil$ and $\lceil \tau_{LOS}/T_s \rceil$, respectively. For $c(k_0 - 1)T_s < l_r + l_t \leq ck_0T_s$ we have $k_1 = k_0$ and the target position is assumed on the direct TX-RX path, leading to a mismatch between the actual and the estimated value of $l_r + l_t$. This position estimation error caused by finite sampling, which is present whatever the target position, is prominent in the proximity of the direct TX-RX path. Fig. 2.3 depicts quantized ellipses for each pair of TX and RX antennas that correspond to the points such that $\lceil \frac{l_r+l_t}{cT_s} \rceil = k$, where $k \in \{k_0, k_0 + 1, \ldots, k_0 + 15\}$ from the inner ellipse toward the outer one. For each TX–RX pair, all the points within the most inner ellipse are mapped onto the direct TX–RX path.

2.3 Conclusion

In this chapter, we introduced the IR-UWB technology as an appropriate choice for short-range radar applications, mainly due to its super resolution

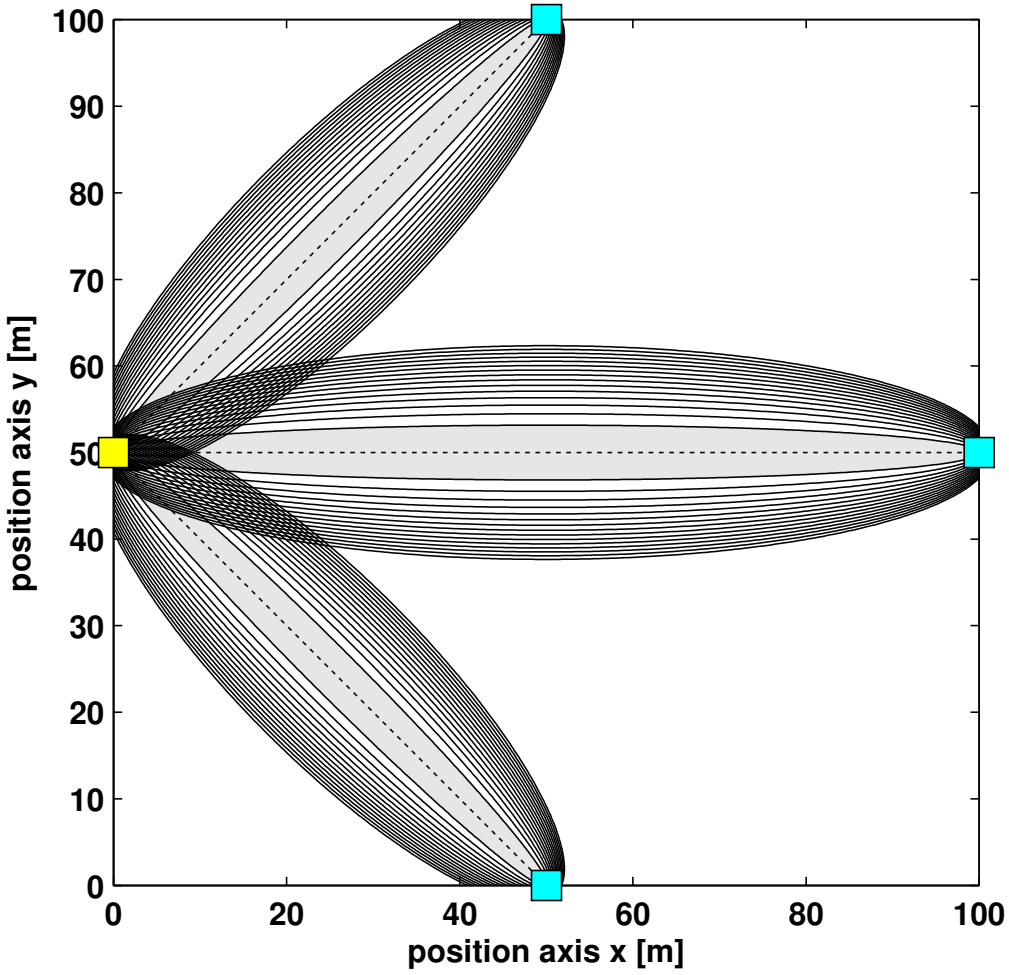


Figure 2.3: Example of quantized ellipses in case of three receivers.

in range and low-cost implementation. We presented a multistatic radar system configuration using IR-UWB, aimed at detecting and tracking of non-collaborating targets moving inside a surveillance area. We also discussed the detection problems caused by blind zones which are inherently associated with such systems.

Chapter 3

Bayesian Filtering

Estimating the state of a dynamic system that evolves with time based on a set of noisy measurements is the problem of many applications. In radar tracking problems, the system state is usually the mobility characteristics of the target. For example, the system state can be a vector consisting of the target position, velocity and acceleration.

In order to solve this estimation problem, we need two models. One model is required to define how the system state is changing over time. This is called the *system* or *dynamic model*. Another model should describe how the measurements are related to the system state. This is called the *measurement* or *observation model*. In this thesis, we consider discrete-time models. When these models are known in a probabilistic form, the Bayesian filter is the optimal solution to the estimation problem.

The Bayesian filter recursively predicts the probability density function (p.d.f.) of the system state based on the system model, and corrects the predicted p.d.f. based on the current observation. This filtering technique can be applied to a general non linear system with embedded non Gaussian noises. However, the integrals over probability density functions involved in the Bayesian filtering may not be generally solved in close forms. Therefore, usually some integration approximations are made which result in particular types of Bayesian filtering. Among these implementation techniques, we focus on the grid-based method and particle filters which are the most common solutions to non linear tracking problems.

Section 3.1 introduces the general Bayesian filter. Section 3.2 describes the well known Kalman filter as a simplified form of the Bayesian filter, when the system and measurement models are linear and the noises are additive Gaussian. Section 3.3 and Section 3.4 explain the grid-based method and particle filters as the two popular solutions to non linear systems with embedded non Gaussian noises. Conclusions are provided in Section 3.5.

3.1 General Bayesian Filter

A non-linear stochastic system can be defined by the dynamic model (also called system model):

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{3.1}$$

and the measurement (observation) model:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k) \tag{3.2}$$

where $\mathbf{x}_k$ is the state vector at time index k, $\mathbf{z}_k$ is the observation vector, and $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are independent, identically distributed (i.i.d.) dynamic (process) and measurement noise vectors, respectively. The deterministic functions $\mathbf{f}_k$ and $\mathbf{h}_k$ (possibly non-linear) relate the previous state $\mathbf{x}_{k-1}$ to the current state $\mathbf{x}_k$ and the current state $\mathbf{x}_k$ to the observation $\mathbf{z}_k$, respectively.

In the Bayesian filtering, the problem is to estimate the current state $\mathbf{x}_k$ based on the set of all available observations $\mathbf{z}_{1:k} \triangleq \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$. In other words, it is required to calculate the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. This can be done recursively in two steps of prediction and correction (update).

The prediction step calculates the prior density $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ using:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \tag{3.3}$$

where the dynamic model defined in (3.1) specifies the predictive transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ assuming a Markov process of order one.

In the correction step, the current observation $\mathbf{z}_k$ is used to correct the prior density (3.3) to obtain the required posterior density:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \tag{3.4}$$

where the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ is specified by the measurement model defined in (3.2) and $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ is a normalizing constant which can be calculated by:

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k \tag{3.5}$$

The two recursive equations (3.3) and (3.4) are the basis of Bayesian filtering. In general, the integrals in these equations can not be evaluated in closed forms. Therefore, some numerical approximations are required for solving the integrals.

Fig. 3.1 shows various implementation techniques of Bayesian filtering based on the simplifying assumptions and numerical approximations in solving the above equations. When the dynamic and measurement equations are

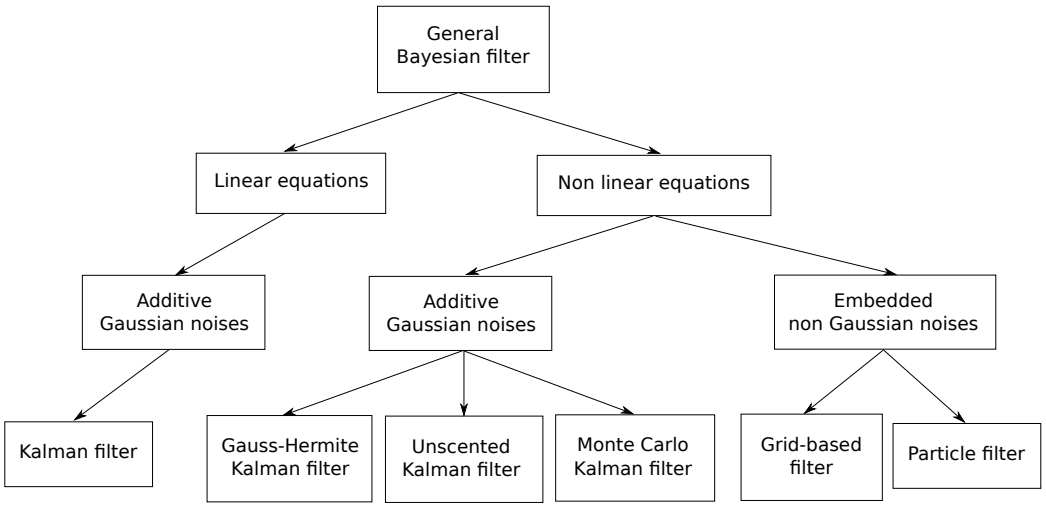


Figure 3.1: Various implementation of Bayesian filtering.

linear and their corresponding noises are additive and Gaussian, the Bayesian filter is actually converted to the well known Kalman filter. When the measurement and observation noises are additive and Gaussian, but the equations are not necessarily linear, the Bayesian filter can still be approximated with a Kalman filter. According to the type of numerical approximation used in this technique, the Bayesian filter results in the so called Gauss-Hermite Kalman filter (GHKF), Unscented Kalman filter (UKF) and Monte Carlo Kalman filter (MCKF).

In the general case, when the equations are non linear and the noises are non additive and/or non Gaussian, the grid-based method and particle filters are the most common implementation of Bayesian filtering.

3.2 Kalman Filter

When the dynamic equation $\mathbf{f}_k$ is a linear function of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$, and the measurement equation $\mathbf{h}_k$ is a linear function of $\mathbf{x}_k$ and $\mathbf{n}_k$, and their corresponding noises $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are additive Gaussian with known mean and variance, then the Bayesian filter is actually simplified to the Kalman filter. With these assumptions, (3.1) and (3.2) can be written as:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{3.6}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k \tag{3.7}$$

where $\mathbf{F}_k$ and $\mathbf{H}_k$ are known matrices which define the linear functions. If we assume that Gaussian noises $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are statistically independent with zero mean and covariance matrices $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$, i.e., $\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{v}_{k-1}; \mathbf{0}, \mathbf{Q}_{k-1})$ and $\mathbf{n}_k \sim \mathcal{N}(\mathbf{n}_k; \mathbf{0}, \mathbf{R}_k)$, respectively, then the prediction and correction steps in the general Bayesian filter are converted to the following recursive equations:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \tag{3.8}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{3.9}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \tag{3.10}$$

where

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \tag{3.11}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1} \tag{3.12}$$

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \tag{3.13}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{3.14}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{3.15}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \tag{3.16}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{3.17}$$

In the above equations, $\tilde{\mathbf{y}}_k$ is the innovation term with the covariance $\mathbf{S}_k$, and $\mathbf{K}_k$ is the Kalman gain. The Kalman filter is indeed the optimal solution for tracking problems in linear Gaussian environments.

3.3 Grid-based Filter

If we approximate the continuous state space $\mathbf{x}_k$ with a discrete state space consisting of N_p pixels $\mathbf{x}_k^1, \cdots, \mathbf{x}_k^{N_p}$, and we denote the center of i'th pixel $\mathbf{x}_k^i$ with $\bar{\mathbf{x}}_k^i$, and the conditional probability $Pr(\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i|\mathbf{z}_{1:k-1})$ with $w_{k-1|k-1}^i$, then the prediction and correction steps in the Bayesian filtering can be approximated as:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \sum_{i=1}^{N_p} w_{k-1|k-1}^i \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i) \tag{3.18}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \approx \sum_{i=1}^{N_p} w_{k|k-1}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{3.19}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_{k|k}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{3.20}$$

where

$$w^i_{k|k-1} \approx \sum_{j=1}^{N_p} w^j_{k-1|k-1} p(\bar{\mathbf{x}}^i_k | \bar{\mathbf{x}}^j_{k-1}) \tag{3.21}$$

$$w^i_{k|k} \approx \frac{w^i_{k|k-1} p(\mathbf{z}_k | \bar{\mathbf{x}}^i_k)}{\sum_{j=1}^{N_p} w^j_{k|k-1} p(\mathbf{z}_k | \bar{\mathbf{x}}^j_k)} \tag{3.22}$$

However, in order to have a good a performance, the number of pixels should be sufficiently large which leads to high computational complexity and memory requirements. Also, higher dimensionality of the state space increases the computational cost dramatically. Moreover, since the partitioning of the state space is fix and predefined, we can not change it dynamically to have higher resolutions in the regions with more probability of target presence.

3.4 Particle Filter

3.4.1 Sequential Importance Sampling Particle Filter

In particle filtering techniques, the posterior density is approximated using a set of random samples (particles) with associated weights. With larger number of samples, this representation becomes more accurate and the particle filter approaches the general Bayesian filter.

If we denote the N_p particles by $\mathbf{x}^1_k, \cdots, \mathbf{x}^{N_p}_k$ and their corresponding weights by $w^1_k, \cdots, w^{N_p}_k$, then the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be approximated as:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w^i_k \delta(\mathbf{x}_k - \mathbf{x}^i_k) \tag{3.23}$$

Usually, the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can not be represented by any analytical p.d.f. and therefore, it is difficult to draw samples from. On the other hand, assume that $q(\mathbf{x}_k | \mathbf{z}_{1:k})$ is another density which can be expressed analytically. This density from which the samples can be easily taken is called the *Importance Density*. Then, the weights in (3.23) are defined as:

$$w^i_k \propto \frac{p(\mathbf{x}^i_k | \mathbf{z}_{1:k})}{q(\mathbf{x}^i_k | \mathbf{z}_{1:k})} \tag{3.24}$$

After some calculations the weights can be updated recursively by:

$$w^i_k \propto w^i_{k-1} \frac{p(\mathbf{z}_k | \mathbf{x}^i_k) p(\mathbf{x}^i_k | \mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k | \mathbf{x}^i_{k-1}, \mathbf{z}_k)} \tag{3.25}$$

The above equation is used to define a type of particle filtering called sequential importance sampling (SIS) particle filter. The steps are described by Algorithm 1.

Algorithm 1 SIS Particle Filter

1: **for** $i = 1 \rightarrow N_p$ **do**
2: Draw particle $\mathbf{x}_k^i$ from $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
3: Update the particle weight by $w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\mathbf{z}_k)}$
4: **end for**

One of the problems associated with the SIS particle filter is that after some iterations, most of the particle weights tend to zero so that after a while, only one particle leads the algorithm. Therefore, most portion of the algorithm computational cost corresponds to the particles which have negligible contribution to the approximation of the posterior density. This phenomenon is called the *degeneracy problem*.

The degeneracy phenomenon is inevitable in the SIS particle filter because the variance of the particle weights increases by time. In order to measure the severity of this problem, the effective sample size N_{eff} has been defined as:

$$\hat{N}_{eff} = \frac{N_p}{1 + Var(w_k^{*i})} \tag{3.26}$$

where $w_k^{*i} = p(\mathbf{x}_k^i|\mathbf{z}_{1:k})/q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$. It can be observed from 3.26 that smaller N_{eff} implies more severity of the degeneracy problem. In practice, an estimate of N_{eff} which can be more easily calculated is used instead, as:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p}(\tilde{w}_k^i)^2} \tag{3.27}$$

where $\tilde{w}_k^i = w_k^i / \sum_{i=1}^{N_p} w_k^i$ is the normalized weight. In order to address the degeneracy problem, one may think of using a large number of particles. However, this is usually impractical. The most common solution is to perform an additional step called *resampling* in the particle filter algorithm.

3.4.2 Sequential Importance Sampling Particle Filter with Resampling

The resampling process replaces small weight particles with high weight, without changing the number of particles. For this purpose, we first construct

the cumulative distribution function (CDF) of the particles as:

$$\begin{aligned} F(\mathbf{x}_k^i) &= Pr(\mathbf{x}_k \leq \mathbf{x}_k^i | \mathbf{z}_{1:k}) \\ &= \int_{-\infty}^{\mathbf{x}_k^i} p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k \\ &\approx \int_{-\infty}^{\mathbf{x}_k^i} \sum_{j=1}^{N_p} w_k^j \delta(\mathbf{x}_k - \mathbf{x}_k^j) d\mathbf{x}_k \\ &= \sum_{j=1}^{i} w_k^j \end{aligned} \tag{3.28}$$

Next, we take N_p samples from a uniform distribution from 0 to 1, i.e., $u_i \sim \mathcal{U}[0,1], i = 1, \cdots, N_p$. Now, imagine the CDF function F in (3.28) to be a mapping from particle indexes $i = 1, \cdots, N_p$ to their corresponding CDF values. Therefore, for each u_i as a CDF value, we can find the nearest particle index i^* from $F^{-1}(u_i)$. We then replace the old particle $\mathbf{x}_k^i$ with the new particle $\mathbf{x}_k^{i*}$. In this way, low probability particles are replaced by higher probability particles. Since the u_i values have been taken from a uniform distribution, all new particles have equal probabilities. Therefore, after resampling, the particle weights should be all replaced by $w_k^i = 1/N_p, i = 1, \cdots, N_p$.

Using above resampling technique in the SIS particle filter, whenever a degeneracy problem is observed, i.e., when $\hat{N}_{eff}$ is smaller than some threshold N_T, results in Algorithm 2.

Algorithm 2 SIS Particle Filter with Resampling

1: **for** $i = 1 \rightarrow N_p$ **do**
2: Draw particle $\mathbf{x}_k^i$ from $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
3: Update the particle weight by $w_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}$
4: **end for**
5: **for** $i = 1 \rightarrow N_p$ **do**
6: Normalize the weight by $\tilde{w}_k^i = w_k^i / \sum_{j=1}^{N_p} w_k^j$
7: **end for**
8: Calculate the effective sample size by $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_p} (\tilde{w}_k^i)^2}$
9: **if** $\hat{N}_{eff} < N_T$ **then**
10: Do resampling
11: **end if**

However, the resampling process may cause another problem called *sample impoverishment*. This problem arises from the fact that particles with

high weights are often selected repeatedly which results in a less diversity in the new set of particles.

3.4.3 Sampling Importance Resampling Particle Filter

The sampling importance resampling (SIR) particle filter is the most commonly used type of particle filtering due to its algorithm simplicity. Making two assumptions converts the SIS particle filter with resampling into the SIR particle filter. First, the predictive transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ is chosen as the importance density $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$. Second, the resampling process is performed at every iteration, without evaluating the amount of degeneracy by $\hat{N}_{eff}$.

For the first assumption, we need to be able to take samples from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. Suppose that we can generate samples from the process noise $\mathbf{v}_{k-1}$ denoted by $\mathbf{v}_{k-1}^i$. Then, from (3.1), we can consider the particles as $\mathbf{x}_k^i = \mathbf{f}_k(\mathbf{x}_{k-1}^i, \mathbf{v}_{k-1}^i))$.

By this choice of importance density and resampling at every iteration, the particle weights can be simply calculated by:

$$w_k^i \propto p(\mathbf{z}_k|\mathbf{x}_k^i) \tag{3.29}$$

The SIR particle filter is described by Algorithm 3. Although the SIR particle filter has the advantage of a simple importance density choice and a simple weight update equation, it may provide a low performance. The reason is that the samples are taken from an importance density which is not relied on the observation $\mathbf{z}_k$. Moreover, resampling at every time index results in a very low diversity among particles. Nonetheless, the SIR particle filter is still prevalent in non linear tracking problems.

Algorithm 3 SIR Particle Filter

1: **for** $i = 1 \rightarrow N_p$ **do**
2: Draw particle $\mathbf{x}_k^i$ from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$
3: Calculate the particle weight by $w_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i)$
4: **end for**
5: **for** $i = 1 \rightarrow N_p$ **do**
6: Normalize the weight by $\tilde{w}_k^i = w_k^i / \sum_{j=1}^{N_p} w_k^j$
7: **end for**
8: Do resampling

3.5 Conclusion

In this chapter, we introduced the Bayesian filter as an optimal solution to the tracking problem of general non linear systems with embedded non Gaussian noises. We reviewed that for linear Gaussian systems, the general Bayesian filter simplifies to the well known Kalman filer. However, in the non linear non Gaussian systems, the integrals involved in the Bayesian filtering can not be solved without some numerical approximations. We focused on the grid-based method and particle filters as the two common approximation of Bayesian filters.

In the next few chapters, we particularly focus on the SIR particle filter for target tracking in UWB multistatic radars. We first derive the particle weights for single target tracking in such systems. Then, we extend the results to a multiple target case. We also compare the particle filter using the derived weights with the grid-based method.

Chapter 4

Single Target Tracking with Particle Filter for UWB Multistatic Radars

Passive localization, which deals with positioning of non-collaborating targets, using multistatic UWB radars is the subject of several works such as [21, 22, 23, 24, 25, 26]. However, only a few of them focus on tracking aspects in this context. For multistatic UWB radars, the measurement equation is no longer linear and also the measurement noise cannot be assumed as additive Gaussian, when estimating the target position. Therefore, the commonly used tracking filters based on linearity and Gaussianity assumptions, such as the Kalman filter, are not expected to exhibit a satisfactory performance, although they may still represent a valuable choice due to their low complexities [27].

Good candidates for UWB-based radar systems are non-linear Bayesian filtering algorithms such as particle filters [32, 33]. It is claimed in [34] that, while such Bayesian nonlinear filters constitute an optimal solution, their implementation is problematic due to their high computational complexity. A new tracking algorithm based on a threshold filter is then proposed. A related tracking approach was proposed in [35]. In [36], both Kalman and Bayesian filtering have been considered, in the framework of a pixel-based approach. Although exhibiting a good performance, this pixel-based approach has also some drawbacks. Firstly, it limits the position accuracy to the pixel size. Secondly, it entails high computational complexity and memory requirements. The complexity grows dramatically with the number of state dimensions, size of the area and number of targets. In [37], an algorithm based on particle filtering for UWB ranging and tracking is proposed. It demonstrates the advantages of particle filtering compared to conventional

tracking methods. However, this algorithm is tailored for bistatic radars and small distance ranging.

In this chapter, a multistatic UWB radar based on one transmitter and multiple receivers, as described in Chapter 2 is considered, employing particle filtering for target tracking. We assume a continuous state space, focusing only on particles wherein the target is more likely to be present. At each scan time, the particles are predicted according to a dynamic model and then evaluated by assigning a weight to each of them based on the observations. The final estimated position is obtained as the weighted sum of all particles. In the process, we derive an expression for weights calculation based on observations from all receivers.

The conventional particle filtering may lead to track divergence due to intrinsic peculiarities of UWB multistatic radars. Mainly, these divergences are caused by blind zones, i.e., regions inside which the target cannot be detected by at least one of the receivers (detailed in Section 2.2). The blind zone problem also affects Kalman filtering, that may even lose the track if the target remains hidden long enough [38]. Around the blind zones where the observations are not reliable for a considerable number of scan times, any strong clutter residue, a target manoeuver or an extremely low signal-to-noise ratio can corrupt the velocity estimation and cause a divergence. The tracking is also prone to be diverged in far areas from the transmitter and the receiver antenna, because UWB signals are highly attenuated by distance.

To cope with the blind zone problem, we propose a low-complexity modified particle filter, to improve the velocity estimation. In particular, we calculate the average velocity over a sliding window. Within each window, we assign a weight to each estimated velocity according to its difference from the average velocity calculated in the previous scan time. The higher the difference, the lower its contribution to the mean. We also propose to increase the process noise standard deviation proportionally to this error to cover any potential manoeuver. We show by numerical simulation that the proposed algorithm provides high estimation accuracy even at low signal-to-noise ratios in the presence of either static or dynamic clutter. Moreover, it can track complicated manoeuvering target trajectories. A further advantage of our algorithm is that the tracking filter does not require the measured target positions as input. The detection and localization steps at every scan time can be skipped for a single target tracking except for the first and second scan times to initialize the track. As mentioned before, while particle filtering has been accepted as the most suitable tracking filter in UWB radars which are characterized by non-linearity and non-Gaussianity, its usage has been avoided because of the illusion that its computational complexity is too

high. In fact, particle filters are algorithmically able to be implemented real-time [32]. Finally, we remark that the proposed algorithm based on particle filtering can be implemented for real-time tracking.

The chapter is organized as follows. In Section 4.1, equations for particle weights are derived. A tracking algorithm based on particle filtering with the derived weights is explained in Section 4.2. In Section 4.3, a modified tracking algorithm is presented to solve divergence problems caused by blind zones. In Section 4.4, numerical results are illustrated and the performance of our proposed particle algorithms is compared to that of Kalman filtering in various scenarios. Conclusions are provided in Section 4.5.

4.1 Particle Weight Derivation

We start by considering a single pair of transmit and receive antennas. Let us assume that $r_j(t)$ is the received signal at the j'th receiver, obtained by averaging the N_s received waveforms corresponding to the N_s pulses transmitted within a frame. In particle filtering the particles are taken on the target state space [32]. We assume that the presence of target has already been detected by the system. To simplify the derivation of particle weights we consider that the received pulse from direct path and the clutter are removed completely.[1] Therefore, the received waveform is[2]

$$r_j(t) = a_j p(t - \tau_j) + n_j(t). \tag{4.1}$$

In the above equation, $p(t)$ is the UWB reference pulse,[3] τ_j is the delay of the pulse scattered by the target and $n_j(t)$ is the additive white Gaussian noise. Note that channel gain a_j is a real positive or negative value and we assume it to be constant over one frame.[4]

After sampling the received signal, we denote the vector of N samples for each frame corresponding to the j'th receiver by $\mathbf{r}^j = \left[r_1^j \cdots r_N^j\right]^T$, where:

$$r_i^j = a_j p_{i-k_j} + n_i^j, \quad i = 1, \ldots, N \tag{4.2}$$

[1]Nevertheless, the numerical results presented in Section 4.4 will account for non-ideal clutter removal to assess the performance of the proposed algorithm in a realistic setup.

[2]Since we consider UWB signals, equivalent baseband notation is not adopted and all signals are real.

[3]It is the transmitted pulse, but it may also include antenna and channel propagation distortions.

[4]The radar cross section of the target depends highly on its orientation and characteristics and also the frequency [39],[40]. However, considering complicated models makes mathematical derivations more complicated and introduces additional computational complexity.

with $k_j = [\tau_j/T_s]$, where $[\cdot]$ stands for rounding toward the closest integer and T_s is the sampling period.

In Bayesian tracking, we need to know the likelihood $p(\mathbf{r}|\mathbf{x})$ for the received vector $\mathbf{r}$ and each of the particles from state space $\mathbf{x}$. The state space $\mathbf{x}$, as we define precisely later for each of our particle algorithms, may include target position components with or without target velocity components. If the target state space $\mathbf{x}$ is known, it means that we know the corresponding delay τ_j of the target echo. According to (4.2), p_{i-k_j}, $i = 1, \ldots, N$, is known and we can estimate the amplitude a_j by means of some estimation approach. For now, assume that we already have the estimation $\hat{a}_j$ for a_j. From (4.2) it can be seen that each r_i^j is a Gaussian distributed random variable with mean equal to $\hat{a}_j p_{i-k_j}$ and variance equal to the received noise variance σ^2, that is, $r_i^j \sim \mathcal{N}(\hat{a}_j p_{i-k_j}, \sigma^2)$, $i = 1, \ldots, N$. Given the statistical independence of noise samples we have:

$$p(\mathbf{r}^j|\mathbf{x}) \propto \exp\left\{-\frac{\sum_{i=1}^{N}(r_i^j - \hat{a}_j p_{i-k_j})^2}{2\sigma^2}\right\}. \tag{4.3}$$

Note that in the above equation, k_j is a function of $\mathbf{x}$. This dependency is not explicitly shown for the sake of simplicity.

In this work, we adopt the maximum likelihood (ML) approach for estimating the echo amplitude, i.e., we determine the a_j which maximizes the likelihood function $p(\mathbf{r}^j|\mathbf{x}, a_j)$. After simple calculations, it turns out that the ML estimation for a_j is

$$\hat{a}_j^{ML} = \frac{\sum_{i=1}^{N} r_i^j p_{i-k_j}}{\sum_{i=1}^{N} p_{i-k_j}^2}. \tag{4.4}$$

By replacing (4.4) in (4.3) we have:

$$p(\mathbf{r}^j|\mathbf{x}) \propto \exp\left\{\frac{\left(\sum_{i=1}^{N} r_i^j p_{i-k_j}\right)^2}{2\sigma^2 \sum_{i=1}^{N} p_{i-k_j}^2}\right\}. \tag{4.5}$$

We can see in the above equation that $\sum_{i=1}^{N} r_i^j p_{i-k_j}$ is equal to k_j'th sample of the vector obtained by cross-correlating the received vector $\mathbf{r}^j$ with the reference pulse vector $\mathbf{p} = \begin{bmatrix} p_1 \cdots p_N \end{bmatrix}^T$. Let us denote this cross-correlated received vector by $\mathbf{z}^j = \begin{bmatrix} z_1^j \cdots z_N^j \end{bmatrix}^T$. Moreover, the expression $\sum_{i=1}^{N} p_{i-k_j}^2$ in the above equation is in fact the energy of the reference pulse vector $\mathbf{p}$, that we denote by E_p. We obtain

$$p(\mathbf{r}^j|\mathbf{x}) \propto \exp\left\{\frac{(z_{k_j}^j)^2}{2\sigma^2 E_p}\right\}. \tag{4.6}$$

We now generalize the likelihood $p(\mathbf{r}|\mathbf{x})$ to the multistatic case with a single TX and N_R RX antennas. Since the received vectors $\mathbf{r}^1$, $\mathbf{r}^2$, ..., $\mathbf{r}^{N_R}$ are statistically independent, we have

$$\begin{aligned} p(\mathbf{r}|\mathbf{x}) \triangleq p(\mathbf{r}^1, \mathbf{r}^2, \cdots, \mathbf{r}^{N_R}|\mathbf{x}) &= \prod_{j=1}^{N_R} p(\mathbf{r}^j|\mathbf{x}) \\ &\propto \prod_{j=1}^{N_R} \exp\left\{\frac{(z_{k_j}^j)^2}{2\sigma^2 E_p}\right\} = \exp\left\{\frac{\sum_{j=1}^{N_R}(z_{k_j}^j)^2}{2\sigma^2 E_p}\right\}. \end{aligned} \tag{4.7}$$

The likelihood $p(\mathbf{r}|\mathbf{x})$ is used to calculate the particle weights during target tracking. In this work, we consider SIR particle filtering [32] in which resampling is done at each scan time. Therefore, the weight of particle i at the n'th scan is only proportional to the likelihood $p(\mathbf{r}|\mathbf{x}_n^i)$ and not dependent on its previous weight, where $\mathbf{x}_n^i$ denotes the state of the i'th particle at the n'th scan.

4.2 Tracking Algorithm

The flowchart of the proposed tracking algorithm is illustrated in Fig. 4.1. At every scan time (frame time), a central node collects the received vectors from all of the RX nodes and performs the tracking according to the following algorithm.

Assume that we have N_p particles. As shown in the figure, initial particles are taken from a normal distribution with mean equal to the estimated initial state $\hat{\mathbf{x}}_0$ and covariance matrix equal to the process noise covariance matrix $\mathbf{Q}$ (defined later), that is $\mathbf{x}_0^i \sim \mathcal{N}(\hat{\mathbf{x}_0}, \mathbf{Q})$. Then in the time update step, particles are predicted according to the dynamic model assumed for the target. In this work, we assume a straight-line constant-velocity motion model for the target according to:

$$\mathbf{x}_n = \mathbf{\Phi}\mathbf{x}_{n-1} + \boldsymbol{\nu}_{n-1} \tag{4.8}$$

where $\mathbf{x}_n$ is the target state at the n'th scan time. We assume that the state is a 4×1 vector containing the position and velocity components in x and y directions, i.e., $\mathbf{x}_n = \begin{bmatrix} x & v_x & y & v_y \end{bmatrix}^T$. In the above equation, $\mathbf{\Phi}$ is the state transition matrix related to the scan time duration (frame duration) T_F by:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T_F & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_F \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.9}$$

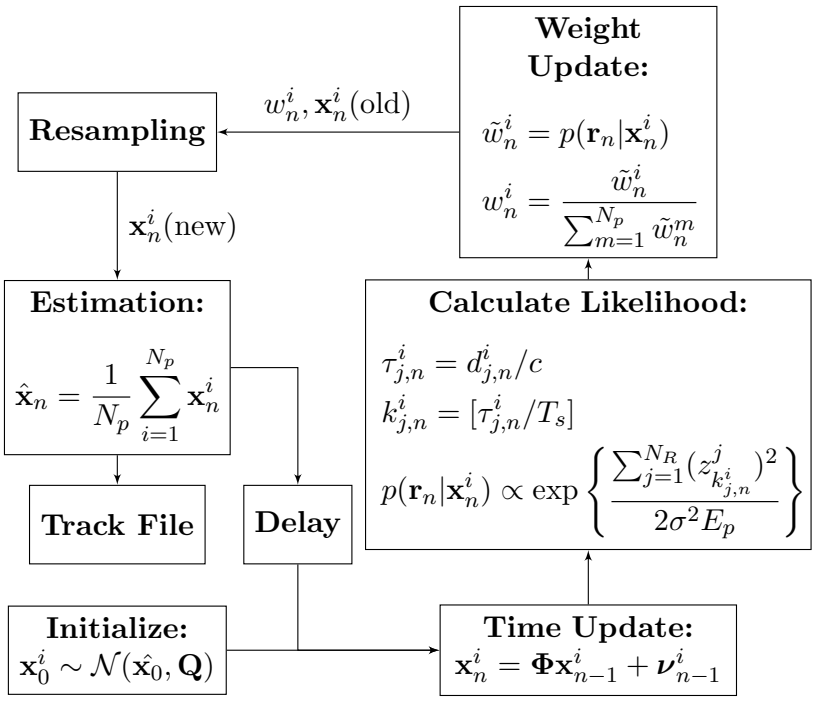


Figure 4.1: Flowchart of the proposed tracking algorithm with particle filtering.

Furthermore, $\boldsymbol{\nu}_n$ is the process noise taken from a normal distribution with zero mean and covariance matrix $\mathbf{Q}$ equal to

$$\mathbf{Q} = \sigma_a^2 \begin{bmatrix} \frac{T_F^4}{4} & \frac{T_F^3}{2} & 0 & 0 \\ \frac{T_F^3}{2} & T_F^2 & 0 & 0 \\ 0 & 0 & \frac{T_F^4}{4} & \frac{T_F^3}{2} \\ 0 & 0 & \frac{T_F^3}{2} & T_F^2 \end{bmatrix} \tag{4.10}$$

where σ_a^2 is the target acceleration noise variance [41].

Next, we need to calculate the likelihood function $p(\mathbf{r}_n|\mathbf{x}_n^i)$ for each particle i. For this purpose, we first calculate the delay of target echo $\tau_{j,n}^i$ with respect to each receive antenna j. This can be obtained by the sum of the particle distances from the transmitter and the j'th receiver, $d_{j,n}^i$, divided by the speed of light c. Indicating with (x_{TX}, y_{TX}) and (x_{RX}^j, y_{RX}^j) the coordinates of the TX and j'th RX antenna, respectively, and with $\boldsymbol{a}(i)$ the i'th

element of a generic vector $\boldsymbol{a}$, we have:

$$\begin{aligned}\tau_{j,n}^{i} &= d_{j,n}^{i}/c \\ &= \Big(\sqrt{(\mathbf{x}_n^i(1) - x_{TX})^2 + (\mathbf{x}_n^i(3) - y_{TX})^2} \\ &+ \sqrt{(\mathbf{x}_n^i(1) - x_{RX}^j)^2 + (\mathbf{x}_n^i(3) - y_{RX}^j)^2}\Big)/c.\end{aligned} \tag{4.11}$$

Then according to this delay which is equivalent to the $k_{j,n}^{i}$'th sample, we compute the exponential function for the likelihood using (4.7). Next in weight update step, we normalize this likelihood over all particles and assign it to w_n^i as the weight of particle i. In order to avoid degeneracy problem of particle filters, systematic resampling is done at each scan time that replaces low probability particles with high probability particles, keeping the number of particles constant [32]. After the resampling step, all new particles will have the same weights equal to $1/N_p$. Then, we calculate the estimated state $\hat{\mathbf{x}}_n$ by the weighted sum of all particles. Finally, the outputs are saved in a track file and the algorithm steps are repeated for the next scan of data.

4.3 Modified Tracking Algorithm

The previous tracking algorithm is a conventional SIR particle filtering in which the weights are derived for UWB multistatic radar applications. Although this approach is algorithmically straightforward, it suffers from the problem of instability. The tracking process may diverge due to strongly corrupted observations that persist for a period of time. This situation is particularly critical for UWB multistatic radar applications in which we have blind zones corresponding to each pair of TX and RX antennas.

It is well-known that particle filtering is powerful in blind zones due to the nonlinearity in state estimation. However, if we let particle filter estimate the velocity by extending the state space dimension and using the same weights for both position and velocity particles, consecutive deteriorations in velocity estimation cause moving the particles rapidly away from the true position and may lead to losing the track. Therefore, around the blind zones where the observations are poor for a considerable number of scan times, any strong clutter residue, a target manoeuver or an extremely low signal-to-noise ratio can corrupt the velocity estimation and cause a divergence. The tracking may also diverge in distant areas from the TX antenna, because UWB signals are highly attenuated by distance. Although this divergence occurs rarely and most of the time the algorithm performs well, it is important to fix the issue and make the algorithm more stable.

In our modified algorithm, we use particle filtering in a more controllable manner for estimating the velocity. As the dynamic model, we assume that the target average velocity is constant during a sliding window over time. Within each window, we give a weight to each estimated velocity according to its difference from the average velocity calculated in the previous scan time. The higher is the difference, the lower will be its contribution in calculating the mean. At the same time, we increase the process noise standard deviation proportional to this error to cover any potential manoeuver.

When the target is moving on a straight line and suddenly we observe that it deviates from its straight trajectory, there are two possibilities: either the target is still going on the straight line and we have corrupted observations or the observations are correct and the target has a manoeuver. In order to cover both possibilities, we assume the first hypothesis, that is the target is still going on straight line and we have a wrong observation. So, we allow that estimation less affects the average velocity. On the other hand, we increase the process noise standard deviation to avoid losing the track if the second hypothesis was true and the target had a manoeuver. Later, in Section 4.4, we show that even without increasing the process noise standard deviation according to errors, the algorithm is able to track extremely complicated trajectories. However, we consider it as a part of our algorithm to add more robustness and flexibility. In fact, our modified algorithm uses a particle filter for estimating the velocity in which the particles are the estimated velocities within the window.

We assume the following dynamic model for the target:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \bar{\mathbf{\Delta}}_{n-1} + \boldsymbol{\nu}_{n-1}. \tag{4.12}$$

Here the state space is a 2×1 vector containing only the position components in x and y directions, that is, $\mathbf{x}_n = \begin{bmatrix} x & y \end{bmatrix}^T$. In the above equation, $\bar{\mathbf{\Delta}}_n$ is a vector containing the mean of the movements for scan time n calculated independently for x and y directions, $\boldsymbol{\nu}_n$ is the process noise vector wherein each component is taken from a normal distribution with zero mean and standard deviation equal to the corresponding element of the vector $\boldsymbol{\sigma}_n$. Again note that in this chapter, we denote the i'th element of a generic vector $\boldsymbol{a}$ by $\boldsymbol{a}(i)$.

At each scan time, after estimating the position, we calculate the estimated movement as

$$\hat{\mathbf{\Delta}}_n = \hat{\mathbf{x}}_n - \hat{\mathbf{x}}_{n-1}. \tag{4.13}$$

If we denote the window size (as number of scan times) by ω, then the mean of the movements $\bar{\mathbf{\Delta}}_n$ defined by the expectation of $\mathbf{\Delta}_n = \mathbf{x}_n - \mathbf{x}_{n-1}$

conditioned on the previous movement means in the current window,

$$\bar{\boldsymbol{\Delta}}_n \triangleq \mathbb{E}\{\boldsymbol{\Delta}_n | \bar{\boldsymbol{\Delta}}_{n-1}, \bar{\boldsymbol{\Delta}}_{n-2}, \cdots, \bar{\boldsymbol{\Delta}}_{n-\omega}\}, \tag{4.14}$$

can be calculated similar to estimated state in a particle filter with ω particles equal to $\hat{\boldsymbol{\Delta}}_n$, $\hat{\boldsymbol{\Delta}}_{n-1}$, ..., $\hat{\boldsymbol{\Delta}}_{n-\omega+1}$ and the observations $\bar{\boldsymbol{\Delta}}_{n-1}$, $\bar{\boldsymbol{\Delta}}_{n-2}$, ..., $\bar{\boldsymbol{\Delta}}_{n-\omega}$. If we denote the weight of the particles by $\boldsymbol{\Delta}\mathbf{w}_n$, $\boldsymbol{\Delta}\mathbf{w}_{n-1}$, ..., $\boldsymbol{\Delta}\mathbf{w}_{n-\omega+1}$, the resulting equation will be:

$$\bar{\boldsymbol{\Delta}}_n(j) \approx \sum_{m=n-\omega+1}^{n} \boldsymbol{\Delta}\mathbf{w}_m(j)\hat{\boldsymbol{\Delta}}_m(j), \quad j = 1, 2 \tag{4.15}$$

where

$$\boldsymbol{\Delta}\mathbf{w}_n \propto \exp\left\{-\frac{\boldsymbol{\Delta}\mathbf{e}_n^2(j)}{2\boldsymbol{\sigma}_{n-1}^2(j)}\right\}, \quad j = 1, 2. \tag{4.16}$$

Here, $\boldsymbol{\Delta}\mathbf{e}_n$ is the error vector defined as $\boldsymbol{\Delta}\mathbf{e}_n \triangleq \hat{\boldsymbol{\Delta}}_n - \bar{\boldsymbol{\Delta}}_{n-1}$.

Now, we add more robustness to our algorithm for tracking high manoeuvering targets. For this purpose, assume that the target is moving on a straight line, then we observe that it deviates at scan time n. Therefore, according to (4.16) the estimated movement at scan time n will have a low weight in calculation of $\bar{\boldsymbol{\Delta}}_n$. Hence, $\bar{\boldsymbol{\Delta}}_n$ is approximately equal to $\bar{\boldsymbol{\Delta}}_{n-1}$. However, if the observed deviation at scan time n is caused by a target manoeuver, then the target movement at the next scan time will be approximately equal to its newly adopted movement $\hat{\boldsymbol{\Delta}}_n$. Therefore, we have to increase the process noise standard deviation vector by $\hat{\boldsymbol{\Delta}}_n - \bar{\boldsymbol{\Delta}}_{n-1} = \boldsymbol{\Delta}\mathbf{e}_n$ to avoid losing the track at scan time $n+1$. In practice, an increment by a fraction of the error is sufficient according to manoeuvrability of the target. Let's denote this coefficient by α. So, the process noise standard deviation vector at scan time n is calculated by:

$$\boldsymbol{\sigma}_n = \min\left\{\sigma_{max}, \sigma_p + \alpha|\boldsymbol{\Delta}\mathbf{e}_n|\right\} \tag{4.17}$$

where σ_p is the process noise standard deviation in each dimension for a non-manoeuvering target and σ_{max} is the maximum allowable process noise standard deviation in each direction to avoid that $\boldsymbol{\sigma}_n$ increases too much by large errors.

The flowchart of the modified algorithm is depicted in Fig. 4.2. Pseudo-code descriptions for *Initialize* and *Calculate Movement Mean and Process Noise Standard Deviation* blocks are given by Algorithms 4 and 5, respectvely. The blocks that are not described are the same as Fig. 4.1. For initialization, we assume that target positions at the first and second scan times are

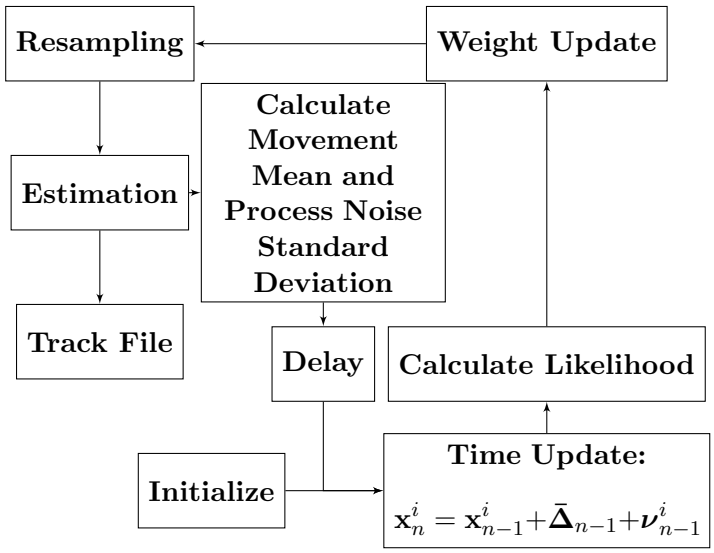


Figure 4.2: Flowchart of the modified tracking algorithm.

known by detection. Since we consider the initial particles as samples from a Gaussian distribution around the target position in the second scan time, having errors in detection can not affect the performance too much.

Since in the modified algorithm we have removed the velocities from the state space, the algorithm complexity reduces much more than what it is added by simple required calculations for the movement means.

Algorithm 4 Initialize

1: $\hat{\mathbf{\Delta}}_1 = \hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_0$
2: $\bar{\mathbf{\Delta}}_1 = \hat{\mathbf{\Delta}}_1$
3: $\mathbf{\Delta e}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
4: $\mathbf{\Delta w}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
5: $\boldsymbol{\sigma}_1 = \begin{bmatrix} \sigma_p \\ \sigma_p \end{bmatrix}$
6: $\mathbf{x}_1^i \sim \begin{bmatrix} \mathcal{N}(\hat{\mathbf{x}}_1(1), \sigma_p^2) \\ \mathcal{N}(\hat{\mathbf{x}}_1(2), \sigma_p^2) \end{bmatrix}$

Algorithm 5 Calculate Movement Mean and Process Noise Standard Deviation

$\hat{\boldsymbol{\Delta}}_n = \hat{\mathbf{x}}_n - \hat{\mathbf{x}}_{n-1}$

$\boldsymbol{\Delta}\mathbf{e}_n = \hat{\boldsymbol{\Delta}}_n - \bar{\boldsymbol{\Delta}}_{n-1}$

$$\boldsymbol{\Delta}\tilde{\mathbf{w}}_n = \begin{bmatrix} \exp\left\{-\dfrac{\boldsymbol{\Delta}\mathbf{e}_n^2(1)}{2\boldsymbol{\sigma}_{n-1}^2(1)}\right\} \\ \exp\left\{-\dfrac{\boldsymbol{\Delta}\mathbf{e}_n^2(2)}{2\boldsymbol{\sigma}_{n-1}^2(2)}\right\} \end{bmatrix}$$

if $n < \omega$ **then**

  for $m = 1 \rightarrow n$ **do**

    $$\boldsymbol{\Delta}\mathbf{w}_m = \begin{bmatrix} \boldsymbol{\Delta}\tilde{\mathbf{w}}_m(1) / \sum_{\acute{m}=1}^{n} \boldsymbol{\Delta}\tilde{\mathbf{w}}_{\acute{m}}(1) \\ \boldsymbol{\Delta}\tilde{\mathbf{w}}_m(2) / \sum_{\acute{m}=1}^{n} \boldsymbol{\Delta}\tilde{\mathbf{w}}_{\acute{m}}(2) \end{bmatrix}$$

  end for

  $$\bar{\boldsymbol{\Delta}}_n = \begin{bmatrix} \sum_{m=1}^{n} \boldsymbol{\Delta}\mathbf{w}_m(1)\hat{\boldsymbol{\Delta}}_m(1) \\ \sum_{m=1}^{n} \boldsymbol{\Delta}\mathbf{w}_m(2)\hat{\boldsymbol{\Delta}}_m(2) \end{bmatrix}$$

else

  for $m = n - \omega + 1 \rightarrow n$ **do**

    $$\boldsymbol{\Delta}\mathbf{w}_m = \begin{bmatrix} \boldsymbol{\Delta}\tilde{\mathbf{w}}_m(1) / \sum_{\acute{m}=n-\omega+1}^{n} \boldsymbol{\Delta}\tilde{\mathbf{w}}_{\acute{m}}(1) \\ \boldsymbol{\Delta}\tilde{\mathbf{w}}_m(2) / \sum_{\acute{m}=n-\omega+1}^{n} \boldsymbol{\Delta}\tilde{\mathbf{w}}_{\acute{m}}(2) \end{bmatrix}$$

  end for

  $$\bar{\boldsymbol{\Delta}}_n = \begin{bmatrix} \sum_{m=n-\omega+1}^{n} \boldsymbol{\Delta}\mathbf{w}_m(1)\hat{\boldsymbol{\Delta}}_m(1) \\ \sum_{m=n-\omega+1}^{n} \boldsymbol{\Delta}\mathbf{w}_m(2)\hat{\boldsymbol{\Delta}}_m(2) \end{bmatrix}$$

end if

$\boldsymbol{\sigma}_n = \min\left\{\sigma_{max}, \sigma_p + \alpha|\boldsymbol{\Delta}\mathbf{e}_n|\right\}$

4.4 Simulation Results

A square surveillance area of 100×100 meters is watched by a UWB radar sensor network composed of one transmitter and three receivers. Each TX and RX antenna is located at the middle of a square side. The origin of our assumed coordinate system is the lower left corner of the square. Therefore, the TX node is located at position $(0, 50)$, while the other 3 RX nodes are at positions $(50, 0)$, $(100, 50)$, and $(50, 100)$, respectively. Later, we will consider an example with more RX antennas.

The TX node emits Gaussian monocycles with duration parameter 1.4 ns and whose power spectral density is assumed to exceed the federal communications commission (FCC) mask by 10 dB. The number of pulses in each frame is $N_s = 134000$ and the frame duration is $T_F = 68.3$ ms, resulting in a pulse interval $T_{IP} = 510$ ns. The sampling frequency is 1.5 GHz. For each RX node, the receiver noise figure and the antenna temperature are $F = 6$ dB and $T_a = 290$ K, respectively. These settings result in a transmitted signal power of -32.5 dBW and a received noise power -86.2 dBW.

We consider a number of 100 pointwise objects to be present in the surveillance area as clutters. The clutters are distributed uniformly over the whole area. First, we consider static clutters with the same radar-cross-sections as that of the target. Later, we will consider moving clutters with different radar-cross-sections, drawn according to a Chi-squared distribution with two degrees of freedom (Swerling type I). Each RX node is assumed to implement a frame-to-frame clutter removal technique based on an infinite impulse response (IIR) filter. In particular, we have considered a high-pass first-order filter with transfer function $H(z) = (1 - z^{-1})/(1 - 0.9z^{-1})$ (hence with a pole equal to 0.9) which operates at a sampling frequency $1/T_F = 14.6$ Hz.

A single target with a radar-cross-section of $1\,\mathrm{m}^2$ is assumed to be present inside the area. The target is moving on a straight line starting from the origin of our coordinate system and with a constant velocity of 10 km/h, representing the speed of a human being walking quickly. Later, we will consider a more complicated trajectory.

At each RX antenna, the received signal is constructed as the superposition of the direct path, clutter echoes, and ground reflection in addition to the target echo and the noise. The echo amplitudes have been simulated according to $\sqrt{G_t G_r \lambda_0^2/((4\pi)^2 l^2)}$ for the LOS path and $\sqrt{G_t G_r \lambda_0^2 \sigma_o/((4\pi)^3 l_r^2 l_t^2)}$ for any object including the target [10], where G_t and G_r are the TX and RX antenna gains, respectively, which are both equal to 0 dB for our omnidirectional antennas and λ_0 is the wave length which has been calculated based on the center frequency 4.5 GHz. Moreover, l is the TX–RX distance, while l_t and l_r are the distances of the object from TX and from that RX,

respectively. Finally, σ_o is the radar-cross-section of the object.

In our numerical results, the system performance is measured in terms of both the cumulative distribution function (CDF) and root mean square (RMS) errors over 50 simulation runs, each with different noise and clutter realizations. For generating the RMS plot, we average the vector of estimation errors at all scan times over all simulation runs. For the CDF plot, we concatenate the estimation error vectors of all simulation runs and then we calculate the CDF for the resulting vector. Moreover, an example of a single simulation run showing the true and estimated target trajectories, antenna and clutter positions inside the area is shown for each scenario.

For each scenario, we evaluate the performance of the particle algorithms described in Fig. 4.1 and Fig. 4.2. The performance of conventional Kalman filtering is shown for comparison. Tracking by Kalman filter requires the measured target positions as inputs. We have adopted the pixel-based approach used in [36] to estimate these values. For all tracking techniques, the corresponding parameters have been set to the values maximizing the performance. For the first particle algorithm σ_a has been set to $8\,\mathrm{m/s^2}$. For the modified particle algorithm we have set the window size to 20. Also, α has been set to zero for all scenarios to show that even without increasing the process noise standard deviation according to errors, the algorithm performs well with both linear and complicated trajectories. Thus, the process noise standard deviation is always equal to σ_p in both x and y directions; we have chosen σ_p in a range between 0.04 m to 0.3 m based on the scenario. Note that in our simulations, we have considered a constant velocity (in amplitude) for the target over the whole trajectory. The role of α is more important for higher manoeuvers, for example when the target not only changes its direction but also introduces an acceleration. We have not considered such a scenario because of our particular surveillance application.

Since the complexity of particle filtering increases with the number of particles, this value should be selected carefully. In Fig. 4.3, we have evaluated the effect of the number of particles on the tracking performance of both particle algorithms in terms of CDF error. The assumed trajectory is a straight line with slope 50°. We can see from the figure that above a certain number of particles, the system accuracy improves very slowly. Thus, we set the number of particles equal to 200 for all our simulations, reaching a satisfying tradeoff between estimation accuracy and complexity. Furthermore, we can see that decreasing the number of particles has a lower impact on the performance of the modified particle algorithm compared to the first algorithm. In general, this would allow to use an even smaller number of particles, significantly reducing the computational complexity. However, we set this number to be 200 for both particle algorithms for comparison purposes.

To evaluate the algorithms behavior over the area, we have simulated a straight line target trajectory with slope 50°, passing through different blind zones. Fig. 4.4 shows the corresponding results. In the subfigures representing the surveillance area, the triangle indicate the TX and the three RX antennas are represented by squares. The gray ellipses represent the blind zones. The small circles over the area represent the static clutters and the three large circles in the middle of three LOS lines show the ground reflection points. It can be seen from the figures that the system performance improves considerably when using particle filtering compared to Kalman filtering, particularly around the blind zones. As we can see from the figures, in these scenarios the performance of the modified particle algorithm slightly improves compared to the first particle algorithm.

One of the advantages of particle filtering is in tracking targets with highly nonlinear trajectories. In Fig. 4.5 we have considered the case of a complex target trajectory. The target is moving on the curve with constant velocity 10 km/h. In this trajectory, we have included various shapes such as wave-like curves, circle-like curves, small and sharp closed curves, straight lines, as well as sudden direction changes with various angles between 0° and 180° (in which the target suddenly reverses its direction). It can be seen from the figure that the modified particle algorithm can track all maneuvers with high precision while Kalman filtering exhibits large estimation errors. Although the first particle algorithm can track the curve with a good precision for most of the simulation runs (in this regard the example reported in Fig. 4.5(a) is meaningful), there are a few realizations for which the tracking diverges. This behavior can be seen from the algorithm CDF error plot.

So far, the simulation results were for scenarios with static clutters having the same radar-cross-section as the target. Next, we consider dynamic clutters. Clutters are assumed to move with random velocities whose amplitudes and phases are drawn from uniform distributions over the ranges $[0, 1\,\text{km/h}]$ and $[0, 2\pi]$, respectively. The radar-cross-section of the clutters are drawn from a Chi-squared distribution with two degrees of freedom (Swerling type I) at the beginning of a simulation and kept constant during that simulation. The echoes due to slowly moving clutters (low frequency components) are attenuated according to the filter frequency response. In contrast, the echo from the target, which is moving with a higher velocity, remains almost unaffected. Fig. 4.6 shows the corresponding plots. In this figure, the size of the circles is proportional to the clutter radar-cross-sections. The slow random movements of the clutters have also been represented in the figure through small clouds over the area. We note that Kalman filtering frequently fails, while the first particle algorithm performs well most of the times. However, as it can be seen from its corresponding CDF error plot, in a few simulations

the algorithm diverges, mainly around the blind zones due to strong clutter residues. Note that for all algorithms we have used a simple IIR filter as the clutter removal technique.

A more complex processing can further reduce the amount of clutter residues and hence the number of divergences. In the modified particle algorithm, for dynamic clutters we have considered a directional process noise standard deviation in the direction of the estimated movement mean according to

$$\boldsymbol{\sigma}_n = \sigma_p \left[\frac{|\bar{\boldsymbol{\Delta}}_n(1)|}{\sqrt{\bar{\boldsymbol{\Delta}}_n^2(1) + \bar{\boldsymbol{\Delta}}_n^2(2)}}, \frac{|\bar{\boldsymbol{\Delta}}_n(2)|}{\sqrt{\bar{\boldsymbol{\Delta}}_n^2(1) + \bar{\boldsymbol{\Delta}}_n^2(2)}} \right]^T . \tag{4.18}$$

This approach reduces the amount of clutter residues affecting the performance compared to the case in which the process noise standard deviation is equal to σ_p in both x and y directions. With this replacement, the modified particle algorithm always performs well in the moving clutters even with a simple IIR filtering as the clutter removal technique.

Increasing the number of RX antennas provides more observations. Hence, it can help improve the tracking performance specially around the blind zones. To see that, we have considered six RX antennas located at positions $(33, 0)$, $(66, 0)$, $(100, 33)$, $(100, 66)$, $(33, 100)$ and $(66, 100)$, respectively. The assumed target trajectory is similar to the one in Fig. 4.5 and the clutters are static. The time settings have been changed slightly to $T_{IP} = 575\,\text{ns}$ and $T_F = 77\,\text{ms}$ to include all possible echoes within a pulse interval. This scenario has been shown in Fig. 4.7. As we can see, the performance of both particle algorithms improves greatly with the number of receivers so that no divergence occurs even for the first particle algorithm.

In all of the above scenarios, it is assumed that the transmitted power spectral density exceeds the FCC mask by 10 dB. In order to see the algorithms behavior in low signal-to-noise ratios, we have also considered a lower transmitted power spectral density fitting the FCC mask, that is, a transmitted signal power of $-42.5\,\text{dBW}$, the received noise power being unchanged. Fig. 4.8 shows the results with static clutters. We see that Kalman filtering always fails while both particle filters still perform well. The modified particle algorithm slightly outperforms the first particle algorithm for this scenario.

As it can be seen from the simulation results, the modified algorithm not only avoids track divergence in all scenarios, but also improves the estimation precision compared to the first particle algorithm. Another advantage of the modified particle algorithm is its high flexibility. Knowing some characteristics of the target can help improve the tracking performance even more by changing the parameter settings. For example, increasing the window

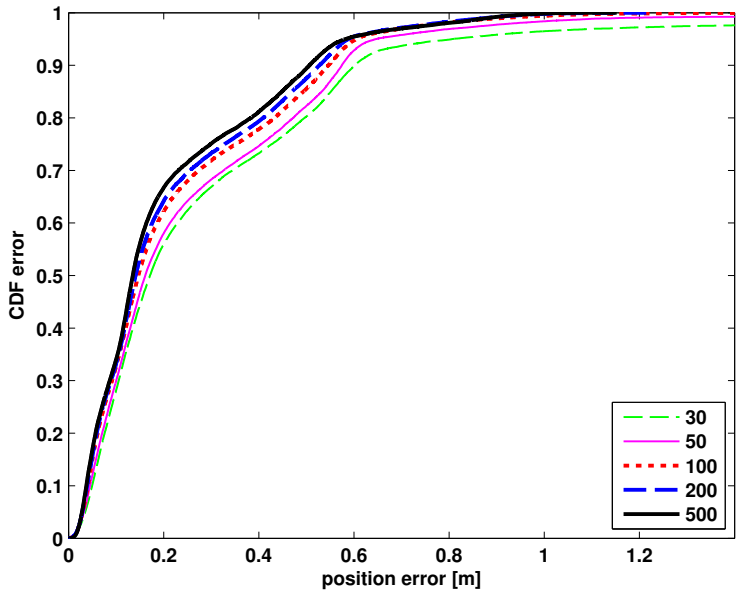
size improves the tracking performance for linear trajectories while decreasing it improves the manoeuvering ones. Similarly, increasing α improves the tracking performance for manoeuvering trajectories while decreasing it improve the linear ones. All above advantages are achieved while the algorithm complexity is reduced significantly.

As a final observation, we have measured the average simulation elapsed time for each data scan while the program was running on an Intel Core i5 CPU at 2.53 GHz with 4 GB RAM. For the first particle algorithm the result has been measured to be about 55 ms while for the modified particle algorithm it has been about 45 ms. Both amounts are less than the considered scan time duration (frame duration) which has been set to 68.3 ms. Thus, the real-time implementation feasibility is guaranteed.

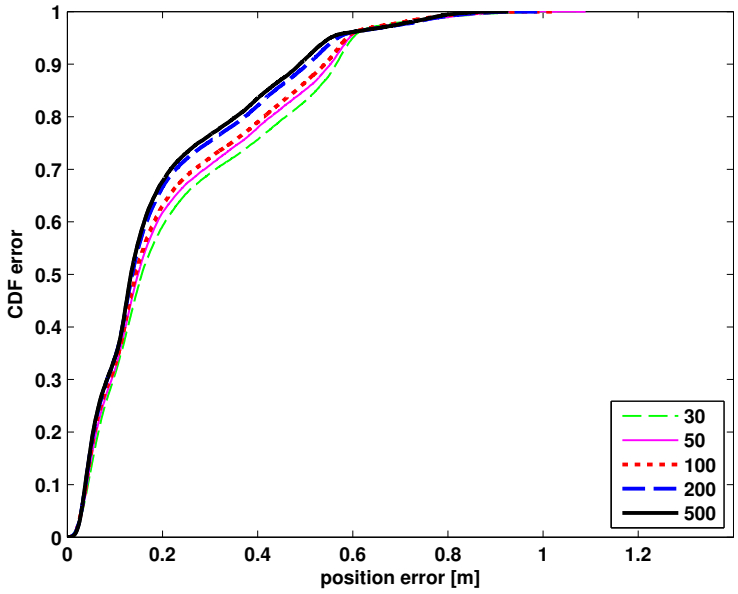
4.5 Conclusion

In this chapter, we proposed a tracking algorithm based on particle filtering for UWB multistatic radars. The particle weights were derived for one transmitter and multiple receivers, assuming a single target scenario. Then, a modified version of the algorithm was presented to solve some tracking problems, mainly caused by blind zones.

We illustrated by numerical results that the proposed algorithm can provide a high estimation accuracy, even at low signal-to-noise ratios, in the presence of either static or dynamic clutter. Moreover, the proposed algorithm is able to track even complicated manoeuvring target trajectories. The calculated computational time showed that the real-time implementation of the proposed tracking algorithm is feasible.



(a)



(b)

Figure 4.3: The effect of number of particles on the tracking performance for static clutter and a straight line trajectory. (a): Particle algorithm. (b): Modified particle algorithm.

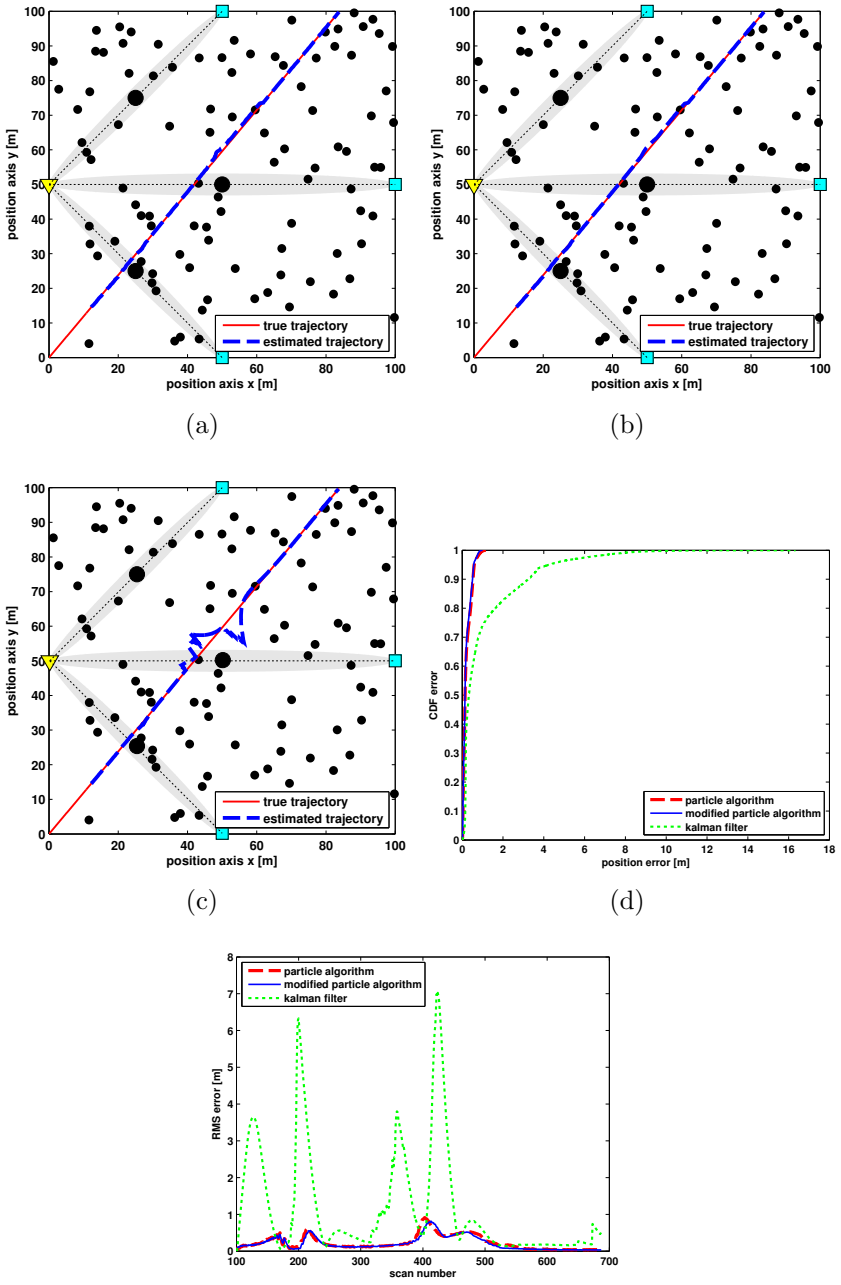


Figure 4.4: Performance comparison of the particle algorithms vs. Kalman filter for static clutter and a straight line trajectory. (a): An example of surveillance area for particle algorithm. (b): An example of surveillance area for modified particle algorithm. (c): An example of surveillance area for Kalman filter. (d): CDF error plots. (e): RMS error plots.

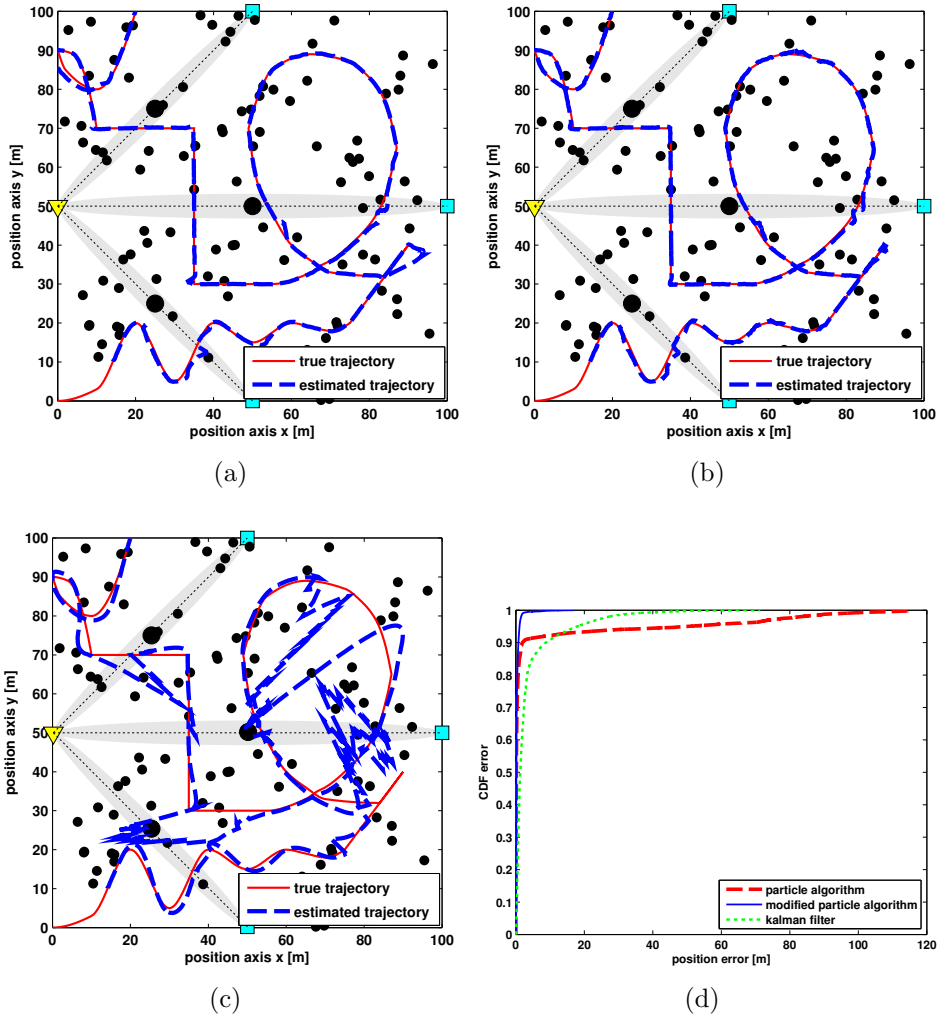


Figure 4.5: Performance comparison of the particle algorithms vs. Kalman filter for a static clutter and a maneuvering target. (a): An example of surveillance area for particle algorithm. (b): An example of surveillance area for modified particle algorithm. (c): An example of surveillance area for Kalman filter. (d): CDF error plots.

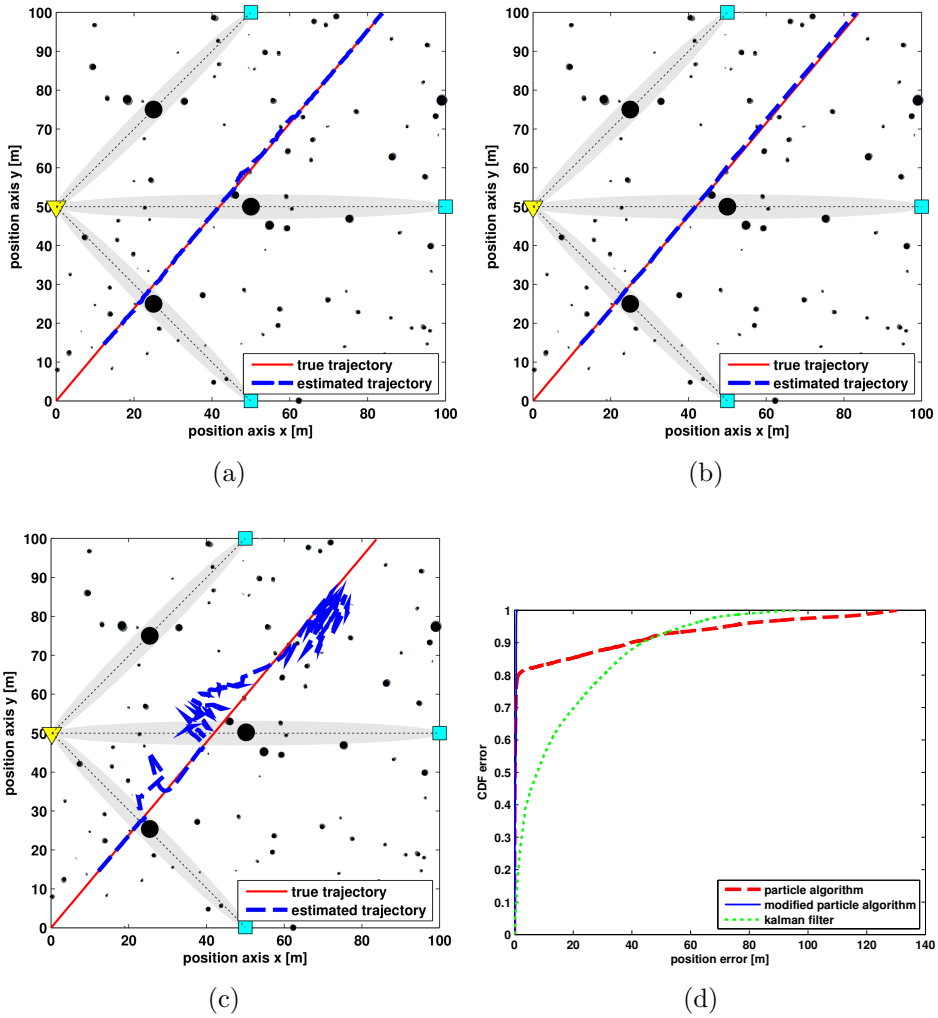


Figure 4.6: Performance comparison of the particle algorithms vs. Kalman filter for dynamic clutter and a straight line trajectory. (a): An example of surveillance area for particle algorithm. (b): An example of surveillance area for modified particle algorithm. (c): An example of surveillance area for Kalman filter. (d): CDF error plots.

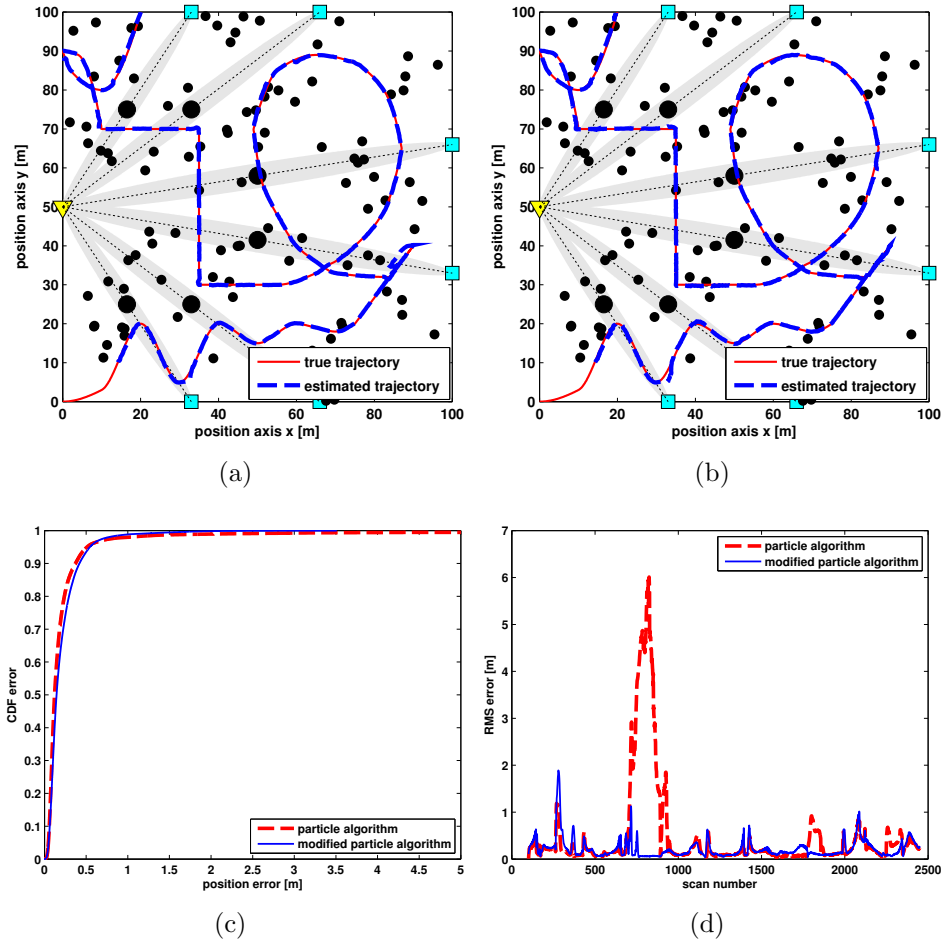


Figure 4.7: Performance comparison of the particle algorithms for static clutter and a complicated trajectory with 6 RX antennas. (a): An example of surveillance area for particle algorithm. (b): An example of surveillance area for modified particle algorithm. (c): CDF error plots. (d): RMS error plots.

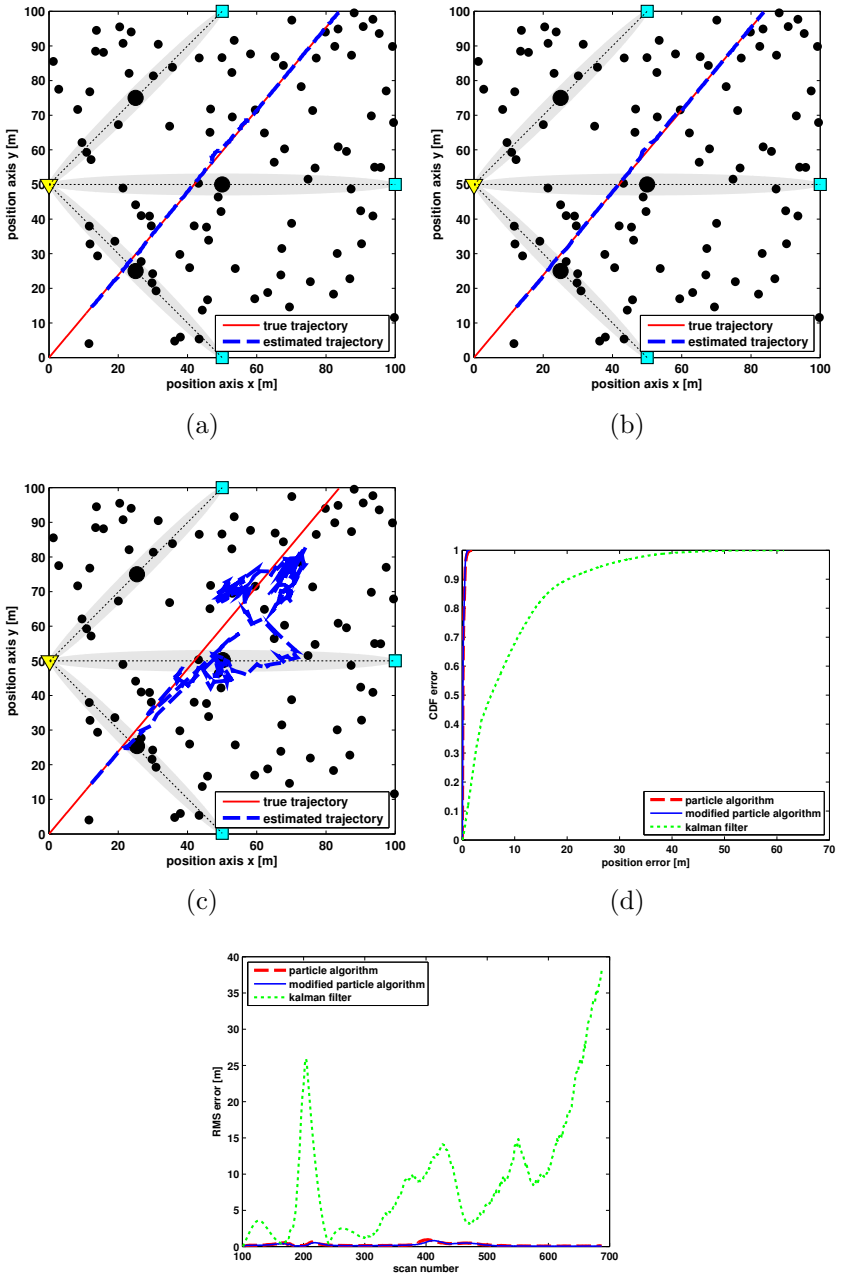


Figure 4.8: Performance comparison of the particle algorithms vs. Kalman filter for static clutter and a straight line trajectory with the power spectral density fitting the FCC mask. (a): An example of surveillance area for particle algorithm. (b): An example of surveillance area for modified particle algorithm. (c): An example of surveillance area for Kalman filter. (d): CDF error plots. (e): RMS error plots.

Chapter 5

Single Target Tracking with Grid-Based Filter for UWB Multistatic Radars

A good candidate for UWB-based radar systems is the non-linear Bayesian filtering [42]. There are many types of filters that fall in the category of general Bayesian filtering. However, the most widely used ones are the grid-based approach [42] and particle filtering [32].

In this chapter, we compare particle filtering with the grid-based Bayesian approach through numerical results and we show how consideration of only a subset of space in particle filtering and, on the other hand, discretization of the space in the grid-based Bayesian approach can affect the tracking performance [42]. Finally, we compare both approaches in terms of algorithm complexity.

The chapter is organized as follows. A grid-based Bayesian approach for a UWB multistatic radar system is explained in Section 5.1. In Section 5.2, numerical results are illustrated and the performance of the two tracking algorithms is compared for a manoeuvering target trajectory. Concluding remarks are provided in Section 5.3.

5.1 Grid-based Bayesian Tracking

Following the approach originally described in [36, 42], in this section we propose a grid-based Bayesian technique suitable for target tracking in UWB multistatic radars. In this approach, the surveillance area is divided into pixels. For each TX-RX pair, each pixel is associated with a specific time delay of the target-scattered pulse with respect to the direct path pulse, assuming

a target is present in that pixel. Therefore, after detecting the target within the surveillance area, estimating its position becomes a decision problem, in which we have to identify the point of the grid that, most probably, is occupied by the target.

In grid-based Bayesian tracking, we assume the dynamic model

$$\begin{cases} \mathbf{X}_n = \mathbf{X}_{n-1} + \mathbf{V}_{n-1}T_F + \mathbf{A}_n T_F^2/2 \\ \mathbf{V}_n = \mathbf{V}_{n-1} + \mathbf{A}_n T_F \end{cases} \tag{5.1}$$

where $\mathbf{X}_n$ and $\mathbf{V}_n$ represent the random target position and velocity vectors at the n'th scan time, respectively, and $\mathbf{A}_n$ is a random acceleration vector, having i.i.d. Gaussian components with zero mean and variance $\sigma_A^2/2$ per dimension.

According to estimation theory, the optimum decision rule, in the sense of maximizing the probability of correctly locating the target at the n'th scan time, estimates the position of the target as the value $\hat{\mathbf{x}}_n$ maximizing the a-posteriori metric

$$\hat{\mathbf{x}}_n = \arg\max_{\mathbf{x}_n \in \mathcal{X}} p(\mathbf{x}_n|\mathbf{r}_{1:n}) = \arg\max_{\mathbf{x}_n \in \mathcal{X}} \ln p(\mathbf{x}_n|\mathbf{r}_{1:n}), \tag{5.2}$$

where $\mathcal{X}$ is a predefined set of grid points and $\mathbf{r}_{1:n} = [\mathbf{r}_1 \dots \mathbf{r}_n]$. Equivalently, we can maximize the normalized version of the a-posteriori metric

$$\Lambda_n(\mathbf{x}_n) = \ln \frac{p(\mathbf{x}_n|\mathbf{r}_{1:n})p(\mathbf{r}_{1:n})}{p(\mathbf{r}_{1:n}|\mathcal{H}_0)}, \tag{5.3}$$

where $p(\mathbf{r}_{1:n}|\mathcal{H}_0)$ is the p.d.f. of the observables when there is no target in the area ($\mathcal{H}_0$). Based on the statistical model assumed for the observations, by successively applying the Bayes' rule, we obtain

$$\begin{aligned} \Lambda_n(\mathbf{x}_n) &= \ln \frac{p(\mathbf{r}_n|\mathbf{x}_n)}{p(\mathbf{r}_n|\mathcal{H}_0)} + \ln \frac{p(\mathbf{x}_n|\mathbf{r}_{1:n-1})p(\mathbf{r}_{1:n-1})}{p(\mathbf{r}_{1:n-1}|\mathcal{H}_0)} \\ &= \Lambda^L(\mathbf{x}_n) + \Lambda_n^A(\mathbf{x}_n), \end{aligned} \tag{5.4}$$

where the log-likelihood ratio $\Lambda^L(\mathbf{x}_n)$ and the logarithm of the normalized a-priori probability $\Lambda_n^A(\mathbf{x}_n)$ have been evidenced.[1]

For the likelihood term, since the waveforms at the receivers are conditionally independent, we can write

$$\Lambda^L(\mathbf{x}_n) = \sum_{j=1}^{N_R} \ln \frac{p(\mathbf{r}_n^j|\mathbf{x}_n)}{p(\mathbf{r}_n^j|\mathcal{H}_0)} = \sum_{j=1}^{N_R} \Lambda_j^L(\mathbf{x}_n). \tag{5.5}$$

[1]We notice that the term $\Lambda^L(\cdot)$ does not depend on index n, as we have assumed likelihood functions invariant with time.

This approach allows to separate the computation of $\Lambda^L(\mathbf{x}_n)$ in two steps. In the first step, each RX node calculates the partial metric $\Lambda_j^L(\mathbf{x}_n)$, while, in the second step, these terms are properly combined at the fusion center to obtain the soft area image constituting the input for the decision process or the tracking filter. In the next subsection, the metric to be used in (5.5) is briefly discussed. In Section 5.1.2 a tracking technique used to calculate the a-priori term $\Lambda_n^A(\mathbf{x}_n)$ based on Bayesian filtering is described.

5.1.1 Likelihood Ratio Metric

We consider a simple model where the target-scattered pulse, after the filtering stages, is equal to a local template at the receivers, with an unpredictable sign[2] and with different amplitudes at different RX nodes. In fact, for a given target position, the echoes received by the N_R RX nodes experience different path lengths, and therefore are characterized by a different energy. Consider now a generic pixel. Under the hypothesis that there is no target in the whole area ($\mathcal{H}_0$) the RX node receives only the thermal noise, while under the hypothesis "target present in $\mathbf{x}$" it receives both the scattered pulse (that is assumed to be an attenuated and delayed version of the reference pulse with an unpredictable sign) and thermal noise.

Thus, with reference to a generic pixel, the signal received by the jth RX node can be written as $r_{i,j} = n_{i,j}$ under $\mathcal{H}_0$ and as $r_{i,j} = \pm A_j p_{i-\tau_j} + n_{i,j}$ if the target is present in the pixel, for $i \in \{1, \ldots, N\}$. Here, we have indicated with $n_{i,j}$ the noise samples, with p_i the reference pulse samples, with $A_j = A_j(\mathbf{x}_n) > 0$ the amplitude of the scattered pulse received by the jth RX node, and with $\tau_j = \tau_j(\mathbf{x}_n)$ the delay, in samples, associated with the pixel in position $\mathbf{x}_n$ (related to its distance from the TX and the jth RX).

According to [36], assuming no information on the sign of the echo pulse is available, it is possible to express the log-likelihood map as:

$$\sum_{j=1}^{N_R} \Lambda_j^L(\mathbf{x}_n) = \sum_{j=1}^{N_R} \ln\cosh\left(\frac{A_j}{\sigma^2}\sum_{i=1}^{N} r_{ij} p_{i-\tau_j}\right) - \frac{E_p}{2\sigma^2}\sum_{j=1}^{N_R} A_j^2 \qquad (5.6)$$

where σ^2 is the variance of the thermal noise terms $n_{i,j}$ and $E_p = \sum_i p_i^2$.

As it can be noted from (5.6), the internal sums represent correlation terms which can be computed separately by each RX node, while their nonlinear combination is performed at the central node (see Fig. 2.1). Each correlation term concurring to form the decision metric is weighted on the amplitude of the corresponding scattered pulse received under the hypothesis

[2]Note that a sign change may be due, for example, to the target scattering.

of target present in the pixel. If reliable estimates of the amplitude of the target-scattered pulse are not available at the RX node, a sub-optimum test can be used, assuming $A_j = A$ for all $j \in \{1, \ldots, N_R\}$.

5.1.2 Bayesian tracking

The soft image of the area given by the values $\Lambda^L(\mathbf{x}_n)$ is used as input by the tracking filter that processes it to provide $\Lambda_n(\mathbf{x}_n)$ in (5.3). To this purpose, the a-priori term $\Lambda_n^A(\mathbf{x}_n)$ in (5.4) is calculated iteratively, based on the previous values $\Lambda_{n-1}(\mathbf{x}_{n-1})$. In fact, we have

$$\begin{aligned}
&\frac{p(\mathbf{x}_n|\mathbf{r}_{1:n-1})p(\mathbf{r}_{1:n-1})}{p(\mathbf{r}_{1:n-1}|\mathcal{H}_0)} \\
&= \sum_{\mathbf{x}_{n-1}\in\mathcal{X}} \sum_{\mathbf{v}_{n-1}\in\mathcal{V}} \frac{p(\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{v}_{n-1}|\mathbf{r}_{1:n-1})p(\mathbf{r}_{1:n-1})}{p(\mathbf{r}_{1:n-1}|\mathcal{H}_0)} \\
&= \sum_{\mathbf{x}_{n-1}\in\mathcal{X}} \sum_{\mathbf{v}_{n-1}\in\mathcal{V}} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{v}_{n-1}) p(\mathbf{v}_{n-1}|\mathbf{x}_{n-1}, \mathbf{r}_{1:n-1}) \\
&\qquad \times \underbrace{\frac{p(\mathbf{x}_{n-1}|\mathbf{r}_{1:n-1})p(\mathbf{r}_{1:n-1})}{p(\mathbf{r}_{1:n-1}|\mathcal{H}_0)}}_{=\exp(\Lambda_{n-1}(\mathbf{x}_{n-1}))}
\end{aligned} \tag{5.7}$$

where $\mathcal{V}$ is a finite set of predefined velocities and, based on (5.1), we have exploited the fact that $\mathbf{X}_n$ is conditionally independent of $\mathbf{R}_{1:n-1}$, namely $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{v}_{n-1}, \mathbf{r}_{1:n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{v}_{n-1})$. The distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{v}_{n-1})$ is jointly Gaussian, with mean $\mathbf{x}_{n-1}+\mathbf{v}_{n-1}\cdot T_F$ and variance $\sigma_A^2 \cdot T_F^4/8$ (on each component). Thus, we note from (5.7) that $\Lambda_n^A(\mathbf{x}_n)$ can be directly calculated if we are able to estimate the conditional distribution $p(\mathbf{v}_{n-1}|\mathbf{x}_{n-1}, \mathbf{r}_{1:n-1})$. In this work, also the a-posteriori distribution of the velocity vector is evaluated only in correspondence of the points of a predefined "grid" $\mathcal{V}$ and its values are estimated through Bayesian filtering.

In order to reduce the complexity of our estimation technique, we make the further assumption that the random variables $\mathbf{X}_n$ and $\mathbf{V}_n$ are conditionally indipendent, i.e.,

$$p(\mathbf{v}_n, \mathbf{x}_n|\mathbf{r}_{1:n}) = p(\mathbf{x}_n|\mathbf{r}_{1:n})p(\mathbf{v}_n|\mathbf{r}_{1:n}). \tag{5.8}$$

Under this hypothesis, the a-posteriori p.d.f. of the target velocity to be used

in (5.7) can be calculated iteratively, as

$$\begin{aligned}
p(\mathbf{v}_n|\mathbf{x}_n, \mathbf{r}_{1:n}) &= p(\mathbf{v}_n|\mathbf{r}_{1:n}) \\
&\overset{(a)}{=} \frac{1}{p(\mathbf{r}_n|\mathbf{r}_{1:n-1})} \sum_{\mathbf{x}_{n-1}\in\mathcal{X}} \sum_{\mathbf{v}_{n-1}\in\mathcal{V}} p(\mathbf{r}_n|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{x}_{n-1}) \\
&\quad \times p(\mathbf{x}_{n-1}|\mathbf{r}_{1:n-1}) p(\mathbf{v}_n|\mathbf{v}_{n-1}) p(\mathbf{v}_{n-1}|\mathbf{r}_{1:n-1}) \\
&\overset{(b)}{=} K \sum_{\mathbf{x}_{n-1}\in\mathcal{X}} \sum_{\mathbf{v}_{n-1}\in\mathcal{V}} p(\mathbf{r}_n|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{x}_{n-1}) \\
&\quad \times e^{\Lambda_{n-1}(\mathbf{x}_{n-1})} p(\mathbf{v}_n|\mathbf{v}_{n-1}) p(\mathbf{v}_{n-1}|\mathbf{r}_{1:n-1}) \qquad (5.9)
\end{aligned}$$

where: in equality (a) we have exploited the fact that $p(\mathbf{r}_n|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{x}_{n-1}, \mathbf{r}_{1:n-1}) = p(\mathbf{r}_n|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{x}_{n-1})$ due to the statistical independence of the observations, $p(\mathbf{x}_{n-1}|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{r}_{1:n-1}) = p(\mathbf{x}_{n-1}|\mathbf{r}_{1:n-1})$ due to the assumption (5.8) and, finally, $p(\mathbf{v}_n|\mathbf{v}_{n-1}, \mathbf{r}_{1:n-1}) = p(\mathbf{v}_n|\mathbf{v}_{n-1})$; while in equality (b) we have introduced the multiplicative factor $K = p(\mathbf{r}_{1:n-1}|\mathcal{H}_0)/p(\mathbf{r}_{1:n})$ and evidenced the term $\Lambda_{n-1}(\mathbf{x}_{n-1})$, which was computed in the previous scan time. In particular, the distribution $p(\mathbf{v}_n|\mathbf{v}_{n-1})$ is jointly Gaussian, with mean $\mathbf{v}_{n-1}$ and variance $\sigma_A^2 T_F^2/2$ for each statistically independent component. Moreover, we have

$$p(\mathbf{r}_n|\mathbf{v}_n, \mathbf{v}_{n-1}, \mathbf{x}_{n-1}) = p(\mathbf{r}_n|\mathcal{H}_0) e^{\Lambda^L\left(\mathbf{x}_{n-1} + \frac{T_F}{2}(\mathbf{v}_n + \mathbf{v}_{n-1})\right)}.$$

As a final remark, we notice that the constant factor K in (5.9) may be neglected as it does not influence the maximization of $\Lambda_n(\mathbf{x}_n)$ for the target position estimation.

5.2 Simulation Results

A square surveillance area of 100×100 meters is watched by a UWB multistatic radar composed of one transmitter and three receivers. Each TX and RX antenna is located at the middle of a square side (see Fig. 2.1). The origin of our assumed coordinate system is the lower left corner of the square. Therefore, the TX node is located at position $(0, 50)$, while the other 3 RX nodes are at positions $(50, 0)$, $(100, 50)$ and $(50, 100)$, respectively.

The TX node emits Gaussian monocycles with duration parameter 1.4 ns and whose power spectral density is assumed to exceed the FCC mask by 10 dB.[3] The number of pulses in each frame is $N_s = 134000$, the frame

[3]This exceeding has been chosen in order to cover the large area assumed. The possibility to exceed the FCC mask even by 20 dB is under consideration in the EU for emergency applications.

duration is $T_F = 68.3\,\text{ms}$, and the sampling frequency is $1.5\,\text{GHz}$. For each RX node, the receiver noise figure and the antenna temperature are $F = 6\,\text{dB}$ and $T_a = 290\,\text{K}$, respectively.

We consider a number of 100 pointwise objects, representing the clutter, to be uniformly distributed over the whole surveillance area. We consider static clutters with the same radar-cross-section as that of the target, set to $1\,\text{m}^2$. Each RX node is assumed to implement a frame-to-frame clutter removal technique based on first order IIR filter with one pole equal to 0.9.

A single target is assumed to be present inside the area. The target is moving on a curve with a velocity changing from $5\,\text{km/h}$ to $20\,\text{km/h}$ in the amplitude. The changes are in such a manner that the velocity in x direction remains constant in the amplitude and the velocity in y direction changes according to the adopted trajectory. The scenario and the trajectory are shown in Fig. 5.1.

At each RX antenna, the received vector is constructed as the superposition of the direct path, clutter echoes, and ground reflection in addition to the target echo and the received noise.

Unlike the grid-based Bayesian approach, in particle filtering we do not verify the whole area for the presence of target. The subspace considered for verification is selected only based on the assumed dynamic model for the target motion. In order to see how our assumed dynamic model for particle algorithm performs for a non matching true trajectory, we have considered such a complicated scenario.

We have used the the particle algorithm described in Section 4.3. For this algorithm, we have set the window size set to 20 and the number of particles to 200. In Chapter 4, we have evaluated the effect of the number of particles on the tracking performance of particle filtering and we have discussed how these values provide a satisfying tradeoff between estimation accuracy and complexity. For the grid-based Bayesian approach, the area is divided into 20cm $\times$ 20cm square pixels.

In Fig. 5.1(a) and (b), an example of a single simulation run showing the true and estimated target trajectories, antenna and clutter positions inside the area are shown. In the same figure, the performance of particle filter tracking and grid-based Bayesian tracking are compared in terms of both CDF and RMS error over 10 simulation runs, each with different noise and clutter realizations.

From Fig. 5.1(c) we can see that, as expected, Bayesian tracking outperforms particle filtering, except at the beginning of the tracking process. The reason of this behaviour is that for initializing the particle algorithm, the target positions at the first and second scan are assumed to be perfectly known while the Bayesian approach makes this assumption only for the first

scan time. This means that with the particle solution we assumed a perfect knowledge of both the initial target position and velocity, while in the grid-based Bayesian approach only the initial position is given and the velocity is only estimated.

Anyway, discretizing the area affects the optimality of the Bayesian approach. In fact, in Fig. 5.1(d) the effect of discretization results in a less smoothed RMS error curve of grid-based Bayesian tracking with respect to particle filter tracking. This is because in the grid-based Bayesian approach, the position accuracy is limited to the pixel size. Such behavior is confirmed by Fig. 5.1(c) where the CDF of position error for the particle filter tracking is lower than that of the Bayesian tracking for small position errors.

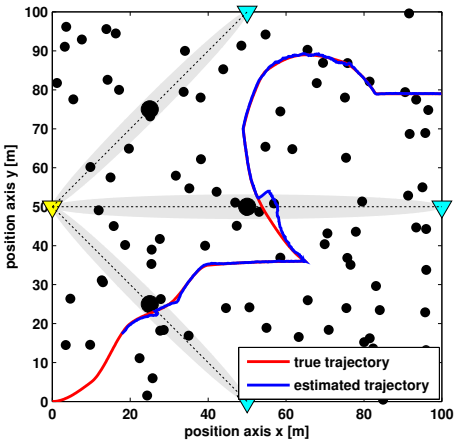
Another important observation is that particle filtering performs better than Bayesian approach for parts of the trajectory that better match the assumed dynamic model. For instance, in the final part of the trajectory the target moves on a straight line with constant velocity which perfectly matches the dynamic model. In this part, the RMS error is lower compared to Bayesian approach. However, the particle algorithm has a worse performance around blind zones where the observations are corrupted for a considerable number of scan times. This leads to prediction of particles in wrong positions and hence introduces errors as can be seen in Fig. 5.1(d) in the scan interval $400-550$. On the contrary, the grid-based Bayesian approach has full control over the area and therefore can provide performance robust to blind zones.

As a final remark, we compared the complexity of the two approaches based on the average simulation time for each scan while the program was running on an Intel Core i5 CPU at 2.53 GHz with 4 GB RAM. For particle filter tracking the time required to perform the task in each scan is about 45 ms, while for grid-based Bayesian approach it is 34 s.

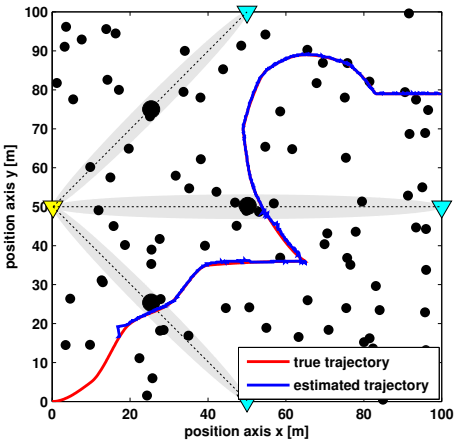
5.3 Conclusion

In this chapter, we introduced a grid-based Bayesian filter for tracking in UWB multistatic radars. We then compared the performance of this approach with the particle filering described in Section 4.3 by simulations in a realistic setting including noise, clutter and signal attenuation.

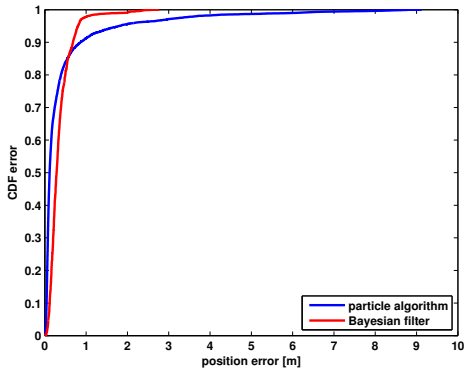
The numerical results showed that the grid-based tracking method outperforms the particle filtering, especially in proximity of blind zones, at the cost of a much higher computational complexity and memory requirements.



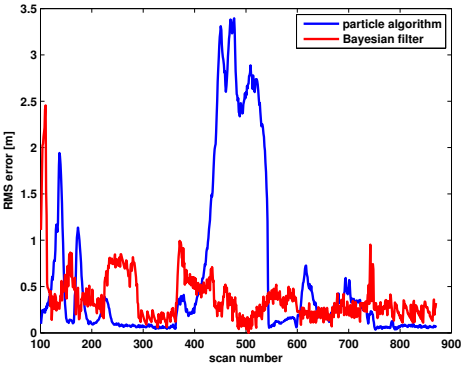
(a) An example of surveillance area for particle algorithm.



(b) An example of surveillance area for grid-based Bayesian approach.



(c) CDF error plots for particle and grid-based Bayesian algorithms.



(d) RMS error plots for particle and grid-based Bayesian algorithms.

Figure 5.1: Performance comparison of the particle algorithm vs. the grid-based Bayesian approach.

Chapter 6

Detection and Localization Techniques for UWB Multistatic Radars

In the last years, passive localization through multistatic UWB radars has been the subject of several works such as [22, 24, 25, 26] and [43]. In most of the works such as [10], [44] and [45], detection and localization are done using constant false alarm rate (CFAR) thresholding and direct method (also called tri-lateration), respectively. The target time of arrival (TOA) is estimated first by CFAR detector at each receiver and then the target location is calculated by finding the intersection of three ellipses corresponding to the target TOAs with respect to three receivers. In [36], a pixel-based detection and localization technique is proposed in which the whole surveillance area is divided into pixels and a soft metric related to the correlation of the received signal and the reference pulse is calculated for each pixel at every receiver. Then, a fusion node combines the soft images provided by all receivers. The pixel with the maximum combined metric determines the target location. This approach suffers from high complexity in calculating the soft metrics for every pixel. Moreover, in this work just a single target scenario is addressed.

In this chapter, we propose a pixel-based localization technique using the CFAR detector for multiple target cases. In this approach, the surveillance area is divided into pixels in a similar manner as in [36]. However, instead of calculating a soft metric for each pixel, the energy of the received sample at each receiver corresponding to that pixel is calculated and then the CFAR thresholding is performed. The final estimated position will be the pixel for which most of the receivers have experienced a threshold exceeding.

We compare our proposed pixel-based localization approach with the con-

ventional direct method of localization using the CFAR detector. Then, we propose to add a simple processing after the CFAR detector and before either of these localization approaches. This processing, which is based on a simplified median filtering done on the previous and current samples around the sample under verification, allows us to reduce the CFAR threshold to a very low value and hence to heavily reduce the number of missed detections. The performances of all above techniques are evaluated numerically in the case of multiple targets accounting for the spatial configuration of the receivers, propagation effects, presence of residual clutter, and noise.

This chapter is organized as follows. Section 6.1 explains the conventional detection and localization techniques. In Section 6.2, our proposed CFAR detection and pixel-based localization techniques are described. In Section 6.3, the performances of all techniques are compared numerically. Concluding remarks are provided in Section 6.4.

6.1 Traditional Detection and Localization Techniques

The aim of the detection stage is to decide if any target is present in the observed area, based on the received radar signals and to estimate the target TOAs at each receiver. Then, at the localization stage, the potential target positions are measured using estimated TOAs. Localization outputs are then passed to the data association step which is responsible for associating the measured positions to the target tracks.

6.1.1 CA-CFAR Detector

For the purpose of target detection by UWB radars, CFAR detectors have been widely used because of the robustness of this method demonstrated in experimental results (see, e.g., [46]). Besides conventional schemes, a number of modifications of the CFAR detector have been proposed, depending on the background noise and clutter models. Out of them, cell averaging CFAR (CA-CFAR) remains the simplest and most common version [47].

Let us assume that $r_{k,n}$ is the n'th sample of the received signal at one of the N_R receivers, obtained by averaging the N_s received signals corresponding to the N_s pulses transmitted within k'th frame. Therefore, for each frame time (scan time) k, we have a vector of N samples at any receiver which is passed to the CFAR detector.

Fig. 6.1 shows the processing done in the CA-CFAR detector for every sample $r_{k,n}$ at either of the UWB radar receivers. In CA-CFAR detector,

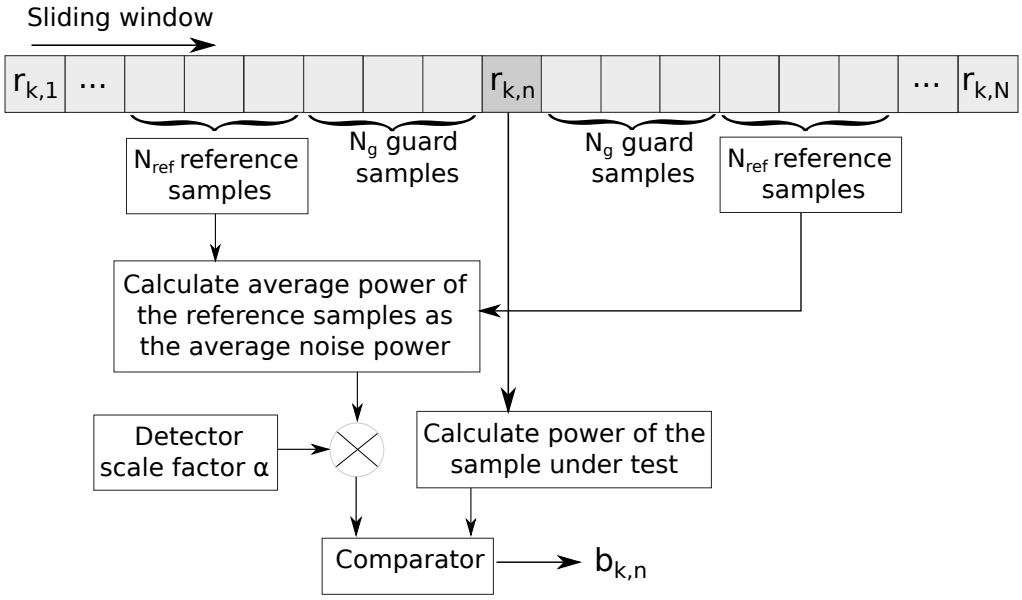


Figure 6.1: CA-CFAR detector.

the threshold level is calculated by estimating the level of the noise floor around the sample under test ($r_{k,n}$). This can be found by taking a block of N_{ref} samples around $r_{k,n}$ and calculating the average noise power level of these reference samples. To avoid corrupting this estimate with power from the sample under test itself, N_g samples immediately adjacent to $r_{k,n}$ are normally ignored and referred to as guard samples. In our case, we have assumed a guard band equal to the transmitted pulse width. A target is declared to be present in the $r_{k,n}$, if its instantaneous power is greater than the calculated local average power level which can be interpreted as a threshold. So, the output of the detector, $b_{k,n}$, is set to 1 if threshold is exceeded, otherwise it is set to 0. The threshold can be increased by a scale factor α depending on the scenario.

This thresholding often generates more than one TOA for each target because of the UWB pulse width.[1] In this work, close TOAs are grouped together in clusters. Then, the middle time in each cluster is considered as the detected TOA.

[1]Received pulse width can vary because of pulse distortions introduced by the channel and clutter removal filter.

6.1.2 Direct Method of Localization

Direct method of localization, also called tri-lateration, is the most conventional approach for measuring target positions in UWB radar sensor networks based on the estimated TOAs [48]. Assume that the transmitter and the j'th receiver are located at (x_t, y_t) and (x_j, y_j), respectively. The TOA of the target at receiver j has been estimated before by the detector as τ_j. The following equation holds assuming no detection errors:

$$\tau_j = \frac{1}{c}\left(\sqrt{(x-x_t)^2+(y-y_t)^2} + \sqrt{(x-x_j)^2+(y-y_j)^2}\right) \tag{6.1}$$

where c is the velocity of the electromagnetic wave propagation and (x, y) are the unknown coordinates of the target. For each triplet of receivers out of the N_R receivers, a system of three simultaneous equations in the form (6.1) is solved by least-squares calculation.

After doing direct calculation for all possible triplets of receivers and considering all possible combinations of TOAs detected by the selected receivers, we have a number of target locations (true or false) measured by each receiver combination. If a preset number of receiver combinations have generated solutions sufficiently close to each other, then those close solutions are kept, while the others are discarded. In this way, the number of false targets are considerably reduced. The final measured location will be obtained by average of all these close solutions over corresponding receiver combinations.

6.2 Proposed Detection and Localization Techniques

6.2.1 Proposed CA-CFAR Detector with Median Filtering

Reducing the threshold in CFAR detectors reduces the number of missed detections but on the other hand, it increases the false alarms. Our proposed CFAR detector makes it possible to set a very low threshold which highly reduces the number of missed detections, but still provides reasonable false alarm rate.

Obviously when setting a very low threshold, CFAR detector generates an intolerable number of false alarms. In the binary image showing the propagation time versus scan time constructed for each receiver, the false alarms appear as *salt and pepper* noises, a typical situation in which median filters turn out to be useful. In conventional median filtering employed in

image processing, each pixel of the image is replaced by the median value in the 3×3 neighborhood pixels around [49]. Since in radar processing we have no knowledge of the upcoming pixels, they are not considered in our median filter in order to allow real-time processing.[2] That is, for the pixel $b_{k,n}$ (the binary output of CFAR detector in Fig. 6.1), corresponding to scan time k and propagation time n, only the 3×2 neighborhood pixels $b_{k-1,n-1}$, $b_{k-1,n}$, $b_{k-1,n+1}$, $b_{k,n-1}$, $b_{k,n}$, $b_{k,n+1}$ are considered for median filtering. These neighborhood pixels are shown in Fig. 6.2. If the majority of these pixels are equal to 1, then the pixel $b_{k,n}$ is set to 1, otherwise it is set to 0, that is:

$$b_{k,n} = \begin{cases} 1 & \text{if } \sum_{i=k-1}^{k} \sum_{j=n-1}^{n+1} b_{i,j} > T_{med} \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

where T_{med} is a threshold chosen according to system parameters such as UWB transmitted pulse width and target minimum velocity. Note that the threshold $T_{med} = 3$ in (6.2) provides acceptable performance for our UWB radar settings in Section 6.3.

This simple image processing allows to considerably reduce the CFAR threshold without being worried about the false alarms. Fig. 6.3 shows an example of the binary image constructed by CFAR detector output before and after the median filtering stage at one of the receivers as obtained in the simulation scenario described in Section 6.3. We can see how setting a low CFAR threshold generates very large number of false alarms and how the median filtering is able to remove them. Note that because the transmitted pulse in IR-UWB radars occupies more than one pixel, median filter is not able to remove the target echoes, but only the false alarms.

6.2.2 Proposed Pixel-based Localization Technique

In the proposed pixel-based localization technique, the whole surveillance area is divided into pixels.[3] After performing CA-CFAR detection for every sample at each receiver j, the TX-pixel-RX time delay, τ_j, for a given pixel is calculated using (6.1), in which (x, y) is replaced by the coordinates of that pixel. The corresponding sample in the received signal will be the $[(\tau_j - \tau_{LOS})/T_s]$'th sample, where τ_{LOS} is the LOS time delay, T_s is the sampling time and [.] stands for nearest integer. Then, the output of CFAR detector is seen at this sample to verify if a threshold exceeding has happened.

[2]However, buffering and making delayed decision can be also used for better robustness.

[3]Note that the definition of pixel here is different from that of used in Section 6.2.1 for image processing.

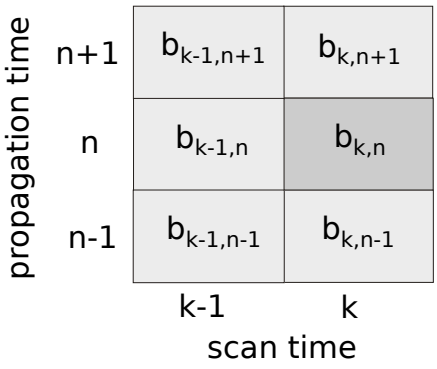


Figure 6.2: Neighborhood pixels for median filtering.

After doing the above processing for all pixels in the area, the final estimated positions will be the pixels for which most of the receivers have experienced CFAR threshold exceeding.

The output at this step will be usually more than one estimated location for each target. Clustering is required to assign close points to each other. In this work, we cluster all points in groups of points based on their relative distances. That is, a point falls into a cluster if its distance from the mean of the points currently assigned to that cluster is lower than a predefined value. The average positions of each cluster are considered as the final estimated locations.

6.3 Simulation Results

In our numerical simulations, we consider a square surveillance area of 100×100 meters is watched by a UWB radar sensor network composed of one transmitter and six receivers located on the square sides. The origin of our assumed coordinate system is the lower left corner of the square. Therefore, the TX node is located at position $(0, 50)$, while the other 6 RX nodes are at positions $(33, 0)$, $(66, 0)$, $(100, 33)$, $(100, 66)$, $(33, 100)$ and $(66, 100)$, respectively.

The TX node emits first derivative Gaussian monocycles with duration parameter $1.4ns$ and whose power spectral density is assumed to exceed the FCC mask by 10 dB.[4] The center frequency is 4.5 GHz and the transmitted

[4]This exceeding has been chosen to cover the large area assumed. The possibility to exceed the FCC mask even by 20 dB is under consideration in the EU for emergency applications.

signal bandwidth (at -10dB w.r.t. the maximum) is 500MHz. The number of pulses in each frame is $N_s = 114000$ and the frame duration is $T_F = 68.4ms$. The sampling frequency is set to 1.5 GHz. For each RX node, the receiver noise figure and the antenna temperature are set to $F = 6$ dB and $T_a = 290K$, respectively. These settings result in a transmitted signal power of -32.5 dBW and a received noise power -86.2 dBW.

We consider a number of 100 pointwise objects to be present in the surveillance area as clutters. The clutters are distributed uniformly over the whole surveillance area. We consider static clutters with $1m^2$ radar-cross-sections, the same as the target. Each RX node implements a frame-to-frame clutter removal technique based on a first-order high-pass IIR filter with one pole equal to 0.9 operating at a sampling frequency $1/T_F = 14.6$ Hz.

Three targets are assumed to be present inside the area. Each target is moving on a straight line trajectory with a constant velocity of $10km/h$, representing the speed of a human being walking quickly. Note that since detection and localization are performed scan by scan, independently, more complicated trajectories do not affect the performance.

At each RX antenna, the received signal is constructed as the superposition of the direct path, clutter echoes, and ground reflection in addition to the target echo and the noise. The channel gains have been simulated according to $\sqrt{G_t G_r \lambda_0^2/((4\pi)^2 l^2)}$ for the LOS and $\sqrt{G_t G_r \lambda_0^2 \sigma_o/((4\pi)^3 l_r^2 l_t^2)}$ for any object including the target [10], where G_t and G_r are the TX and RX antenna gains, respectively, which are both equal to 0 dB for our omnidirectional antennas and λ_0 is the wave length. Moreover, l is the TX–RX distance, while l_t and l_r are the distances of the object from TX and from that RX, respectively. Finally, σ_o is the radar-cross-section of the object.

This scenario is shown in Fig. 6.4. The triangle represents the transmitter and the squares are RX antenna positions. The gray ellipses are the blind zones corresponding to each pair of TX-RX antennas. The assumed trajectory of the three targets are shown in the figure for a duration of 600 scan times. Their starting positions are shown with circles.

In the CA-CFAR detector, $N_g = 20$ guard samples (equal to the transmitted pulse width) and $N_{ref} = 50$ reference samples are considered. The clustering distance is defined as $3m$.

Fig. 6.5 plots missed detection probability versus false alarm probability for proposed approaches compared to the conventional techniques. The axes are both in logarithmic scale. The plots have been developed by changing the value of CFAR detector scale factor α.

In order to calculate these probabilities, we consider a circular gate of radius $5m$ around the target true position and we assign the nearest estimated point to the target, if it falls in the gate. False alarms are defined

as all the remaining non assigned estimated points. A target is missed if no estimated point can be assigned to it. Then, the missed detection probability is calculated as the ratio of the number of missed targets to the total number of targets averaged over all scan times and all simulation runs with a given CFAR threshold, whereas the false alarm probability is defined as the average of the number of false alarms over all scan times and all simulation runs with that CFAR threshold. A total number of 50 simulation runs, each with a different noise and clutter realization has been considered for generating the plots.

The comparison shows that the pixel-based localization outperforms conventional direct method of localization in a general view. The performance of both localization techniques improves when employing median filter in CFAR detector. This improvement is more evident for direct localization compared to the pixel-based one. Altogether, the combination of pixel-based localization with median filter seems to provide the best tradeoff for false alarm probabilities higher than 10^{-1}, whereas for lower values direct localization with median filter is the preferable. Note that the performances shown in Fig. 6.5 (as well as in the subsequent Fig. 6.6) are obtained without applying any tracking algorithm as the focus of this work is on detection. It has been observed that, if target tracking is employed, the performances in Fig. 6.5 and Fig. 6.6 improve substantially, as expected, the relative performance between different schemes behaving in a similar way.

Fig. 6.6 shows the CDF localization error plots for all the above approaches. The CFAR detector scale factor for each technique has been chosen so that the best tradeoff between the missed detection probability and false alarm probability is achieved. Since the typical ranges of these probabilities are values lower than 10^{-1} (the grey region in Fig. 6.5), the best tradeoff is achieved almost at the points specified with bigger markers in Fig. 6.5. These points correspond to the CFAR detector scale factor, α equal to 10.6 for direct localization, 6.3 for pixel-based localization, 3 for direct localization with median filter and 3.5 for pixel-based localization with median filter. To generate each CDF error curve, we concatenate the estimation error vectors of all three targets over all simulation runs and then we calculate the CDF for the resulting vector.

Again, it can be seen from Fig. 6.6 that the pixel-based localization outperforms conventional direct method of localization. The localization error of both techniques improves by employing median filterig. In particular, the pixel-based localization with median filter seems to provide the lowest positioning error among all.

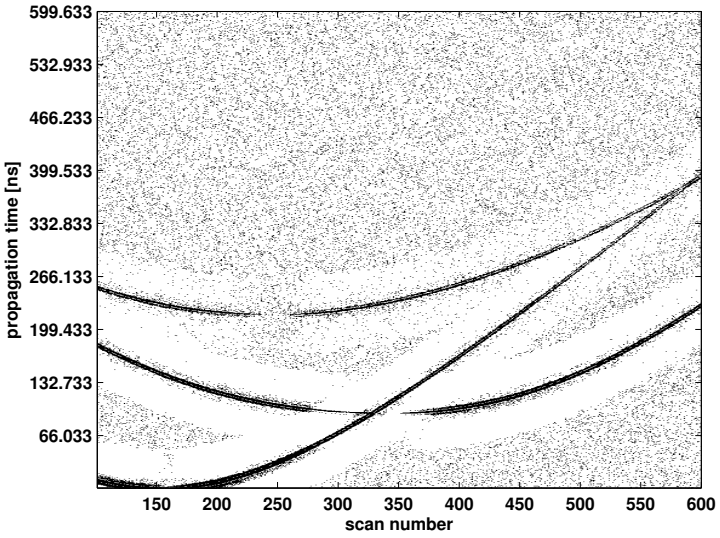
According to the above results, the proposed pixel-based localization technique with median filter can be quite appropriate for surveillance applications

of UWB multistatic radars in which the blind zones can hide the target for a period of time long enough to lose the track.

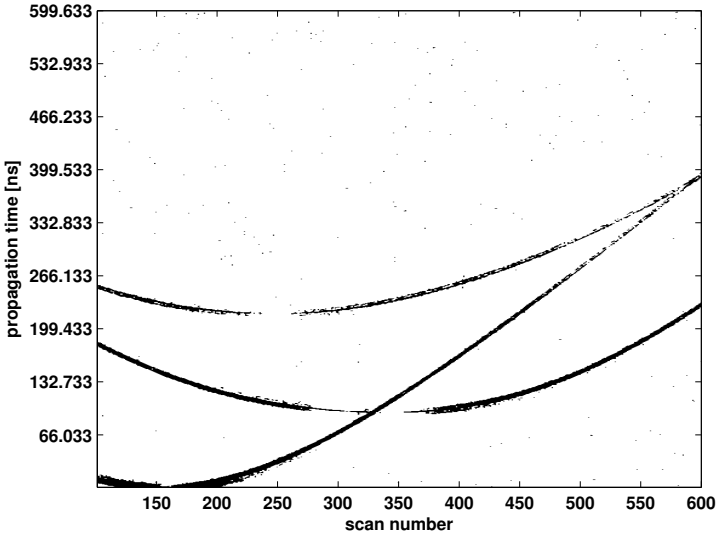
6.4 Conclusion

In this chapter, we proposed a new CFAR detector based on a simplified median filter and a pixel-based localization technique for UWB multistatic radars. The performance of the proposed approaches was evaluated numerically for multiple targets, accounting for the spatial configuration of the receivers, propagation effects, the presence of residual clutter, and noise.

The numerical results showed that the proposed pixel-based localization technique combined with median filtering provides a low positioning error and a good tradeoff between false alarms and missed detections. Thus, it can be a good choice for surveillance applications of UWB multistatic radars with the blind zone problem.



(a) Before median filtering



(b) After median filtering

Figure 6.3: Binary image constructed by CFAR detector output.

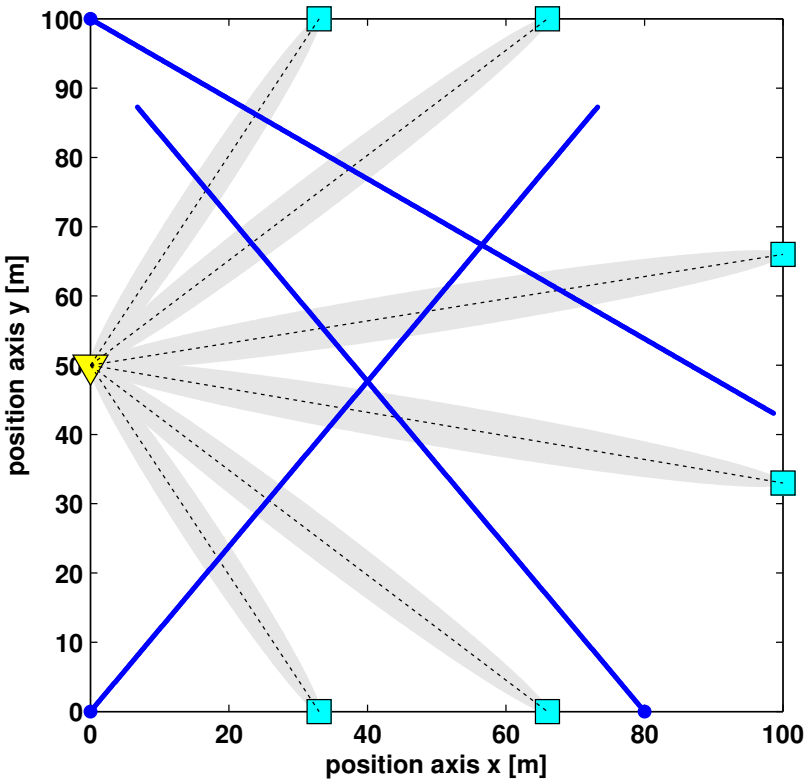


Figure 6.4: The assumed scenario.

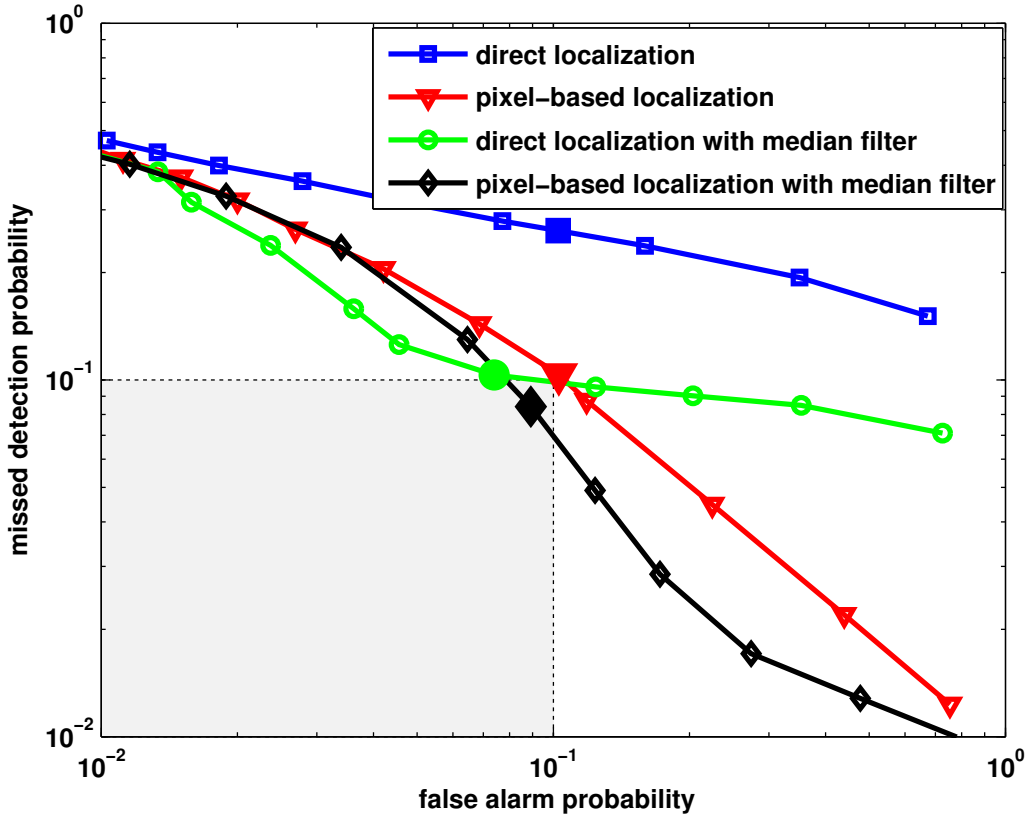


Figure 6.5: Comparison of the tradeoff between missed detection probability and false alarm probability.

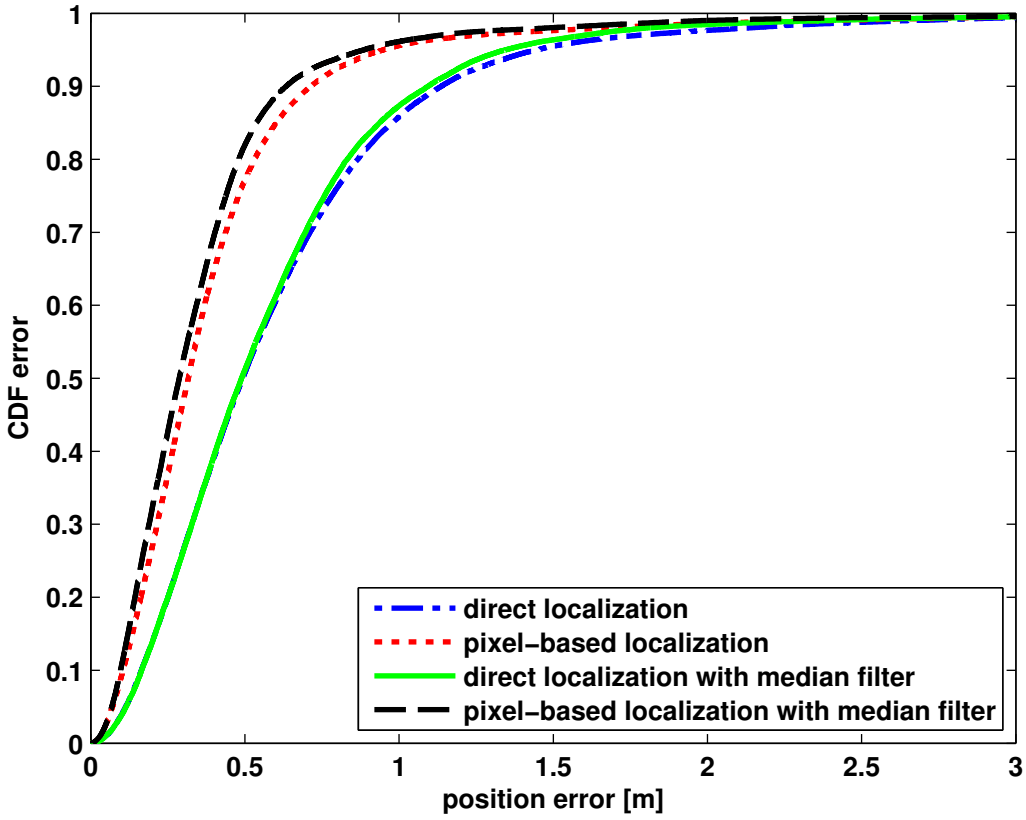


Figure 6.6: Comparison of localization errors.

Chapter 7

Multiple Target Tracking with Particle Filter for UWB Multistatic Radars

In Chapter 4, we proposed an algorithm based on particle filtering for tracking a single target in a UWB multistatic radar. In Chapter 5, we compared this particle filtering algorithm with the grid-based Bayesian approach. We showed that the real-time implementation of particle filtering is feasible in this context.

In this chapter, the particle algorithm in Chapter 4 is extended to the challenging problem of multiple target tracking in UWB multistatic radars with one transmitter and several receivers. Analytical discussions are first provided for calculating particle weights based on observations from all of the receivers. Then, these particle weights are used for estimating target positions by the weighted sum of all particles. In a multiple target scenario, clutter removal, detection, localization and data association have to be performed before the tracking part by particle filtering. The clutter removal step removes the contribution of unwanted objects to the received signals. The detection step decides if any target is present in the area and estimates the target TOAs. The localization step generates the position measurements based on the estimated target TOAs. The data association step verifies these position measurements for track assignment, initiation, confirmation or deletion.

This chapter is organized as follows. In Section 7.1, particle weights are derived considering a multiple target case. The tracking system structure is described in Section 7.2. In Section 7.3, numerical results are illustrated and the performance of the proposed particle filtering algorithm is compared with the Kalman filter for a multiple target scenario. Concluding remarks

are provided in Section 7.4.

7.1 Particle Weights Derivation

We start by considering a single pair of TX-RX antennas. Let us assume that $r_j(t)$ is the received signal at the j'th receiver, obtained by averaging the N_s received waveforms corresponding to the N_s pulses transmitted within a frame [10]. After sampling the received signal, we denote the vector of L samples for each frame corresponding to j'th receiver by $\mathbf{r}^j = [r_1^j \cdots r_L^j]^T$.

Assume that we have M targets, each with a state space $\mathbf{x}_m = [x_m \; y_m]^T$, $m = 1, \ldots, M$ consisting of position components in x and y directions. We extend this state space vector to a matrix containing all of the targets as:

$$\mathbf{X} = [\mathbf{x}_1 \; \cdots \; \mathbf{x}_M] = \begin{bmatrix} x_1 & \cdots & x_M \\ y_1 & \cdots & y_M \end{bmatrix} \tag{7.1}$$

Assume that the channel delays and amplitudes corresponding to the targets are $k_1^j, \ldots, k_M^j$ and $a_1^j, \ldots, a_M^j$, respectively. Note that channel gains are real positive or negative values and we assume them to be constant over one frame.

To simplify the derivation of particle weights we assume that the received pulse from direct path and the clutters have been removed completely.[1] Therefore:

$$r_l^j = a_1^j p_{l-k_1^j} + \cdots + a_M^j p_{l-k_M^j} + n_l^j, \quad l = 1, \ldots, L \tag{7.2}$$

In the above equation, p_l is the l'th sample of the UWB reference pulse,[2] k_m^j is the sample delay of the pulse scattered by the m'th target and n_l^j is the l'th sample of the received additive white Gaussian noise.[3]

If we define the vector of channel amplitudes and target echo pulses as:

$$\boldsymbol{a}^j = [a_1^j \; \cdots \; a_M^j]^T \tag{7.3}$$

$$\mathbf{p}_l^j = [p_{l-k_1^j} \; \cdots \; p_{l-k_M^j}]^T, \quad l = 1, \ldots, L \tag{7.4}$$

Then (7.2) can be simply written as:

$$r_l^j = {\mathbf{p}_l^j}^T \boldsymbol{a}^j + n_l^j, \quad l = 1, \ldots, L \tag{7.5}$$

[1] Anyway, numerical results account for non-ideal clutter removal to assess the performance of the proposed algorithm in a realistic setup.

[2] It is the transmitted pulse, but it can also include antenna and channel propagation distortions.

[3] Since we consider UWB signals, equivalent baseband notation is not adopted and all signals are real.

In particle filtering, the particles are taken on the target state space $\mathbf{X}$ [32]. Since we want to derive the weight of each particle, the channel delay corresponding to a given particle is known by the particle distances from TX and RXs. For now, assume that we have already estimated the channel gain vector $\boldsymbol{a}^j$. Therefore, ${\mathbf{p}_l^j}^T\boldsymbol{a}^j$ is known for a given particle. From (7.5) it can be seen that each r_l^j is a Gaussian distributed random variable with mean equal to ${\mathbf{p}_l^j}^T\boldsymbol{a}^j$ and variance equal to the received noise variance σ^2, that is, $r_l^j \sim \mathcal{N}({\mathbf{p}_l^j}^T\boldsymbol{a}^j, \sigma^2)$, $l = 1, \ldots, L$. Given the statistical independence of noise samples we have:

$$p(\mathbf{r}^j|\mathbf{X}) \propto \exp\left\{-\frac{\sum_{l=1}^{L}(r_l^j - {\mathbf{p}_l^j}^T\boldsymbol{a}^j)^2}{2\sigma^2}\right\}. \tag{7.6}$$

Note that in the above equation, $\mathbf{p}_l^j$ is a function of $\mathbf{X}$. This dependency is not explicitly shown for the sake of simplicity.

In this work, we adopt the maximum likelihood (ML) approach for estimating the channel amplitudes, i.e. we determine the $\boldsymbol{a}^j$ which maximizes the likelihood function $p(\mathbf{r}^j|\mathbf{X}, \boldsymbol{a}^j)$. After some computations, it turns out that the ML estimation for $\boldsymbol{a}^j$ should satisfy the following equation:

$$\mathbf{P}^j\boldsymbol{a}^j = \sum_{l=1}^{L} r_l^j\mathbf{p}_l^j \tag{7.7}$$

where, the matrix $\mathbf{P}^j$ is defined by the cross correlation of target echo pulses as:

$$\mathbf{P}^j = \sum_{l=1}^{L}\mathbf{p}_l^j{\mathbf{p}_l^j}^T = \begin{bmatrix} \sum_{l=1}^{L} p_{l-k_1^j}^2 & \cdots & \sum_{l=1}^{L} p_{l-k_M^j}p_{l-k_1^j} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{L} p_{l-k_1^j}p_{l-k_M^j} & \cdots & \sum_{l=1}^{L} p_{l-k_M^j}^2 \end{bmatrix} \tag{7.8}$$

It can be seen that $\mathbf{P}^j$ is a square symmetric matrix. Using this property and having (7.7), (7.6) simplifies to:

$$p(\mathbf{r}^j|\mathbf{X}) \propto \exp\left\{\frac{1}{2\sigma^2}(\boldsymbol{a}^j)^T\mathbf{P}^j\boldsymbol{a}^j\right\}. \tag{7.9}$$

The set of linear equations in (7.7) yields a unique solution for $\boldsymbol{a}^j$, if $\mathbf{P}^j$ is an invertible matrix. However, this may not be the case. For instance, consider a situation where for a given frame time, all of the targets lie on the ellipses corresponding to equal TOAs. Therefore, all of the target echo delays will

be the same, resulting in equal rows of the matrix $\mathbf{P}^j$. So, $\mathbf{P}^j$ will not be a full rank matrix for that case and hence it is not invertible.

For a general $\mathbf{P}^j$ with rank r, we use QR factorization approach to find a possible solution for (7.7) that better fits our UWB multistatic radar configuration.

7.1.1 QR Factorization for Estimating Channel Amplitudes

We start with the full QR factorization of $\mathbf{P}^j$ with column permutations:

$$\mathbf{P}^j\mathbf{E} = \mathbf{QR} \tag{7.10}$$

where, $\mathbf{E}$ is an $M \times M$ permutation matrix and $\mathbf{Q}$ and $\mathbf{R}$ are $M \times M$ orthogonal and upper triangular matrices, respectively which can be written as:

$$\mathbf{Q} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix} \tag{7.11}$$

$$\mathbf{Q} = \begin{bmatrix}\mathbf{R}_1 & \mathbf{R}_2 \\ \mathbf{0}_{(M-r)\times r} & \mathbf{0}_{(M-r)\times(M-r)}\end{bmatrix} \tag{7.12}$$

Here, the submatrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ have $M \times r$ and $M \times (M-r)$ dimensions, respectively. Also, $\mathbf{R}_1$ is an $r \times r$ upper triangular with nonzero elements along its main diagonal, and $\mathbf{R}_2$ has $r \times (M-r)$ dimension. The zero submatrices in the bottom row of $\mathbf{R}$ have $M-r$ rows.

Let's define the $M \times 1$ vector $\mathbf{z}$ as:

$$\mathbf{z} = \mathbf{E}^T\boldsymbol{a}^j = [\mathbf{z}_1\ \mathbf{z}_2]^T \tag{7.13}$$

where $\mathbf{z}_1$ and $\mathbf{z}_2$ are $r\times1$ and $(M-r)\times1$ vectors respectively. If $\mathbf{Q}_2^T(\sum_{l=1}^{L} r_l^j\mathbf{p}_l^j) \neq 0$, there is no solution for (7.7). Now let's assume that we do have $\mathbf{Q}_2^T(\sum_{l=1}^{L} r_l^j\mathbf{p}_l^j) = 0$. Then (7.7) reduces to:

$$\mathbf{R}_1\mathbf{z}_1 + \mathbf{R}_2\mathbf{z}_2 = \mathbf{Q}_1^T(\sum_{l=1}^{L} r_l^j\mathbf{p}_l^j) \tag{7.14}$$

which is a set of r linear equations with M unknown variables. So, we have $M-r$ excessive variables. The common and easiest way to solve such set of equations is to set all excessive variables to zero and find the others. This is done by setting $\mathbf{z}_2 = 0$. However, this solution is not appropriate for our application, because it means that we are removing the contribution of some of the targets by setting their echo amplitudes to zero.

A possible solution that seems to be more appropriate for our case is obtained by setting all the $M - r$ excessive variables equal to each other and solve all variables from the set of equations so that their variances is minimum. This means that all target echo amplitudes are fairly solved. After doing the required computations, the following solution is achieved:

$$\mathbf{z} = \mathbf{v} - \frac{cov(\mathbf{v}, \mathbf{s})}{var(\mathbf{s})}\mathbf{s} \tag{7.15}$$

where the two $M \times 1$ vectors $\mathbf{v}$ and $\mathbf{s}$ are defined as:

$$\mathbf{v} = \begin{bmatrix} \mathbf{R}_1^{-1}\mathbf{Q}_1^T(\sum_{l=1}^{L} r_l^j \mathbf{p}_l^j) & \mathbf{0}_{(M-r)\times 1} \end{bmatrix}^T \tag{7.16}$$

$$\mathbf{s} = \begin{bmatrix} \mathbf{R}_1^{-1}\mathbf{R}_2\mathbf{1}_{(M-r)\times 1} & -\mathbf{1}_{(M-r)\times 1} \end{bmatrix}^T \tag{7.17}$$

Here, $\mathbf{0}$ and $\mathbf{1}$ are $(M - r) \times 1$ vectors of all zero and all one elements, respectively. Also, *cov* stands for covariance between the two vectors and *var* denotes the vector variance.

Finally, the ML estimation of target echo amplitudes $\boldsymbol{a}^j$ can be find by:

$$\boldsymbol{a}^j = \mathbf{E}\mathbf{z} \tag{7.18}$$

Note that (7.15) is valid only if $var(\mathbf{s}) \neq 0$. For the particular case of $var(\mathbf{s}) = 0$ we proceed by usual approach of setting $\mathbf{z}_2 = 0$ to find a possible solution of (7.7). So, in this case we obtain $\mathbf{z} = \mathbf{v}$ instead of (7.15).

By replacing the solved $\boldsymbol{a}^j$ in (7.9), the likelihood function is derived for a single pair of TX-RX antennas. For the general case of multistatic radar with a single TX and N_R RX antennas, the following expression is achieved by statistically independence of the received vectors:

$$\begin{aligned} p(\mathbf{r}|\mathbf{X}) = p(\mathbf{r}^1, \mathbf{r}^2, \cdots, \mathbf{r}^{N_R}|\mathbf{X}) &= \prod_{j=1}^{N_R} p(\mathbf{r}^j|\mathbf{X}) \\ &\propto \exp\left\{ \frac{1}{2\sigma^2} \sum_{j=1}^{N_R} (\boldsymbol{a}^j)^T \mathbf{P}^j \boldsymbol{a}^j \right\}. \end{aligned} \tag{7.19}$$

In this work, we consider the SIR type of particle filtering [32] in which resampling is done at each scan time (frame time). Therefore, the weight of particle i at scan time n is only proportional to the likelihood $p(\mathbf{r}|\mathbf{X}_n^i)$ and not dependent on its previous weight.

7.2 Tracking System Structure

After clutter removal, there are a few processes that should be done in the central node before performing particle filtering for multiple target tracking. Fig. 7.1 shows the overall view of the system functions at each scan time.

The aim of the detection stage is to decide if any target is present in the area based on the received signals. It processes the received signals and estimates target TOAs for each receiver. The localization step then uses the estimated TOAs to generate position measurements called *observations*. The outputs are then passed to the data association step which associates the observations to the corresponding tracks. It is responsible for initiating tracks, forming potential tracks, confirming tracks and deleting them.

The detection and localization techniques adopted in this work are the CA-CFAR detector with median filtering and direct method of localization described in Section 6.2.1 and Section 6.1.2, respectively. In Chapter 6, we showed that this combination provides the best tradeoff between the probability of false alarms and missed detections for the false alarm probability range of interest.

Since the focus of this work is on the tracking phase, we have adopted the simple and most commonly used data association logic in [50] to avoid further algorithm complexity. In this technique, all of the *potential tracks* are constructed at each scan time and under certain conditions are converted into *confirmed tracks*. A potential track is a sequence of observations over time that are likely to be a track, whereas a confirmed track is certainly a track.

An observation is assigned to a (potential or confirmed) track, if it falls inside a gate around the previous observation assigned to that track. The radius of the gate is defined according to the target maximum velocity. If a potential track reaches a certain number of assignments, it is then converted to a confirmed track. If a (potential or confirmed) track is not assigned any observation (it is missed) for a certain number of times, it is deleted.

Considering the above logic, three cases may happen with respect to an observation. If an observation falls into the gate of only one track, it is then assigned to that track. If an observation falls into the gate of more that one track, it is assigned to the nearest one. If an observation does not fall inside the gate of any track, then it will start a *new track*.

On the other side, three cases may happen with respect to each track. If a single observation falls into the track gate, then that observation is assigned to the track. If more than one observation falls into the track gate, then the nearest observation is assigned to the track. If no observation falls inside the track gate, then the track is considered as a *missed track*.

At each scan time, the confirmed tracks are passed to the particle filtering which smooths out the target state estimations.

7.2.1 Particle Filtering

Particle filtering is only performed for the confirmed tracks. It predicts the next state of the targets and then it corrects its prediction according to the observations.

Since the particles are taken on the state space $\mathbf{X}$ in (7.1) which jointly contains all target states, the optimum way for constructing the particles is by considering all possible combination of particles from individual tracks. Obviously, this approach results in a very time consuming tracking algorithm in which the number of particles increases exponentially with the number of targets.

In this work, we consider a suboptimal approach in which the tracking complexity increases only linearly with the number of targets. For this purpose, particle filtering is done separately for each confirmed track. When considering a confirmed track, all the other (non missed) confirmed track states are fixed to their assigned observation.

Using the same particle filtering algorithm as in Section 4.3, the flowchart in Fig. 7.2 is achieved for confirmed track m at scan time n. We consider N_p particles for confirmed track m. The state of its i'th particle at scan time n is denoted by $\mathbf{x}^i_{m,n}$.

The initial particles for track m are taken from a normal distribution around its assigned observation. Next in the time update step, particles are predicted according to the dynamic model assumed for the target motion. As the target dynamic model we assume that the target average velocity is constant during a sliding window over time. Within each window, we give a weight to each estimated velocity according to its difference from the average velocity calculated in the previous scan time. The higher is the difference, the lower will be its contribution in calculating the mean (See Section 4.3).

In order to avoid tracking divergence due to any large detection error, every time a particle is predicted too far from the track assigned observation, it is reset to a sample from a Gaussian distribution around that observation.

Next, in order to calculate the likelihood, we have to construct the state matrix $\mathbf{X}^i_{m,n}$ for each particle i of track m. This matrix is formed by fixing all the other (non missed) confirmed track states to their assigned observation. For example, if there are M_c non missed confirmed tracks other than track m, with their assigned observations as $\mathbf{x}^o_{1,n}, \ldots, \mathbf{x}^o_{M_c,n}$, then the state matrix

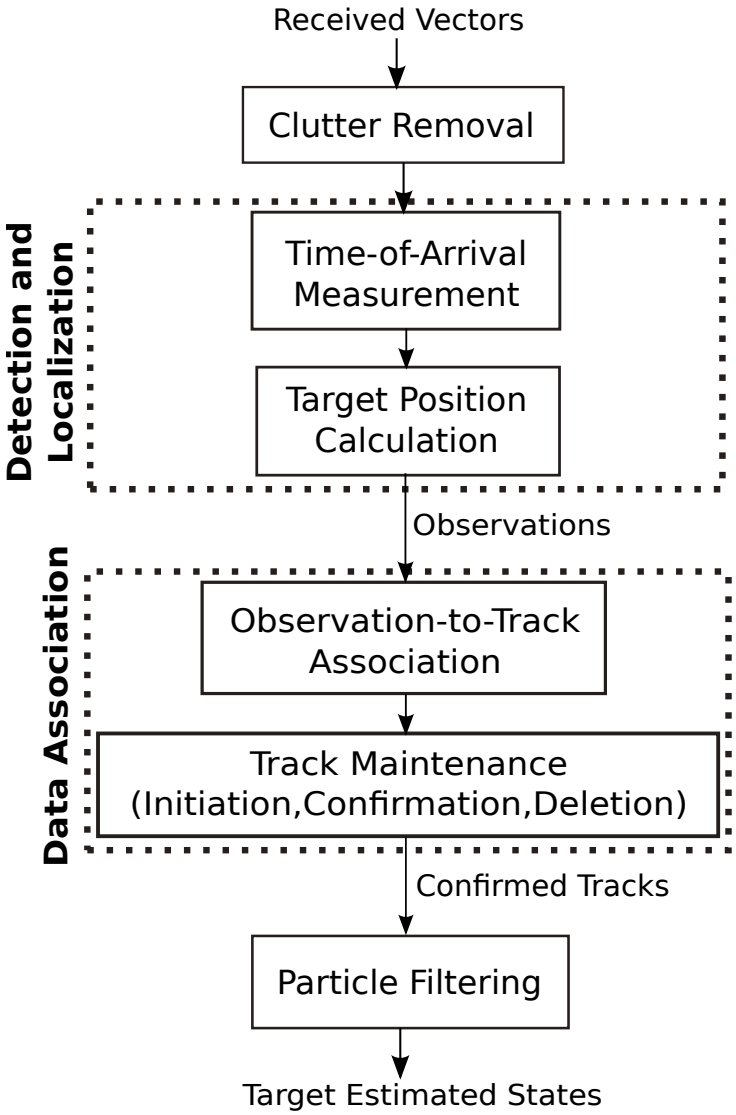


Figure 7.1: Conceptual system scheme.

for particle i of track m is formed by:

$$\mathbf{X}^i_{m,n} = [\mathbf{x}^o_{1,n} \ \cdots \ \mathbf{x}^o_{M_c,n} \ \mathbf{x}^i_{m,n}] \tag{7.20}$$

Now, the sample delays corresponding to each state element $\mathbf{x}^o_{1,n}, \ldots, \mathbf{x}^o_{M_c,n}, \mathbf{x}^i_{m,n}$ of $\mathbf{X}^i_{m,n}$ with respect to receiver j can be easily calculated by sum of the state distances from the transmitter and the receiver j, divided by the speed of light. Then, the matrix $\mathbf{P}^j$ can be constructed from (7.8) and amplitude vector $\boldsymbol{a}^j$ is estimated by QR factorization described in Section 7.1.1. Finally, the likelihood $p(\mathbf{r}|\mathbf{X}^i_{m,n})$ is obtained by (7.19) for particle i of track m.

Next in weight update step, we normalize this likelihood over all particles of track m to get the particle weights. In order to avoid degeneracy problem of particle filters, systematic resampling is done at each scan time that replaces low probability particles with high probability particles, keeping the number of particles constant [32]. After the resampling step, all new particles will have the same weights equal to $1/N_p$.

Then, the estimated state of track m is calculated by weighted sum of all particles. Using the estimated states of track m during a sliding window over time, we calculate the movement mean as in Section 4.3. Finally, the outputs are saved in a track file and the algorithm steps are repeated for the another confirmed track at scan time n.

7.3 Simulation Results

A square surveillance area of 100×100 meters is watched by a UWB radar sensor network composed of one transmitter and six receivers located on the square sides. The origin of our assumed coordinate system is the lower left corner of the square. Therefore, the TX node is located at position $(0, 50)$, while the other 6 RX nodes are at positions $(33, 0)$, $(66, 0)$, $(100, 33)$, $(100, 66)$, $(33, 100)$ and $(66, 100)$, respectively.

The TX node emits first derivative Gaussian monocycles with duration parameter $1.4ns$ and whose power spectral density is assumed to exceed the FCC mask by 10 dB. [4] The number of pulses in each frame is $N_s = 114000$ and the frame duration is $T_F = 68.4ms$. The sampling frequency is set to 1.5 GHz. For each RX node, the receiver noise figure and the antenna temperature are set to $F = 6$ dB and $T_a = 290K$, respectively. These settings result in a transmitted signal power of -32.5 dBW and a received noise power -86.2 dBW.

[4] This exceeding has been chosen to cover the large area assumed. The possibility to exceed the FCC mask even by 20 dB is under consideration in the EU for emergency applications.

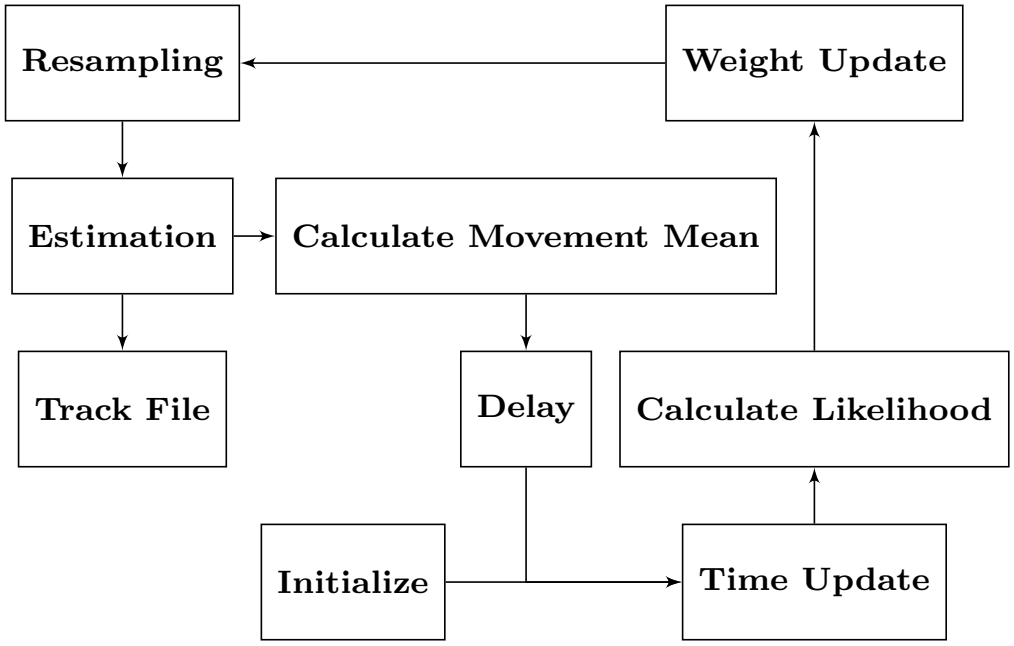


Figure 7.2: Flowchart of the particle algorithm.

We consider a number of 100 pointwise objects to be present in the surveillance area as clutters. The clutters are distributed uniformly over the whole surveillance area. We consider static clutters with $1m^2$ radar-cross-sections, the same as the target. Each RX node implements a frame-to-frame clutter removal technique based on a first-order high-pass IIR filter with one pole equal to 0.9 and a delay of 100 scan times operating at a sampling frequency $1/T_F = 14.6\,\mathrm{Hz}$.

At each RX antenna, the received signal is constructed as the superposition of the direct path, clutter echoes, and ground reflection in addition to the target echo and the noise. The channel gains have been simulated according to $\sqrt{G_t G_r \lambda_0^2/((4\pi)^2 l^2)}$ for the LOS and $\sqrt{G_t G_r \lambda_0^2 \sigma_o/((4\pi)^3 l_r^2 l_t^2)}$ for any object including the target [10], where G_t and G_r are the TX and RX antenna gains, respectively, which are both equal to 0 dB for our omnidirectional antennas and λ_0 is the wave length which has been calculated based on the center frequency 4.5 GHz. Moreover, l is the TX–RX distance, while l_t and l_r are the distances of the object from TX and from that RX, respectively. Finally, σ_o is the radar-cross-section of the object.

In the numerical results, the tracking performance is measured in terms of the number of confirmed tracks, RMS and CDF errors over 20 simulation runs, each with a different noise and clutter realization. The number

of confirmed tracks at each scan time is averaged over all simulation runs. For generating the RMS error plot of each target, we average the vector of estimation errors of that target at all scan times over all simulation runs. For the CDF error plot, we concatenate the estimation error vectors of all targets over all simulation runs and then we calculate the CDF for the resulting vector. For each scenario, the performance of well known Kalman filter is also shown for comparison.

For all parts of the tracking system, the corresponding parameters have been set to the values maximizing the performance. In the detection, a guard band equal to 20 samples (the reference pulse width) is chosen for CFAR thresholding. The average noise power is calculated for a number of 50 samples after each guard band. The threshold is constructed by multiplying the calculated noise power with a constant. This constant has been set to 11 for the first scan time and then reduced to a very low value of 3 for all other times. The square of the sample under verification is then compared with the resulting threshold.

In data association, if a track is missed for 10 times, then it is deleted. A potential track is confirmed, when it is assigned at least 3 consecutive non missed observations.

For particle filtering, we have set the number of particles to 200 for each confirmed track. In Chapter 4, it has been shown that this number reaches the best tradeoff between the estimation accuracy and complexity. The size of the sliding window for calculating the movement means has been set to 20 scan times and the process noise standard deviation has been chosen to be $0.1m$ for both x and y directions. For Kalman filter, the acceleration noise and measurement noise standard deviations have been set to $1.6m/s^2$ and $3m$, respectively.

Fig. 7.3 shows the tracking performance of filtering techniques for two targets moving on straight line trajectories with a constant velocity of $10km/h$. The assumed target trajectories are shown in the first subfigure for a duration of 600 scan times. Their starting positions are shown with circles. In this subfigure, the triangle represents the transmitter and the squares are RX antenna positions. The gray ellipses are the blind zones corresponding to each pair of TX-RX antennas.

In the CDF and RMS error plots, the observation (measurement) errors corresponding to the target assigned track are also shown in the plots to see how much the tracking filter can improve the measurement accuracy. It can be seen that the measurement error plots for particle and Kalman filters almost coincide. This coincidence also happens for the plots corresponding to the number of confirmed tracks. Since these outputs are generated before the tracking filter is performed, this fact insures that the scenarios for

both particle and Kalman filters have been the same on the average over all simulation runs.

It can be seen from the results that Kalman filter generates estimation errors even larger than measurement errors, while particle filtering improves the precision for most of the time. For higher measurement errors, the performance of particle filtering comes close to that of the measurements. This is because when forming the particle state matrix for a track according to (7.20), all the other track states are fixed to their assigned observations. Therefore, when these observations differ too much from the true target positions due to large detection errors (which are caused either by interference of target echo pulses or blind zones), the particle state matrix is not formed correctly and hence the performance of particle filtering deteriorates.

Fig. 7.4 adds another target to the scenario shown in Fig. 7.3. By increasing the number of targets, the number of times that target echo pulses interfere each other increases, which results in larger detection errors. Therefore, particle filtering can less improve the measurement accuracy compared to the two targets case. However, it is evident from CDF error plots that particle filter still outperforms the measurements for most of the time, whereas Kalman filter is always worse.

In Fig. 7.5 manoeuvering trajectories are assumed for the targets instead of straight lines. In the trajectories, we have tried to include various shapes such as wave-like curves, circle-like movements, as well as sudden direction changes. The two targets are moving on the curves with a constant velocity of $10km/h$. It can be seen from the figure that the particle filter can track all maneuvers with high precision while Kalman filter generates larger estimation errors.

7.4 Conclusion

In this chapter, the particle algorithm in Chapter 4 was extended to the case of multiple target tracking in UWB multistatic radars. The particle weights were derived analytically assuming one transmitter and multiple receivers. We illustrated through numerical results that the proposed particle filtering algorithm outperforms the Kalman filter in multiple target scenarios accounting for the spatial configuration of the receivers, propagation effects, the presence of residual clutter and noise.

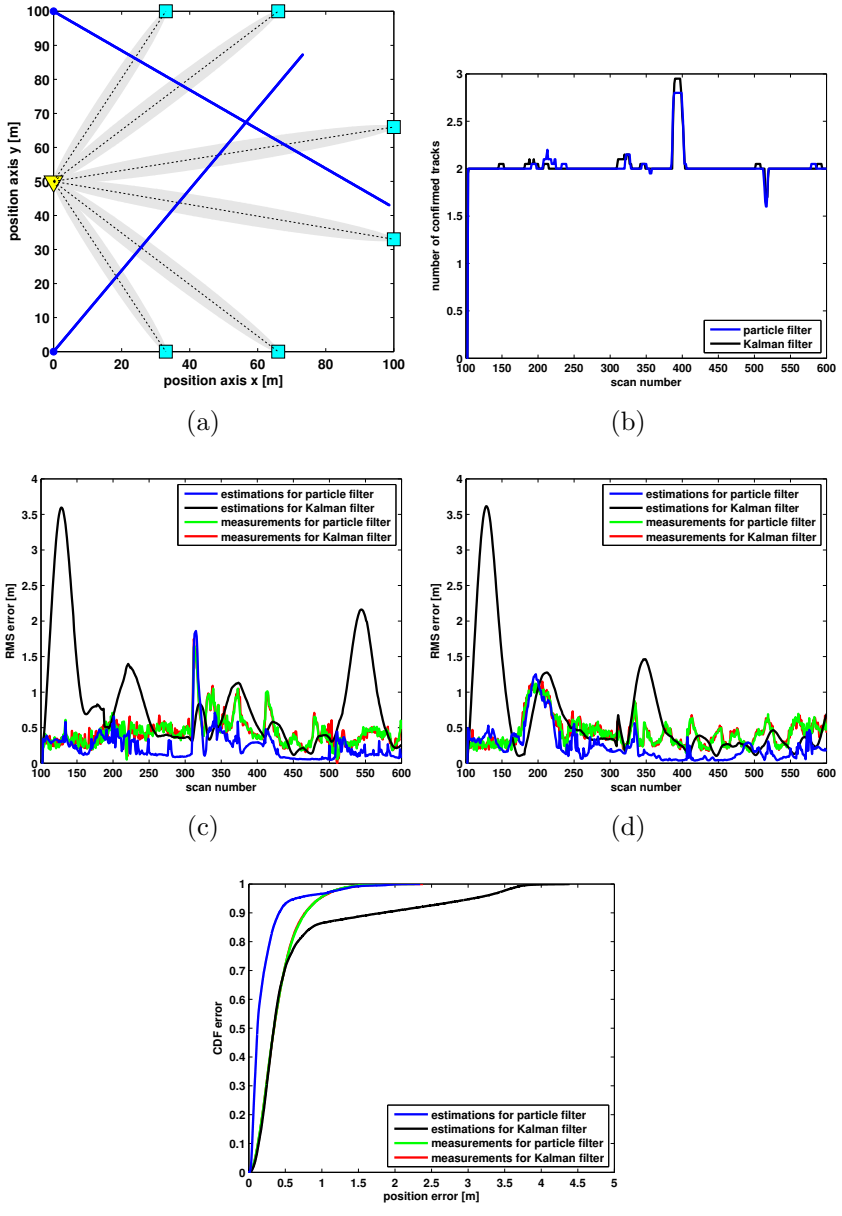


Figure 7.3: Performance comparison of particle filter and Kalman filter for two targets moving on straight lines. (a): The assumed target trajectories. (b): Number of confirmed tracks vs. time. (c): RMS error plots for the target with initial position $(0, 0)$. (d): RMS error plots for the target with initial position $(0, 100)$. (e): CDF error plots for all targets.

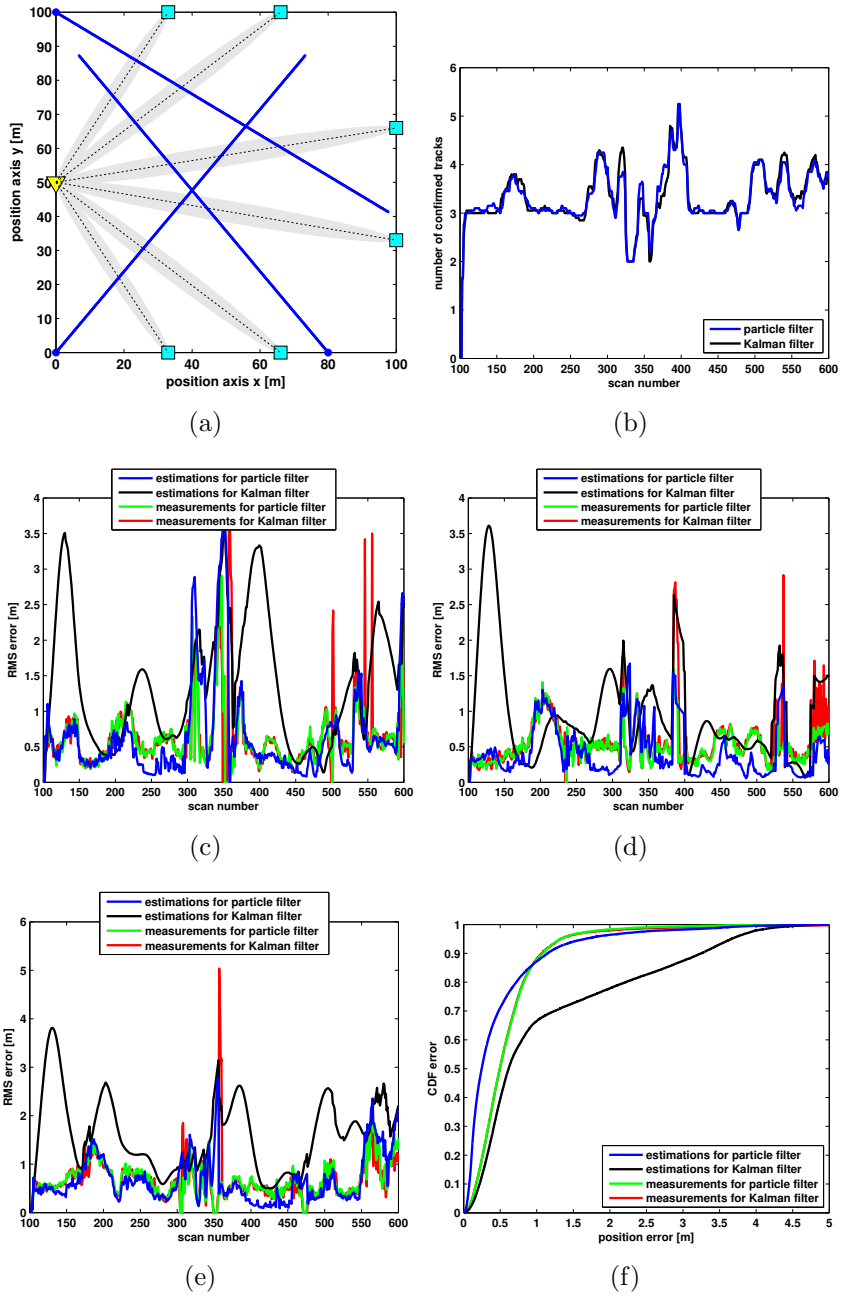


Figure 7.4: Performance comparison of particle filter and Kalman filter for three targets moving on straight lines. (a): The assumed target trajectories. (b): Number of confirmed tracks vs. time. (c): RMS error plots for the target with initial position $(0, 0)$. (d): RMS error plots for the target with initial position $(0, 100)$. (e): RMS error plots for the target with initial position $(80, 0)$. (f): CDF error plots for all targets.

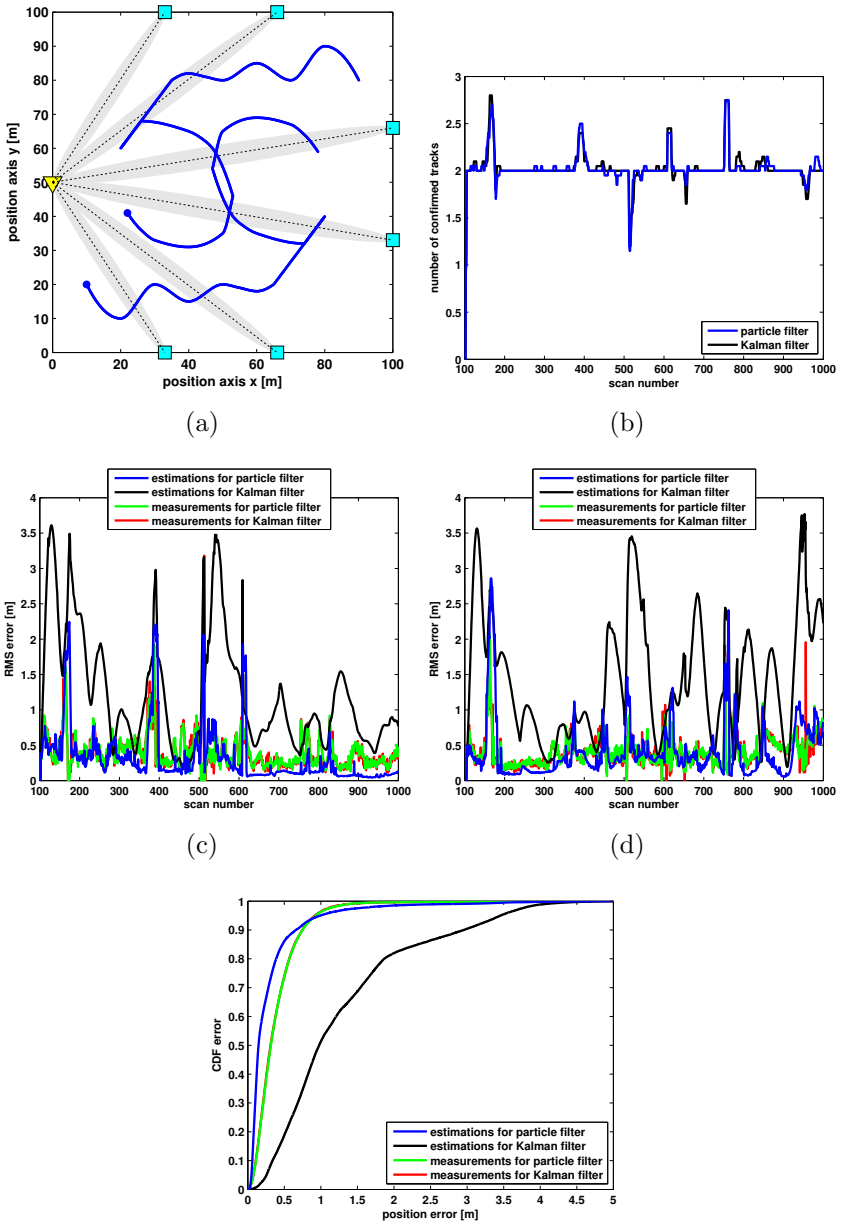


Figure 7.5: Performance comparison of particle filter and Kalman filter for two targets with manoeuvering trajectories. (a): The assumed target trajectories. (b): Number of confirmed tracks vs. time. (c): RMS error plots for the target with initial position $(10, 20)$. (d): RMS error plots for the target with initial position $(22, 41)$. (e): CDF error plots for all targets.

Chapter 8

Multiple Target Detection and Tracking with the Hough Transform for UWB Multistatic Radars

The Hough transform is a feature detection method originally developed for the detection of straight lines in an image corrupted by noise [51]. However, in principle, other arbitrary curves can be detected as well.

In the radar literature, the Hough transform was firstly introduced for target detection improvement and simultaneous track acquisition in [52, 53, 54]. The technique combines range data from previous search scans into a two-dimensional data map which can be considered as an image. Targets appear as curves or *features* in this data space. Then, the Hough transform is used to extract the target tracks from the data. This new method has the advantage of not discarding data from past scans which still may contain target returns even if no detection was declared (as opposed to most traditional radar processing in which previous data is discarded). More recent researches on the Hough transform for radar applications can be found in [55, 56, 57].

The Hough transform has many other advantages. It can be used to detect any desirable predefined curve [58]. Hence, more a priori information about the target movement can be introduced in processing. The Hough transform is able to perform target detection, data association, track initiation and track maintenance at the same time. Moreover, it can simultaneously detect multiple curves, which is suitable for multiple target tracking [59]. The main limitations are related to the computational complexity and memory requirements when the dimension of the data space or the parameter space is large. Also, the performance can be greatly deteriorated if the target movement is

not in accordance with the assumed mobility model.

Although the literature on the Hough transform in radar applications is quite extensive, there are only a few works studying the Hough transform in UWB radars. In [60], a vehicle detection scheme is suggested using the Hough transform where some moving vehicles are discriminated from significant clutter by using linear trajectory detection in the Hough space. The same technique is proposed in [61] for pedestrians and two-wheelers. In [62], an approach for noncoherent integration of pulses based on the Hough transform is proposed for UWB radar. All the above works do not consider a multi static radar, and are based on the Hough transform for straight-line detection.

Up to our best knowledge, no work has been done yet on the usage of the Hough transform in UWB multistatic radars, particularly human tracking in indoor surveillance applications.

In this chapter, we propose two approaches which use the Hough transform for tracking humans walking inside the surveillance area. In one approach, the Hough transform is employed to find the time of arrival (TOA) curves of the targets in the scan vs. propagation time image, for IR-UWB multistatic radars. In order to define the target TOA curve, a straight-line constant-velocity mobility model is assumed for the target, which is typically a human walking inside the surveillance area. We show that the Hough space is a four-dimensional space, and we discuss the feasibility of our proposed algorithm implementation in terms of computational complexity and data storage requirements.

In the other approach, first the pixel-based localization technique, described in Section 6.2.2, is employed to find the potential target positions in the area. Then, two 2D Hough transforms are applied sequentially for x and y directions to find the target trajectory.

We experimentally evaluate the performance of the proposed approaches for monitoring a small indoor area. Here, due to the presence of furniture, walls and other objects, a dense clutter is observed. One of the most common clutter removal techniques is the empty-room method (background subtraction) [10, 63]. However, in indoor environments, the target presence may shadow other objects in the area, affecting their reflected signals. Eventually, this shadowing effect may result in a large number of false alarms after background subtraction. We therefore propose a new empty-room clutter removal technique based on dissimilarities between the received signals with and without the target instead of simply subtracting them. We show that the proposed approach can considerably reduce the number of false alarms.

This chapter is organized as follows. Our channel measurement setup is given in Section 8.1. In Section 8.2, a new clutter removal technique is

proposed. Section 8.3 briefly introduces the concept of the Hough transform. Section 8.4 explains our proposed TOA association technique using the Hough transform. In Section 8.5, the proposed technique is evaluated by experimental measurements. Section 8.6 presents another approach which extracts the target trajectory by performing two 2D Hough transforms. In Section 8.7, this approach is evaluated with experimental measurements. Concluding remarks are provided in Section 8.8.

8.1 Channel Measurement Setup

The measurements were done in a seminar room with the dimensions of $3.52m \times 5.5m \times 3m$ (width/length/height) at IHE. During the measurements, all the furniture and equipment were present, producing significant indoor clutter. Fig. 8.1 shows our measurement scenario. A single TX antenna and three RX antennas were placed in the perimeter of the area, composing a multistatic radar system. In Fig. 8.1(a), the triangle represents the TX antenna and the squares are RX antenna positions. All antennas were directive Vivaldi antennas.

A person, $1.65m$ tall, was walking within the room taking the blue trajectory shown in Fig. 8.1(a) with the initial position depicted by a circle. We used the Agilent E5071C ENA network analyzer with a noise figure of 38dB to measure the channel between the TX antenna and every RX antenna at 14 positions in the person trajectory which have been marked with the crosses in Fig. 8.1(a). We were able to measure only a single channel at a time with our network analyzer. Therefore, we measured the three channels one after the other for each person position.

The S_{21} parameter was measured in the frequency range of 3GHz to 6GHz, with a number of 600 points, an IF bandwidth of 100Hz and an averaging number of 16. A power of +5dBm was delivered to the transmit antenna.

8.2 Proposed Empty-Room Clutter Removal Technique

Target detection in environments with dense scatterers is a challenging task due to the undesired background clutter caused by antenna cross-talk, reflection from the walls and other unwanted objects.

One of the most common approaches to remove the clutter from the received signal is the empty-room (also called background subtraction) method [10, 63]. The empty-room technique subtracts the time-invariant background

of the static scenario in the absence of the target from any received signal composed of both the target and clutters. Unfortunately, this approach is very sensitive to environment changes and the background is in reality a time-variant signal. Moreover, the target presence does not appear only at its corresponding TOA in the received signal as a single point, but rather it may affect many points of the received signal due to the shadowing effect. Therefore, simply subtracting the received signals with and without the target (even if enough number of averaging has been taken) may yield too many false alarms. Thus, clutter reduction is still a challenging task for such dense clutter environments.

According to the experiments, the shadowing effect appears mostly as an attenuation or amplification in the received signal, and the general shape of the received signals with and without the target seems to remain almost unchanged for all TOAs except that of the target. Therefore, in the proposed empty-room clutter removal technique, we look for dissimilarities between the received signals with and without the target instead of simply subtracting them.

To describe this idea mathematically, we assume a sliding window over the propagation time. For each propagation time n, assume that $\mathbf{x}_n = (x_{n-L+1}, \ldots, x_n)^T$ is the received vector within this sliding window without the target, and $\mathbf{y}_n = (y_{n-L+1}, \ldots, y_n)^T$ is the one with the target. Now, we assume that inside the sliding window, there is a shadowing gain a_n, common to all samples within this window, so that $a_n\mathbf{x}_n$ should be removed. The size of the sliding window, L, can be optimized experimentally. Then, instead of simply looking at the energy of the difference vector $\mathbf{y}_n - \mathbf{x}_n$, as in the conventional empty-room, we look at the least square error

$$S_n = \min_a \|\mathbf{y}_n - a\mathbf{x}_n\|^2 = \|\mathbf{y}_n - \hat{a}_n\mathbf{x}_n\|^2 \tag{8.1}$$

where the estimated shadowing coefficient $\hat{a}_n$ is given by

$$\hat{a}_n = \frac{\langle \mathbf{x}_n, \mathbf{y}_n \rangle}{\|\mathbf{x}_n\|^2} = \frac{\sum_{i=1}^{L} x_{n-L+i} y_{n-L+i}}{\sum_{i=1}^{L} x_{n-L+i}^2} \tag{8.2}$$

where $\|\mathbf{x}_n\|$ denotes the norm of $\mathbf{x}_n$ and $\langle \mathbf{x}_n, \mathbf{y}_n \rangle$ represents the inner product of $\mathbf{x}_n$ and $\mathbf{y}_n$. By replacing (8.2) into (8.1), we have

$$\begin{aligned} S_n &= \|\mathbf{y}_n\|^2 - \frac{(\langle \mathbf{x}_n, \mathbf{y}_n \rangle)^2}{\|\mathbf{x}_n\|^2} \\ &= \sum_{i=1}^{L} y_{n-L+i}^2 - \frac{(\sum_{i=1}^{L} x_{n-L+i} y_{n-L+i})^2}{\sum_{i=1}^{L} x_{n-L+i}^2} \ . \end{aligned} \tag{8.3}$$

Therefore, at each propagation time n, we calculate the metric defined in (8.3). A thresholding is then applied on this metric to get the potential target TOAs.

To appreciate the effectiveness of the proposed technique, we compare it with the conventional empty-room method on experimental measurements. In particular, Fig. 8.2(a) shows an example of the channel impulse response obtained by means of a network analyzer for one of the RX antennas with the target at a certain position. The corresponding channel impulse response without the target (empty-room) is also plotted. The green circle represents the theoretically expected target TOA (assuming a pointwise target). The first peak corresponds to the LOS echo which is much bigger than that of the target. It can be observed that detection of the target echo which is hidden by heavy clutter is a challenging task. The two signals do not differ only in the expected target TOA (green point), but also in many other parts due to the shadowing effect. However, it can be seen that around the green point, the two signal shapes seem to change, whereas in the other parts, the general shape is almost preserved. Fig. 8.2(b) shows the simple subtraction of the two signals, which is what is actually done in the conventional empty-room approach. Fig. 8.2(c) plots the metric in (8.3) for the proposed empty-room approach based on dissimilarities between the two signals. We can see that the conventional empty-room technique may generate many false peaks which results in a large number of false alarms after energy thresholding, whereas the proposed approach can greatly reduce the number of false peaks.

8.3 The Hough Transform

The Hough transform is a feature detection technique which can be used to find a desired shape in an image [58]. It requires that the desired curve is specified in some parametric form. The classical Hough transform is most commonly used for detection of straight lines in an image corrupted by noise. We explain the concepts of the Hough transform by the example of detecting straight lines.

Every arbitrary straight line in the (x, y) space can be represented by two parameters ρ (the length of its perpendicular from the origin to the line) and θ (angle of its perpendicular to the origin) as (see Fig. 8.3(a))

$$\rho = x \cos\theta + y \sin\theta \ . \tag{8.4}$$

From (8.4) it can be seen that each point in the data space (x, y) is transformed to a sinusoidal curve in the parameter space (ρ, θ). Points lying on

the same line are mapped onto curves that have a common point of intersection. Therefore, lines in the (x, y) space can be detected by finding the intersection points in the (ρ, θ) space. This concept of the Hough transform is shown in Fig. 8.3.

The Hough transform can be generalized to find any desired curve that can be expressed parametrically. For a practical implementation, both the data space and the parametric space are discretized into pixels. The basic steps are therefore as follows:

1. The desired curve is expressed in a parametric form.
2. The data space is converted to the parameter space using the defined Hough transform.
3. Each pixel in the parameter space is assigned a count value equal to the number of points in the data space that are mapped to it.
4. The pixel whose corresponding count value exceeds a predefined threshold determines the desired curve parameters.
5. The inverse Hough transform is applied to find the desired curve in the data space.

It should be noted that high dimensionality of either the data space or the parameter space results in more computational time and larger memory requirements so that the implementation may become impractical.

8.4 Proposed TOA Association with the Hough Transform

In the proposed approach, the Hough transform is used to find the corresponding TOA curve of the targets in the measured radar image. In order to construct this image, binary data after energy thresholding is collected over a time history (sliding window). The resulting binary image shows the propagation time τ versus scan number $k = 1, \ldots, N_w$, where N_w is the total number of scans in the sliding window.

In order to characterize the target TOA curve, a straight-line constant-velocity mobility model is assumed for the target. This seems the closest model that represents a human walk in anti-intruder applications. In fact, the human walk can be well approximated with this model over a window of at least a fraction of a second. In indoor applications, wherein the human walking speed is slow, the model applies to a window of several seconds.

We parametrize the target TOA curve with four parameters: velocity amplitude (V), velocity angle (α), and initial position (x_0, y_0) at the beginning of the sliding window. It is known that the Hough transform algorithm complexity increases dramatically with the number of parameters. Hence, one may concern about the computational complexity and memory requirements. It should be noted that the human walking velocity range is very limited in indoor environments. Therefore, the range of the parameter V to be discretized is also small. Another point is that the UWB coverage area is inherently small (less than $100m \times 100m$), as compared to conventional radars that deal with areas of many square kilometers. The surveillance area is even smaller in indoor applications. So, we are not dealing with large ranges for the parameters x_0 and y_0 as well. Moreover, coarser discretizations can be used in tradeoff with resolution requirements.

Therefore, the data space is a 2D-space of propagation time τ and scan number k, whereas the parameter space is a 4D-space of (V, α, x_0, y_0). Assuming a pointwise target at height z, the two spaces are related by

$$\begin{aligned} x &= x_0 + (k-1)T_{scan}V\cos\alpha \\ y &= y_0 + (k-1)T_{scan}V\sin\alpha \\ d_1 &= \sqrt{(x-x_t)^2 + (y-y_t)^2 + (z-z_t)^2} \\ d_2 &= \sqrt{(x-x_r)^2 + (y-y_r)^2 + (z-z_r)^2} \\ \tau &= (d_1 + d_2)/c \end{aligned} \tag{8.5}$$

where T_{scan} is the scan time, (x_t, y_t, z_t) and (x_r, y_r, z_r) are the transmitter and receiver coordinates respectively, and c is the speed of light. The flowchart of the algorithm based on the Hough transform is described in Fig. 8.4. First, energy thresholding is performed on the received data and the binary image is constructed as the input for the Hough transform. The parameter space is then discretized.

For each point in the data space, the whole pixels in the parameter space are verified to see if the Hough transform defined in (8.5) is satisfied by that point in the data space and the pixel under test in the parameter space. If so, a count number corresponding to that pixel in the parameter space is increased by one. After completing this step for all points in the data space, a threshold is applied on the count numbers (in single target scenarios this thresholding can be simply replaced by using the maximum) and the parameters of the most probable TOA curves are obtained.

For each curve, the whole points in the data space are verified to see if the Hough transform defined in (8.5) is satisfied by that curve parameters and the point under test in the data space. If so, that point in the data space is assigned to the curve (this is actually the inverse Hough transform).

The next steps are some additional processing that are required to reduce the number of output curves. For example, equal curves, (i.e. having the same points assigned in the data space) should merge together, or if one curve includes all points of another curve, then the shorter curve should be discarded.

Next, the TOA of each remaining curve is calculated for all scan numbers in the sliding window, using (8.5) and the curve parameters. Finally, the curves with close TOAs are assigned together in clusters and replaced by one curve with TOAs averaged over the cluster.

The proposed Hough transform is able to perform TOA detection (distinguish target TOAs from false alarms), data association (associate corresponding observations over scan time) and tracking (smoothing TOA estimation, predicting target TOA and filling out the missed detections) at the same time. It can also be used in multiple target scenarios to estimate the number of targets, initiate and maintain the tracks simultaneously. All the above advantages are due to the Hough transform which uses memory from past data for a more robust decision.

8.5 Experimental Results on the Proposed Hough Transform TOA Association Technique

In order to validate the proposed approach, we used the experimental channel measurements described in Section 8.1. More precisely, we convolved the transmitted signal with the measured channel impulse responses, to get the IR-UWB radar received signals.

As the transmitted pulse, we have chosen the first derivative Gaussian monocycle, modulated with a carrier frequency of 4.5GHz. The modulation for this type of pulse is required to respect the FCC regulations [31].[1] The pulse duration parameter τ_p was chosen to be 0.3ns which achieves a good range resolution for our scenario with a sampling frequency of 20GHz. The amplitude of the pulse was set to fulfill the FCC mask. Fig. 8.5 shows such a designed pulse in both time and frequency domains. It can be seen that the pulse spectral density doesn't exceed FCC regulations.

Considering our antenna configuration and the room size, the maximum

[1]We have chosen this derivative order to match our network analyzer measurable frequency range. Higher orders, such as 5'th derivative Gaussian monocycle may also be considered without modulation, but with a much higher bandwidth. The main concept of the proposed algorithm doesn't depend on the UWB pulse shape used.

possible target echo delay is 35ns. Therefore, the pulse interval T_{IP} was chosen to be 24ns which corresponds to the propagation time difference between the LOS and the worst case target echo.

Fig. 8.6 shows the binary image after energy thresholding constructed from the propagation time vs. scan number (person position number in our case, see Fig. 8.1(a)) for the first RX antenna, as an example, in the UWB multistatic radar system. The figure compares the images using both the conventional and proposed empty-room techniques. In the proposed empty-room approach, the size of the sliding window, L, has been chosen as 15. The threshold has been set to 0.16 of the maximum cluttered removed signal energy for the conventional empty-room approach, and to 0.4 of the maximum metric for the proposed empty-room technique. These threshold values have been chosen experimentally to achieve the best tradeoff between the missed detections and false alarms for both methods. The figure shows the theoretically expected target TOA curve as well. It can be seen that the proposed approach decreases considerably the number of false alarms. These binary images form the data space for our proposed Hough transform TOA association algorithm. Since the Hough transform algorithm complexity increases dramatically with the number of points in the data space, the proposed empty-room approach greatly reduces the required computational time.

Fig. 8.7(a-c) show the results of applying the proposed Hough transform TOA association algorithm, described in Section 8.4, on the radar binary images using the proposed empty-room clutter removal technique. The human average height, z, has been set to $1.5m$ in (8.5). Considering the typical human walking speed of $0.6m/s$ in indoor environments, the time interval between two position measurements will be $0.5s$. We discretized the parameter space as $0.4m/s$ to $1.5m/s$ with steps of $0.1m/s$ for V, 0 to 360 degree with steps of 0.5 degree for α, 0 to $3.52m$ (maximum width of the room) with steps of $0.1m$ for x_0, and 0 to $5.5m$ (maximum length of the room) with steps of $0.1m$ for y_0. Please note that for larger surveillance areas, we can further reduce the range of parameter values x_0 and y_0 by limiting them around the previously estimated target position.

After estimating the target TOA curves by the proposed Hough transform approach, the direct localization (trilateration) [48] was used to localize the target. Fig. 8.7(d) shows the estimated target positions.

We observe that the target TOA points can be well discriminated from false alarms and associated over scan time, forming the depicted target TOA curves. These TOA curves were estimated with a good accuracy, even though the target was missed at some points. For example, at the fourth point of the trajectory (see Fig. 8.1(a)), the target lies in the blind zone corresponding to

the third RX antenna, which results in a miss detection for this scan number in Fig. 8.7(c). However, the estimated target TOA curve corresponding to this antenna is able fill out this miss detection.

It should be noted that the true target TOA curves in these figures have been calculated theoretically assuming a pointwise target at the human average height of $1.5m$. However, in UWB radars, the range resolution is smaller than human dimensions which results in different scattering centers on the target. Therefore, what the figures show as the true target TOAs are only rough approximations of these values.

The proposed Hough transform TOA association algorithm was using a number of $12 \times 721 \times 36 \times 56$ pixels for the parameter space (V, α, x_0, y_0). This equals a memory requirement of 17.5MB for saving the one-byte count values.

It should be noted that considering the constant-velocity assumption does not necessarily mean that the target must strictly follow this mobility model for the Hough transform algorithm to work properly. In fact, since the UWB transmitted pulse width occupies more than one sample in the received signals, the potential target TOAs after energy thresholding do not appear as a single pixel in the binary images for each scan number. This can also be observed in the previously shown figures, wherein the potential target TOAs for each scan number appear as a short line instead of a single point. This allows more flexibility in the target real movement with respect to the assumed model. Moreover, what we measured as the target TOAs are actually different from the real ones due to the limited accuracy of range estimates in indoor areas. Nonetheless, the Hough transform was still able to extract the target TOA curves with a non perfectly matching mobility model.

Using finer discretizations would result in more accurate TOA estimations at the cost of computational complexity and memory requirements. Using more RX antennas can also help improve the positioning accuracy due to spatial diversity, since when the target is in the blind zone for one RX antenna, other antennas may be employed for localization. Increasing the antenna beamwidth can also help reduce the number of missed detections by providing a better area coverage.

8.6 Target Trajectory Extraction Using the Hough Transform

In this section, another approach is presented to extract the target trajectory using the Hough transform. In order to employ the Hough transform to find

the trajectory of the target moving within the surveillance area, the data space should be considered as a 3D space (x, y, t), where x and y are the Cartesian coordinates of the area and t represents the scan times over a time history (sliding window).

Again, we model the target trajectory as straight-line in both (x, t) and (y, t) spaces. However, extracting straight-lines in a 3D data space with the Hough transform requires a 5D parameter space [55]. The implementation of such a large dimensional space is almost impractical due to the high computational time and data storage requirements. To solve this problem, instead of doing one Hough transform for the 3D data space (x, y, t), we perform two sequential Hough transforms for the two 2D data space (x, t) and (y, t) in a similar manner as in [55].

To generate the data space as the input for the Hough transform, we first remove the static clutter at each receiver by the proposed technique explained in Section 8.2. Next, an energy thresholding is applied on the clutter removed signals. Then, we perform the pixel-based localization technique in Section 6.2.2 to construct the potential target positions in the area.

The output of the pixel-based localization technique is collected over a history of scan times which is actually a sliding window. These points in the (x, y, t) data space provide the input for the Hough transform described in Algorithm 6.

In this algorithm, the (x, y, t) data space is projected to the (x, t) plane, and a 2D Hough transform is performed to extract the (x, t) lines. For each detected line, the related y coordinates are used to construct the (y, t) data space. Then, another 2D Hough transform is performed onto this space to extract the (y, t) lines. Since for each target trajectory line in the (x, t) space, there should be a related line in the (y, t) plane, the final target trajectory line in the (x, y, t) data space is made by combining the results of the two performed Hough transforms. The algorithm contains some additional processing that reduces the number of output lines by removing less probable ones.

8.7 Experimental Results on the Hough Transform Algorithm for Trajectory Extraction

To implement the algorithm described in Section 8.6, we considered the pixel size in the pixel-based localization technique as 15cm for both x and y directions. In order to calculate a given pixel propagation time, we assumed a pointwise target at the human average height of 1.5m at that pixel. For

Algorithm 6 The Hough Transform to Extract the Target Trajectory

1: Perform the Hough transform on the points in the (x, t) data space.

- Merge equal detected (x, t) lines and if one line includes the whole points of another line, discard the shorter line.
- Discard lines with large gaps in either of x or t dimensions.

2: For each surviving (x, t) line, perform the Hough transform on their corresponding points in the (y, t) data space.

- Merge equal detected (y, t) lines and if one line includes the whole points of another line, discard the shorter line.
- Discard lines with large gaps in either of y or t dimensions.

3: Form (x, y, t) lines from the surviving (y, t) lines and their (x, t) parent lines.

- Merge equal detected (x, y, t) lines and if one line includes the whole points of another line, discard the shorter line.
- If two lines have more than a certain number of common points, discard the shorter line.

4: For each surviving (x, y, t) line, calculate its smoothed x and y positions at all scan times in the sliding window, using (8.4) (for the (x, t) and (y, t) plane, respectively) and the line parameters.

- discard (x, y, t) lines if any of their x and y positions exceeds the surveillance area.
- discard lines if their corresponding velocity is not acceptable as the human walking speed in indoor applications.
- Assign lines with close x and y positions together in clusters and replace each cluster by one line with positions averaged over the cluster.

each of the Hough transforms, we discretized the parameter θ into 256 values from 0 to π. Also, we quantized the calculated range for the parameter ρ into 50 values.

Fig. 8.8(a) shows the output of the Hough transform described in Algorithm 6 before Step 2. In our case the scan number is actually the person position number. Performing the Hough transform in the (x, t) space results in the detection of 4 lines. Applying the Hough transform on the (y, t) space corresponding to each detected (x, t) line generates a total of 11 lines before Step 3 of Algorithm 6, as depicted in Fig. 8.8(b). However, the combination of both Hough transform outputs leads to a single detected line in the (x, y, t) space at the end of the algorithm, which is shown in Fig. 8.8(c).

8.8 Conclusion

In this chapter, the Hough transform was employed in two different approaches for tracking humans walking inside the surveillance area. In one approach, the Hough transform finds the TOA curves of the targets in the scan vs. propagation time image. In order to parametrize the target TOA curve, a straight-line constant-velocity model was adopted for the target movement, providing a reasonable computational complexity and data storage requirement.

In the other approach, the Hough transform finds the target trajectory in the (x, t) and (y, t) images provided by the pixel-based localization technique. Then, the results are combined to find the target trajectory in the (x, y, t) space.

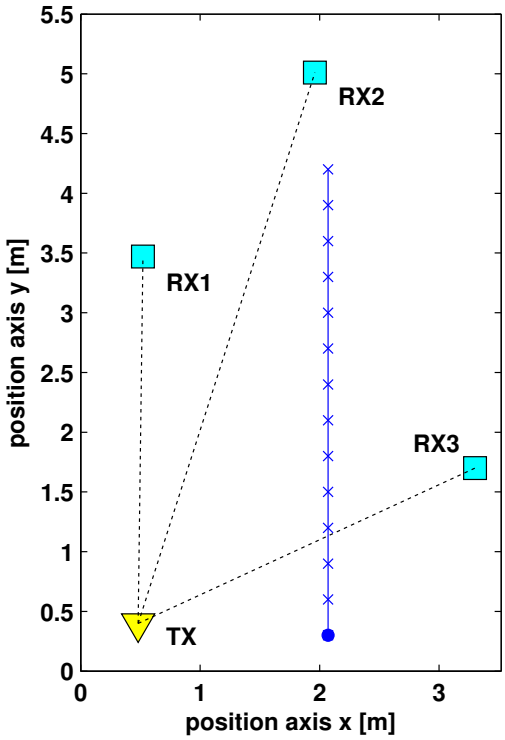
By collecting data over time, the Hough transform is able to perform target detection, data association and tracking, all at the same time. Another advantage is simultaneous estimation of the number of targets, track initiation and maintenance in multiple target scenarios.

In order to combat shadowing effects, particularly severe in indoor environments, a new empty-room clutter removal technique was proposed based on dissimilarities between the received signals with and without the target. We showed that the proposed approach can considerably reduce the number of false alarms compared to the conventional empty-room technique.

To evaluate the performance of all the above approaches, we provided experimental results by channel measurements in a $3.5m \times 5.5m \times 3m$ room, with the presence of significant clutter.

It should be noted that our measurements were carried out for a certain posture of the person, whereas the person may take various postures and orientations during the walk. Therefore, in order to design and implement

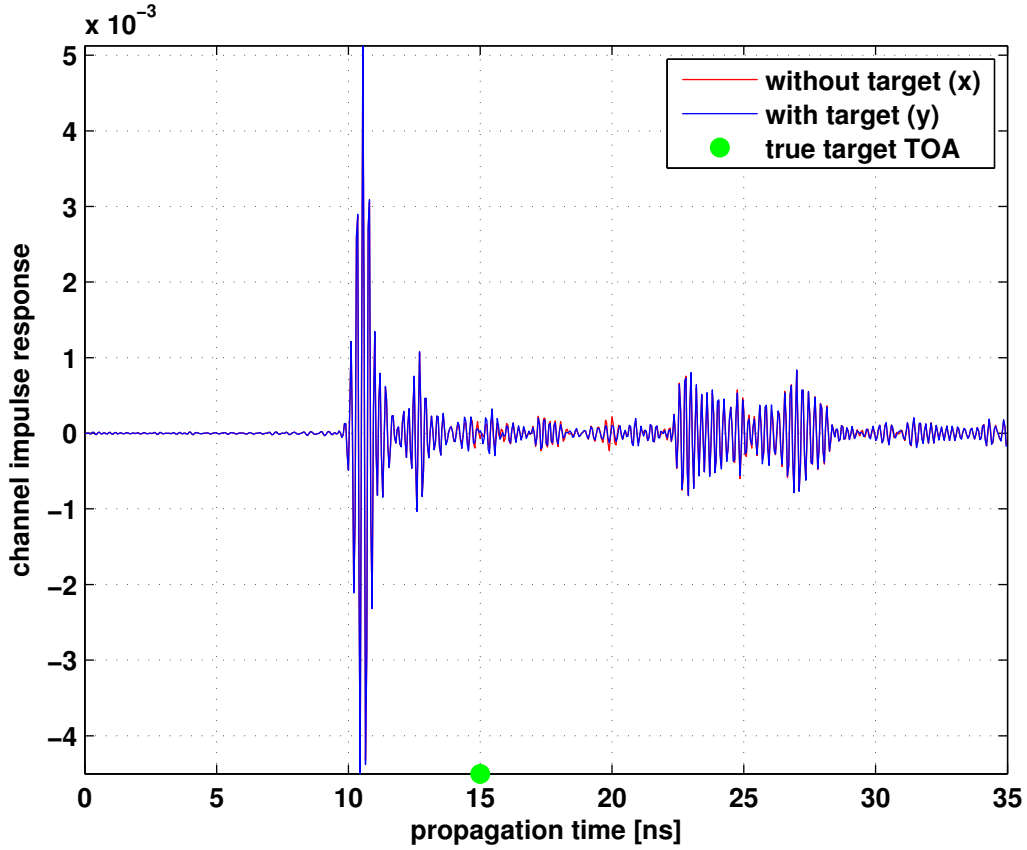
the system, proper statistical studies are required on the human radar-cross-section for multistatic UWB radars.



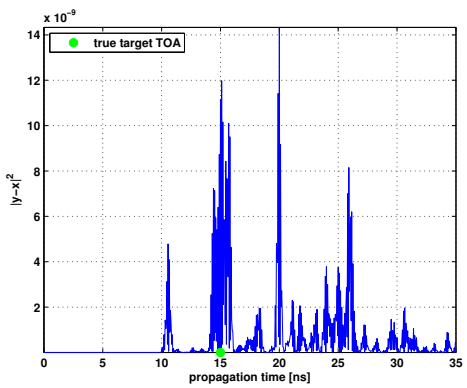
(a) Antenna positions and the target trajectory.

(b) The measurement room photo.

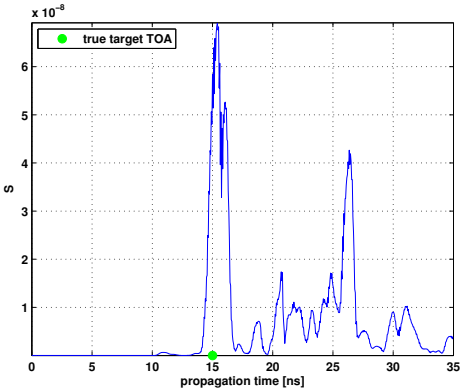
Figure 8.1: The measurement scenario.



(a) The channel impulse response.

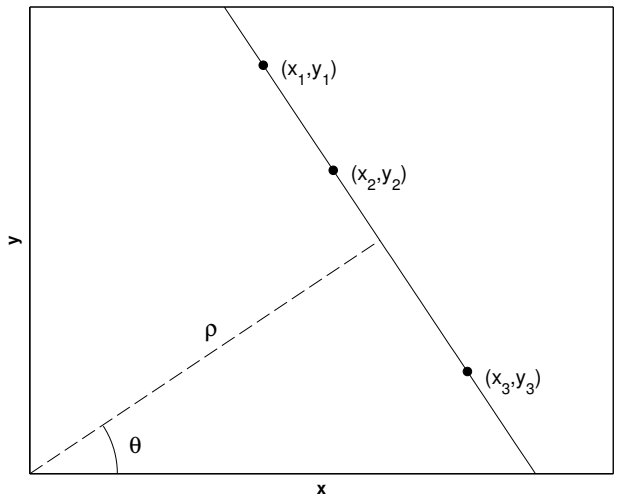


(b) Energy after the conventional empty-room.

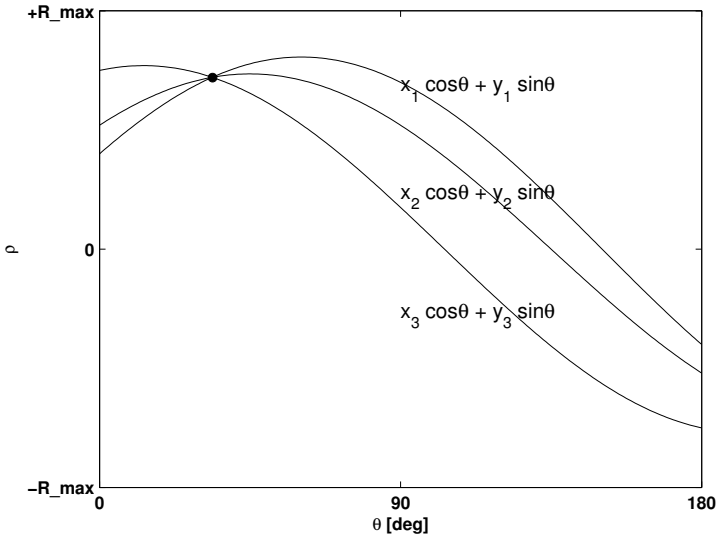


(c) Energy after the proposed empty-room.

Figure 8.2: Clutter removal in the presence of shadowing. System parameters are described in Section 8.5.



(a) The data space.



(b) The parameter space.

Figure 8.3: The Hough transform for straight line detection.

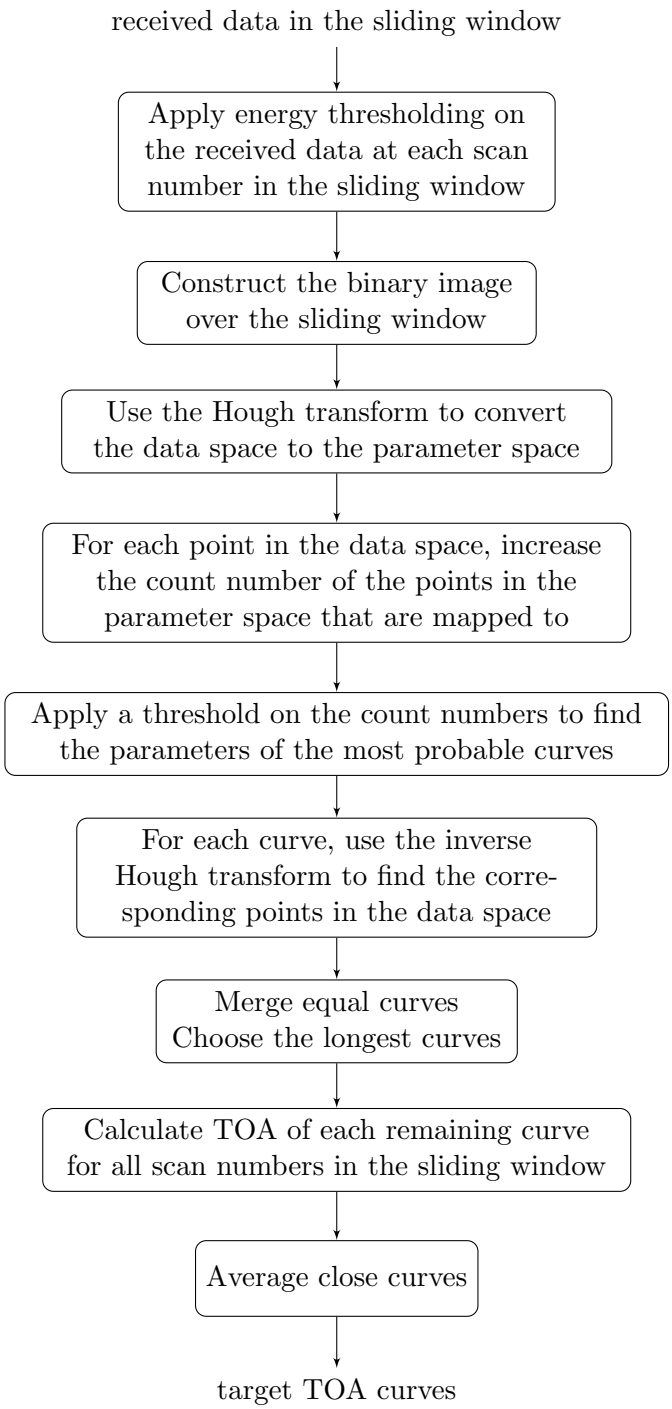
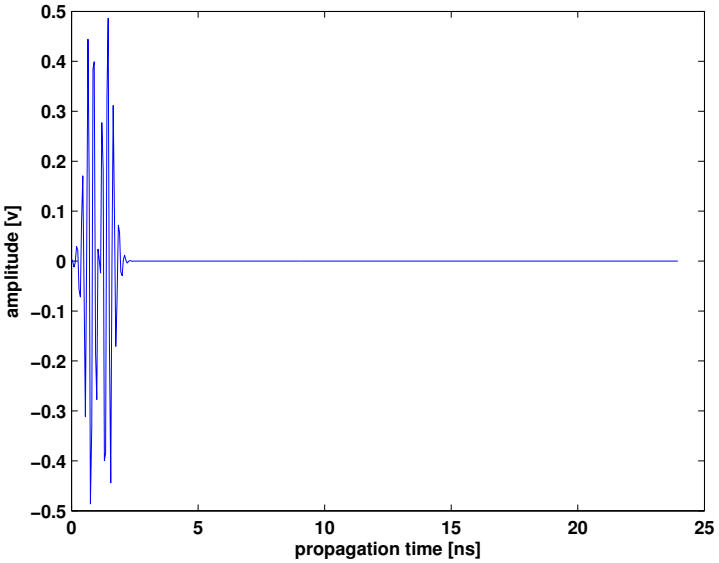
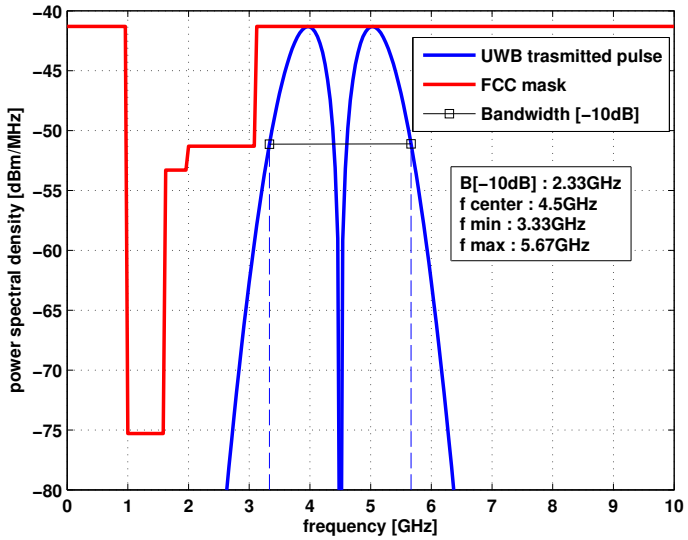


Figure 8.4: The flowchart of the proposed Hough transform TOA association algorithm.

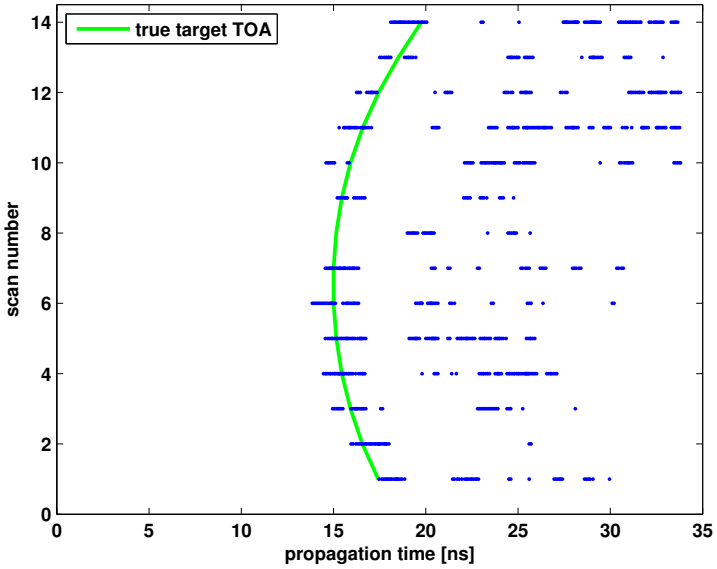


(a) The UWB transmitted pulse in time domain.

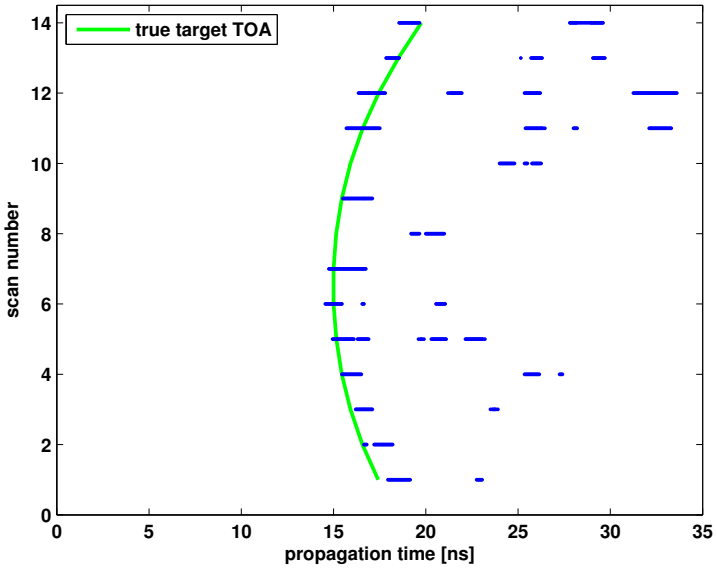


(b) The UWB transmitted pulse power spectral density.

Figure 8.5: The UWB designed transmitted pulse.

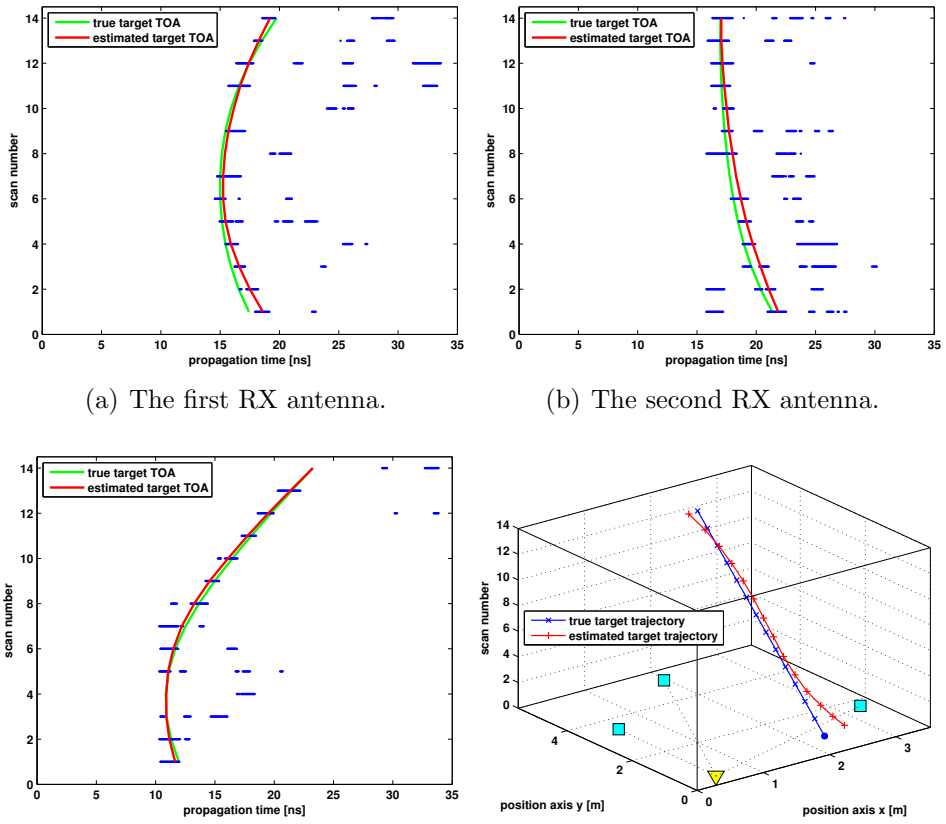


(a) The conventional empty-room approach.



(b) The proposed empty-room approach.

Figure 8.6: The radar image at the first RX antenna after energy thresholding.



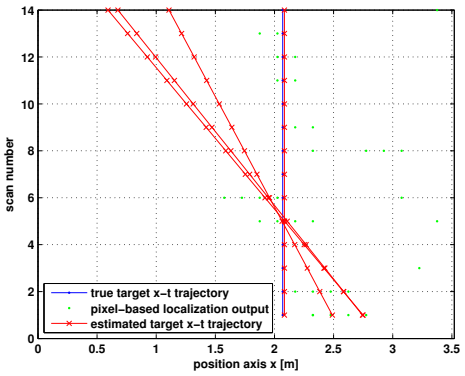
(a) The first RX antenna.

(b) The second RX antenna.

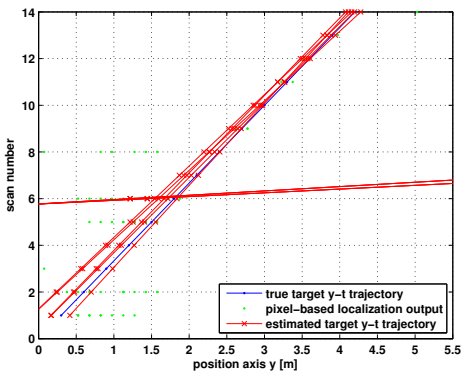
(c) The third RX antenna.

(d) the estimated target trajectory.

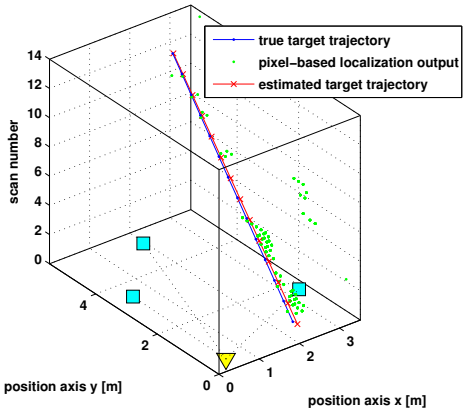
Figure 8.7: The proposed Hough transform TOA association technique.



(a) The Hough transform in the (x, t) data space.



(b) The Hough transform in the (y, t) data space.



(c) The combination of both Hough transforms in the (x, y, t) data space.

Figure 8.8: The Hough transform to extract the target trajectory.

Chapter 9

Conclusions and Future Works

In this thesis, we introduced a UWB multistatic radar as a good infrastructure for an anti-intruder system, due to a high resolution in range offered by the UWB impulse-radio and a high area coverage and spatial diversity provided by the multistatic configuration. Moreover, IR-UWB devices are usually light-weight and cost-effective and their low transmit power makes them appropriate for short-range monitoring applications. Chapter 2 described such a UWB multistatic radar system for surveillance applications.

We focused on the detection, localization and tracking of non-collaborative targets, typically human beings, moving inside a surveillance area for UWB multistatic radars. In order to improve the localization accuracy and achieve a higher level of system reliability, tracking is an essential part. However, there are a few researches in the literature studying the suitable tracking techniques in this context.

Since UWB radars can not be considered as linear Gaussian systems, the commonly used tracking filters, such as the Kalman filter, which were developed based on linearity and Gaussianity assumptions, are not expected to provide a good performance. In Chapter 3, we introduced the Bayesian filter as an appropriate choice for UWB radars. In the category of Bayesian filtering, we particularly focused on the grid-based method and particle filters as the two common implementations.

In Chapter 4, we proposed a particle filtering algorithm for tracking a single target in UWB multistatic radars composed of one transmitter and several receivers. The algorithm was developed in a way to overcome the detection problem in such systems, mainly caused by blind zones. We showed through numerical results that the proposed algorithm achieves a high estimation accuracy, even at low signal-to-noise ratios, in the presence of either static or dynamic clutter, and it works well for complicated target trajectories. We also showed that the real-time implementation of the proposed

tracking algorithm is feasible.

In Chapter 5, we compared our proposed particle filtering algorithm with the grid-based Bayesian filter for a single target scenario in terms of both the performance and computational complexity. The numerical results showed that the grid-based tracking method outperforms the particle filtering, especially around the blind zones, at the cost of a higher computational complexity and memory requirements.

Chapter 6 presented the CFAR detector and direct method of localization as the most common detection and localization techniques used in UWB multistatic radars. Then, we proposed a new CFAR detector based on a median filter and a pixel-based localization technique. The numerical results showed that the combination of these techniques achieves a low positioning error and a good tradeoff between false alarms and missed detections. Thus, it can be a good choice for UWB multistatic radar systems with the poor detection around the blind zones.

Chapter 7 extended the particle filtering algorithm proposed in Chapter 4 to the tracking problem of multiple targets. For this challenging problem, the whole tracking system components, namely the clutter removal, detection, localization, data association and tracking were implemented by numerical simulations.

Chapter 8 presented the experimental results provided by channel measurements in a multistatic antenna configuration. The measurements were carried out for tracking a person walking in an indoor area, with the presence of a dense clutter. We proposed a new empty-room clutter removal technique to cope with the shadowing effect in indoor environments. We showed that the proposed approach can greatly reduce the number of false alarms compared to the conventional empty-room technique. We also proposed a simultaneous detection, data association and tracking technique using the Hough transform on the data collected over time. In a multiple target scenario, the proposed approach is able to estimate the number of targets, perform the track initiation and maintenance, all at the same time.

9.1 Future Works

Although the most focus of this thesis was on the tracking phase in UWB multistatic radars, according to our observations based on experimental measurements, the major problem in such systems is actually the detection part. The poor detection problem is mainly caused by clutters which may hide the target echo by having higher radar-cross-sections, shadowing effects which are particularly severe in indoor environments, blind zones which are intrin-

sically associated with the multistatic configuration, direct path residue after clutter removal which may be much higher than the target echo, low transmit power which is restricted by FCC regulations, and the antenna beamwidth which may not fully cover the area.

Therefore, developing more effective clutter removal and detection techniques is still crucial for such systems to make the proposed tracking algorithms in this thesis more efficient.

Evaluating other multiple target tracking algorithms such as the probability hypothesis density (PHD) filter for UWB multistatic radars and comparing them with the proposed approaches in this thesis can be a useful direction of research.

The experimental results presented in the last chapter were provided by channel measurements using a network analyzer. Performing experimental measurements with a real IR-UWB radar system is essential for a more realistic study.

Furthermore, in order to design and implement the system, proper statistical studies are required on the bistatic human radar-cross-section, as the person may take various postures and orientations during the walk.

Studying the optimization techniques of the antenna configuration can also be helpful to better cope with the blind zone problem in a UWB multistatic radar system.

List of Figures

List of Algorithms

References

References [1] to [7] are in *Publications* sections of the thesis.

[8] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.

[9] W. Headley, C. da Suva, and R. Buehrer, "Indoor location positioning of non-active objects using ultra-wideband radios," in *Proc. IEEE Radio and Wireless Symp.*, Jan. 2007, pp. 105–108.

[10] E. Paolini, A. Giorgetti, M. Chiani, R. Minutolo, and M. Montanari, "Localization capability of cooperative anti-intruder radar systems," *EURASIP J. Advances in Signal Processing*, vol. 2008, pp. 1–14, 2008.

[11] V. S. Chernyak, *Fundamentals of Multisite Radar Systems*. Gordon and Breach Science Publisher, 1998.

[12] A. Haimovich, R. Blum, and L. Cimini, "MIMO radar with widely separated antennas," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2008.

[13] R. Zekavat and R. Buehrer, *Wireless Localization Using Ultra-Wideband Signals*, 1st ed. Wiley-IEEE Press, 2012.

[14] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009.

[15] S. Bartoletti, A. Giorgetti, and A. Conti, "Sensor radars with subset diversity," in *Proc. IEEE Workshop Advances in Network Localization and Navigation*, June 2013, pp. 32–36.

[16] M. Chiani and A. Giorgetti, "Coexistence between UWB and narrowband wireless communication systems," *Proc. IEEE*, vol. 97, no. 2, pp. 231–254, Feb. 2009.

[17] Y.-J. Park, K.-H. Kim, S.-B. Cho, D.-W. Yoo, D.-G. Youn, and Y.-K. Jeong, "Buried small objects detected by UWB GPR," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 10, pp. 3–6, Oct. 2004.

[18] A. Yarovoy, L. Ligthart, J. Matuzas, and B. Levitas, "UWB radar for human being detection," *IEEE Aerospace and Electronic Systems Magazine*, vol. 21, no. 3, pp. 10–14, March 2006.

[19] S. Chang, M. Wolf, and J. Burdick, "Human detection and tracking via ultra-wideband (UWB) radar," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2010, pp. 452–457.

[20] E. Staderini, "UWB radars in medicine," *IEEE Aerospace and Electronic Systems Magazine*, vol. 17, no. 1, pp. 13–18, Jan. 2002.

[21] M. Svecova, D. Kocur, R. Zetik, and J. Rovnakova, "Target localization by a multistatic UWB radar," in *20th International Conference Radioelektronika*, April 2010, pp. 1–4.

[22] Z. Xiao-wei, A. Gaugue, C. Liebe, J. Khamlichi, and M. Menard, "Through the wall detection and localization of a moving target with a bistatic UWB radar system," in *Proc. European Radar Conf. (EuRAD)*, Sep. 2010, pp. 204–207.

[23] S. Gezici, T. Zhi, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.

[24] W. Guohua, Z. Yuxiang, and W. Siliang, "Detection and localization of high speed moving targets using a short-range UWB impulse radar," in *Proc. IEEE Radar Conf.*, May 2008, pp. 1–4.

[25] K. Dae-Hyun, L. Dong-Woo, S. L. K. Hyung-Myung, W.Sung-Chul, and K. Hyun, "Localization methods of multi-targets for UWB radar sensor networks," in *Proc. Int. Asia-Pac. Conf. on Synth. Aperture Radar (APSAR)*, Sep. 2011, pp. 1–4.

[26] D. Hong, C. Chen, P. Shirui, L. Xin, and Z. Linhua, "Multistatic ultra-wideband localization for NLOS environments," in *Proc. Int. Conf. on Intelligent Sys. Design and Eng. App. (ISDEA)*, Jan. 2012, pp. 380–384.

[27] H. Yuan, T. Savelyev, and A. Yarovoy, "Two-stage algorithm for extended target tracking by multistatic UWB radar," in *IEEE CIE International Conference on Radar*, vol. 1, Oct. 2011, pp. 795–799.

[28] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009.

[29] A. Giorgetti and M. Chiani, "Time-of-arrival estimation based on information theoretic criteria," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 1869–1879, April 2013.

[30] E. Saberinia and A. Tewfik, "Ranging in multiband ultrawideband communication systems," *IEEE Trans. on Vehicular Technology*, vol. 57, no. 4, pp. 2523–2530, July 2008.

[31] G. Breed, "A summary of FCC rules for ultra wideband communications," in *High Frequency Electronics*, vol. 5, Jan. 2005, pp. 42–44.

[32] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Proc.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[33] A. Haug, *A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes.* The MITRE Corporation, Jan. 2005.

[34] T. Kai, W. Shiyou, C. Jie, and L. X. an F. Guangyou, "An efficient and low-complexity through wall moving target tracking algorithm by UWB radar," in *14th International Conference on Ground Penetrating Radar (GPR)*, June 2012, pp. 966–971.

[35] D. Kocur, J. Gamec, M. Svecova, M. Gamcova, and J. Rovnakova, "Imaging method: A strong tool for moving target tracking by a multistatic UWB radar system," in *IEEE 8th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Jan. 2010, pp. 11–19.

[36] M. Chiani, A. Giorgetti, M. Mazzotti, R. Minutolo, and E. Paolini, "Target detection metrics and tracking for UWB radar sensor networks," in *IEEE Int. Conf. on Ultra-Wideband (ICUWB)*, Sep. 2009, pp. 469–474.

[37] T. Thiasiriphet and J. Lindner, "Particle filtering for UWB radar applications," in *IEEE International Conference on Ultra-Wideband (ICUWB)*, Sep. 2011, pp. 248–252.

[38] D. Zaugg, A. Samuel, D. Waagen, and H. Schmitt, "A combined particle/kalman filter for improved tracking of beam aspect targets," in *IEEE Workshop on Statistical Signal Processing*, Sep-Oct. 2003, pp. 535–538.

[39] R. Herrmann, J. Sachs, M. Kmec, M. Grimm, and P. Rauschenbach, "Ultra-wideband sensor system for remote monitoring of vitality at home," in *9th European Radar Conference (EuRAD)*, 2012, pp. 234–237.

[40] E. Piuzzi, S. Pisa, P. D'Atanasio, and A. Zambotti, "Radar cross section measurements of the human body for UWB radar applications," in *IEEE International Instrumentation and Measurement Technology Conference*, 2012, pp. 1290–1293.

[41] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[42] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, July-Sep. 2003.

[43] S. Bartoletti, M. Guerra, and A. Conti, "UWB passive navigation in indoor environment," in *Proc. Int. Symp. on App. Sciences in Bio. and Comm. Tech.*, Barcelona, Spain, Oct. 2011, pp. 1–5.

[44] J. Rovnakova and D. Kocur, "TOA association for handheld UWB radar," in *Proc. 11th Int. Radar Symp. (IRS)*, 2010, pp. 1–4.

[45] J. Rovnakova and D. Kocur, "UWB radar signal processing for through wall tracking of multiple moving targets," in *European Radar Conf. (EuRAD)*, 2010, pp. 372–375.

[46] D. Urdzik and D. Kocur, "CFAR detectors for through wall tracking of moving targets by M-sequence UWB radar," in *Proc. 20th Int. Conf. Radioelektronika*, April 2010, pp. 1–4.

[47] A. Maali, A. Mesloub, M. Djeddou, G. Baudoin, H. Mimoun, and A. Ouldali, "CA-CFAR threshold selection for IR-UWB TOA estimation," in *Proc. 7th Int. Workshop on Systems, Signal Process. Appl. (WOSSPA)*, May 2011, pp. 279–282.

[48] M. Svecova, D. Kocur, and R. Zetik, "Object localization using round trip propagation time measurements," in *Proc. 18th Int. Conf. Radioelektronika*, April 2008, pp. 1–4.

[49] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*. McGraw-Hill, 1995.

[50] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, 2004.

[51] R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Communication of the ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[52] B. Carlson, E. Evans, and S. Wilson, "Search radar detection and track with the Hough transform. Part I: system concept," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 102–108, Jan. 1994.

[53] B. Carlson, E. Evans, and S. Wilson, "Search radar detection and track with the Hough transform. Part II: detection statistics," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 109–115, Jan. 1994.

[54] B. Carlson, E. Evans, and S. Wilson, "Search radar detection and track with the Hough transform. Part III: detection performance with binary integration," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 1, pp. 116–125, Jan. 1994.

[55] A. Moqiseh and M. Nayebi, "Combinational Hough transform for surveillance radar target detection in a 3-D data map," in *IEEE Radar Conf.*, May 2008, pp. 1–6.

[56] L. Moyer, J. Spak, and P. Lamanna, "A multi-dimensional Hough transform-based track-before-detect technique for detecting weak targets in strong clutter backgrounds," *IEEE Transanctions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 3062–3068, Oct. 2011.

[57] Z. Jiankui and H. Zishu, "Detection of weak target for MIMO radar based on Hough transform," *Journal of Systems Engineering and Electronics*, vol. 20, no. 1, pp. 76–80, Feb. 2009.

[58] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing for Computer Vision*, 3rd ed. Academic Press, Oct. 2012.

[59] S. W. Yankowich and M. Farooq, "A Hough transform based multisensor, multitarget track initiation technique," in *Proc. 36th IEEE Conf. on Decision and Control, 1997.*, vol. 5, Dec. 1997, pp. 5018–5023.

[60] Y. Okamoto, I. Matsunami, and A. Kajiwara, "Moving vehicle discrimination using Hough transformation," in *IEEE Radio and Wireless Symposium (RWS)*, Jan. 2011, pp. 367–370.

[61] Y. Okamoto, I. Matsunami, and A. Kajiwara, "Pedestrian and two-wheeler detection using ultra-wideband vehicular radar," in *IEEE Sensors Applications Symposium (SAS)*, Feb. 2012, pp. 1–4.

[62] M. Mohammadi, A. Moqiseh, H. Gheidi, and M. Nayebi, "Noncoherent integration of UWB radar signals using the Hough transform," in *European Radar Conference (EuRAD)*, Oct. 2008, pp. 9–12.

[63] F. Abujarad, A. Jostingmeier, and A. Omar, "Clutter removal for landmine using different signal processing techniques," in *Tenth Int. Conf. on Ground Penetrating Radar (GPR)*, June 2004, pp. 697–700.