Department of Electromagnetic Field
Faculty of Electrical Engineering
Czech technical university in Prague

Multipath Assisted Positioning using Machine Learning

Doctoral thesis

Author: Ing. Rostislav Karásek
Supervisor: Prof. Ing. Pavel Pechač, Ph.D.
Co-Supervisor: Dr.-Ing. Christian Gentner

Study Programme: Electrical Engineering and Information Technology (P2612)
Specialization: Radioelectronics (2601V010)

September, 2022

Abstract

The multipath propagation of the radio signal was considered a problem for positioning systems that had to be eliminated. However, a groundbreaking new approach called multipath assisted positioning caused a paradigm shift, where multipath propagation improves the positioning performance. Moreover, the multipath assisted positioning algorithm called Channel-SLAM shows the possibility of using a single physical transmitter in a multipath environment for positioning. In this thesis, I open a discussion on some problems that have vital importance for multipath assisted positioning algorithms with a focus on pedestrian positioning. Using the idea of multipath assisted positioning, I present a single frequency network positioning algorithm. I evaluated the single frequency network-based positioning algorithm for positioning in a real scenario using a terrestrial digital video broadcasting transmission. I propose a novel pedestrian transition model utilizing the inertial measurements from a handheld inertial measurement unit. The proposed pedestrian transition model improves the precision and reliability of the Channel-SLAM. Comparing the proposed transition model with the Rician transition model previously used in Channel-SLAM quantifies the performance improvement. This thesis proposes a joint data association technique that overcomes the strong dependence on the radio channel estimation algorithm used in Channel-SLAM. The joint data association allows reusing the previously observed virtual transmitters after an outage of multipath component tracking. The evaluation based on the walking pedestrian scenario shows that the joint data association algorithm provides superior positioning precision. The virtual transmitter position estimation yields a significant computational load in Channel-SLAM. I propose a method that represents the virtual transmitter by a Gaussian mixture model and learns its parameters. The evaluation shows that the proposed method outperforms the previous approach while decreasing the computational load. Also, the current methods for radio channel estimation yield a considerable computational load that prohibits a real-time deployment. The thesis investigates the possibility of using artificial neural networks trained to estimate the number of multipath components and corresponding delays in a noisy measurement of a channel impulse response. The artificial neural network-based delay estimator provides a superresolution performance and faster runtime than the classical approaches. The precision of the trained artificial neural network architecture is evaluated and compared to the Cramer-Rao lower bound theoretical limit and classical channel estimation algorithms.

Keywords: Channel-SLAM, convolutional neural network, data association, Gaussian mixture model, line spectral estimation, machine learning, multipath assisted positioning, multipath radio channel, multiple hypothesis tracking, noise regularization, particle filter, pedestrian navigation, positioning, radio channel parameter estimation, signals of opportunity, ultra-wideband technology.

Abstrakt

Vícecestné šíření rádiového signálu bývalo považováno za problém pro určování polohy, který bylo nutné potlačit. Avšak převratný nový přístup určování polohy s využitím vícecestného šíření způsobil posun paradigmatu, kdy vícecestné šíření vylepšuje přesnost určování polohy. Algoritmus určování polohy s využitím vícecestného šíření nazvaný Channel-SLAM navíc ukazuje možnost použití jediného fyzického vysílače v prostředí s vícecestným šířením pro určování polohy. V této práci otevírám diskusi k některým problémům, které mají zásadní význam pro algoritmy určování polohy s využitím vícecestného šíření se zaměřením na určování polohy chodců. S využitím myšlenky určování polohy pomocí vícecestného šíření předkládám algoritmus určování polohy v jednofrekvenční síti. Navržený algoritmus určování polohy v jednofrekvenční síti jsem vyhodnotil v reálném případě s využitím pozemního digitálního vysílání televize. Dále v práci navrhuji pohybový model pro chodce využívající inerciální měření z příruční inerciální jednotky. Navržený pohybový model pro chodce vylepšuje přesnost a spolehlivost určování polohy pomocí Channel-SLAM. Poskytnuté porovnání s Riceho pohybovým modelem používaným v Channel-SLAM kvantifikuje zlepšení přesnosti navrženým pohybovým modelem. Tato práce navrhuje asociaci dat založenou na sdružené hustotě pravděpodobnosti, která překonává silnou závislost Channel-SLAM na algoritmu pro odhad a sledování parametrů rádiového kanálu. Asociace dat založená na sdružené hustotě pravděpodobnosti umožňuje opětovné využití dříve pozorovaných virtuálních vysílačů po výpadku sledování komponent vícecestného šíření. Vyhodnocení experimentu reálné chůze chodců ukazuje, že algoritmus asociace dat na bázi sdružené hustoty pravděpodobnosti překonává předchozí metodu. Odhad polohy virtuálních vysílačů představuje extrémní výpočetní zátěž v Channel-SLAM. Proto navrhuji metodu, která modeluje virtuální vysílač jako směs Gaussovských distribucí a zároveň se učí její parametry. Vyhodnocení ukazuje, že navržená metoda přesností překonává předchozí přístup a zároveň snižuje výpočetní zátěž. Současné metody pro odhad parametrů rádiového kanálu také představují značnou výpočetní zátěž znemožňující nasazení v reálném čase. Proto tato práce zkoumá možnost využití umělých neuronových sítí k odhadu počtu komponent vícecestného šíření a jejich odpovídajících časových zpoždění při měření impulsní odezvy komunikačního kanálu s vícecestným šířením a rádiovým šumem. Odhad zpoždění založený na umělých neuronových sítích poskytuje rozlišení překonávající Rayleighův limit, tzv. super-rozlišení, a je rychlejší než klasické metody. Přesnost odhadu pomocí natrénované umělé neuronové sítě je porovnána s teoretickým limitem, daným Cramer-Rao dolní mezí, a klasickými algoritmy odhadu parametrů komunikačního kanálu.

Klíčová slova: asociace dat, Channel-SLAM, částicový filtr, konvoluční neuronové sítě, navigace pro chodce, odhad parametrů rádiového kanálu, odhad čárového spekta, rádiový kanál s vícecestným šířením, regularizace šumem, signály příležitosti, sledování více hypotéz, směs Gassovských distribucí, strojové učení, ultraširokopásmová technologie, určování polohy, určování polohy pomocí vícecestného šíření.

Declaration

I hereby declare I have written this doctoral thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis.

In Prague, 30th September, 2022.

Contents

Chapter 1

Introduction

The increasing demand for fast and convenient delivery of large volumes of goods while decreasing the number of necessary human workers puts high weight on developing precise, reliable, and cheap positioning systems. Since a large portion of goods handling happens inside large warehouses, where Global Navigation Satellite System (GNSS)s do not work, we must develop alternative positioning systems with indoor capability. At the same time, the indoor positioning system must be convenient to deploy and occupy minimal space. Creating an indoor positioning system for a large warehouse by reusing the principle of GNSSs would require using many transmitters. This would be expensive, take a large portion of the warehouse space, and unreliable due to signal reflection and scattering. Addressing these problems requires changing the way of thinking about positioning systems. That is what the newly developing research of multipath assisted positioning algorithms do. It breaks the lived-in stereotypes of classical positioning systems and turns problems into features.

The core principle of positioning has been unchanged throughout human history. Observation of known waypoints allows obtaining an observer's position. Suppose the lost observer can spot three distinct objects of the known position. In that case, he can calculate the distance to the objects by triangulation and then use the triangulated distances to calculate his position using trilateration. The distance calculation can be cumbersome due to the required angle measurements. However, modern technology's precision allows measuring the distance directly without triangulation using a precise time measurement. Today, the waypoint of a know location is transmitting a radio signal that the lost observer receives. The observer obtains the transmitter distance as a difference between the radio signal's Time of Arrival (ToA) and the time the radio signal was transmitted multiplied by the speed of light. An alternative time-based positioning method does not require knowing when the radio signal was transmitted. When transmitters start the transmission simultaneously, the position is obtained from radio signal Time Difference of Arrival (TDoA) by constructing hyperbolas. Each hyperbola has its

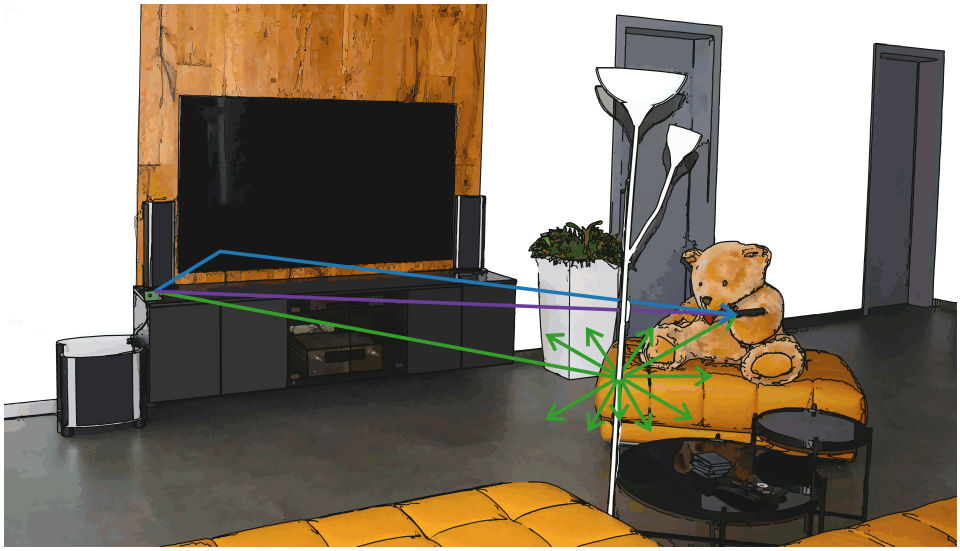


Figure 1: Typical indoor multipath scenario with one transmitter and one receiver. The purple line shows the LoS propagating directly from the transmitter to the receiver. The blue line represents the NLoS propagating towards the receiver via reflection from the wall-like obstacle. Finally, the green line shows the scattering of the transmitted signal on an electrically small obstacle. In this case, the transmitted signal impinges the lamp, which scatters the signal in all directions. The receiver receives this MPC scattered by the lamp.

focus point located on the transmitter position. The observer stands at the intersection of the hyperbolas.

The shift from visual observation to radio signal measurement allows positioning in difficult visual conditions, e.g., at night, in fog, or at extreme distances using small waypoints. However, to measure the ToA correctly, the radio signal must propagate directly between the transmitter and receiver on the so-called Line-of-Sight (LoS). The surrounding environment can reflect and scatter the transmitted radio signal in multiple directions. Then, the receiver can receive the radio signal even if the LoS is obstructed. We call such an indirectly received radio signal a Non-Line-of-Sight (NLoS) component, and we need to recognize and separate it from LoS. Since the multiple radio signal replicas propagate toward the receiver along different paths, we call this principle multipath propagation and each signal replica Multipath Component (MPC). The surrounding environment channels the transmitted radio signal towards the receiver while distorting it. Hence, the mathematical model of the radio signal distortion is called the radio channel model. The multipath radio channel model describes the radio signal distortion when propagated towards the receiver along multiple paths. Fig. 1 shows a typical indoor multipath propagation scenario.

It can be that multiple transmitters are transmitting the same radio signal simultaneously in so-called Single Frequency Network (SFN). Then, the receiver receives a sum of the delayed radio signal replicas that the receiver must separate to measure the ToA from the individual transmitters. The effect of SFN is indistinguishable from multipath propagation, and the multipath radio channel model describes the influence of the SFN.

Historically, the multipath propagation has been seen as an undesired effect decreasing the positioning precision. The positioning algorithm had to suppress the influence of multipath propagation before calculating the position. However, this paradigm is changing with the development of multipath assisted positioning algorithms. The multipath assisted positioning algorithms do not need to suppress the radio channel influence. Moreover, it uses the information about the environment in the multipath radio channel to improve the precision of position estimation. The reflection and scattering of the radio signal are imagined as additional Virtual Transmitters (VTs). Hence, the multipath assisted positioning methods allow estimating position using even a single transmitter while utilizing the multipath radio channel information. In the beginning, the multipath assisted positioning methods needed to know the environment before positioning. However, the latest multipath assisted positioning methods use a Simultaneous Localization And Mapping (SLAM) approach to estimate the environment at the same time while estimating position.

This vast step in radio signal-based positioning research provides what a modern user requires from a positioning system. It works indoors and reduces the required number of transmitters, which decreases the price and space requirements. Moreover, automatic environment estimation simplifies the deployment. The main drawback of multipath assisted positioning algorithms is a high calculational load which currently does not allow a real-time application. This thesis addresses the calculational complexity, and other drawbacks of the current multipath assisted positioning algorithms.

1.1 Multipath Radio Channel

The multipath radio channel is a mathematical model widely used to model radio signal propagation in a complex environment. The environment consists of a medium where the radio signal propagates, and obstacles distort the transmitted radio signal. Hence, the received signal is an imperfect replica of the transmitted signal. The transmitted signal propagates with a finite speed through the medium, causing a delay between transmission and reception. The part of the transmitted signal propagating directly toward the receiver is called LoS.

The obstacles in the environment are separated into wall-like obstacles and scatterers. The interaction between the propagated radio signal and the wall-like obstacle is modeled as a plane wave reflection and refraction on a boundary between two media using Snell's law.

In this work, I model the scatterers as electrically small scatterers, which scatter the impinging radio signal in all directions. The size of the scatterer is comparable to or smaller than the wavelength of the transmitted signal.

The reflection and scattering effect creates replicas of the transmitted signal received as so-called NLoS components. The delay of the NLoS components is always larger than the delay of the LoS component. The received LoS and NLoS replicas of the transmitted signal are called MPCs.

Furthermore, each MPC consists of multiple sub-paths caused by propagation through a turbulent medium, scattering on rough surfaces of the obstacles, or subsurface scattering. The sub-path delays differ from the corresponding MPC delay less than the reciprocal of the signal bandwidth and are assumed as a random part of the MPC. The random part of the MPC is called Dense Multipath Component (DMC). Finally, the last source of distortion is Additive White Gaussian Noise (AWGN) added to the received signal by the receiver.

According to the definition of the multipath radio channel above, the received radio signal consists of delayed replicas of the transmitted signal. It was shown in [Bel63] that the channel can be assumed wide-sense stationary with uncorrelated scattering, or Wide-Sense Stationary Uncorrelated Scattering (WSSUS) channel for a short period of time. The WSSUS channel is linear and does not depend on the choice of the transmitted signal waveform. The mathematical description of the received signal propagated through a multipath radio channel is given by

$$s\left(t\right)=\sum_{\ell=1}^{L=\infty}\alpha_{\ell}\left(t\right)r\left(t-\tau_{\ell}\left(t\right)\right)+n\left(t\right), \tag{1.1}$$

where $s\left(t\right)$ is the received signal at time t, $r\left(t\right)$ is the transmitted signal, $\tau_{\ell}(t)$ is the delay of the ℓ-th MPC, and $\alpha_{\ell}\left(t\right)$ is the complex-valued amplitude of the ℓ-th MPC. The model order of the multipath radio channel is given by the number of MPCs by L. Finally, $n\left(t\right)$ is a zero-mean AWGN with variance $\sigma_{n}^{2}\left(t\right)$.

Based on WSSUS channel, the amplitude, delays, and noise variance can be assumed constant for a short period of time, $\alpha_{\ell}\left(t\right)=\alpha_{\ell}$, $\tau_{\ell}(t)=\tau_{\ell}$, and $\sigma_{n}^{2}\left(t\right)=\sigma_{n}^{2}$. The model order can be assumed finite $L<\infty$ using a standard approximation ignoring diffuse scattering, parts of diffracted waves, etc. [Mol09]. Hence, the mathematical model of the multipath radio channel is defined by

$$s\left(t\right)=\sum_{\ell=1}^{L}\alpha_{\ell}r\left(t-\tau_{\ell}\right)+n, \tag{1.2}$$

for a short period of time. The same multipath radio channel can be characterized by its Channel Impulse Response (CIR)

$$h\left(\tau\right)=\sum_{\ell=1}^{L}\alpha_{\ell}\delta\left(\tau-\tau_{\ell}\right), \tag{1.3}$$

where $\delta\left(\cdot\right)$ is a Dirac delta distribution. Then, the received signal is given as a convolution between the transmitted signal $r\left(t\right)$ and CIR as

$$s\left(t\right)=r\left(t\right)*h\left(\tau\right)+n=\tilde{s}\left(t\right)+n, \tag{1.4}$$

where $\tilde{s}\left(t\right)$ is a noise-free received signal.

Any real transmitter can only transmit a time- and band-limited signal. Additionally, finite CIR allows sampling of the received signal without loss of information [Boa15]. Hence, the sampled received signal can be written in a vector form as

$$\boldsymbol{s} = [s_0, \ldots, s_m, \ldots, s_{M-1}]^{\mathrm{T}}, \tag{1.5}$$

where m indexes the equidistant samples with sampling period T. The total number of samples is M. The transpose operation is marked by T. The noise-free sampled received signal can be defined similarly to $\boldsymbol{s}$ as

$$\tilde{\boldsymbol{s}} = [\tilde{s}_0, \ldots, \tilde{s}_m, \ldots, \tilde{s}_{M-1}]^{\mathrm{T}}. \tag{1.6}$$

The model parameters consisting of all delays

$$\boldsymbol{\tau} = [\tau_1, \ldots, \tau_\ell, \ldots, \tau_L]^{\mathrm{T}} \tag{1.7}$$

and all complex-valued amplitudes

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_\ell, \ldots, \alpha_L]^{\mathrm{T}} \tag{1.8}$$

describe the sampled received signal in the frequency domain

$$\boldsymbol{S} = \boldsymbol{B}(\boldsymbol{\tau})\,\boldsymbol{\alpha} + \boldsymbol{n} = e^{-j2\pi \boldsymbol{F}\boldsymbol{\tau}^{\mathrm{T}}}\boldsymbol{\alpha} + \boldsymbol{n} = \tilde{\boldsymbol{S}} + \boldsymbol{n}, \tag{1.9}$$

where $\boldsymbol{B}(\boldsymbol{\tau})$ is a nonlinear function describing the multipath radio channel structure with $\boldsymbol{F} = [0, \ldots, m, \ldots, M-1]^{\mathrm{T}}$, and $\tilde{\boldsymbol{S}}$ is noise-free frequency spectrum of $\tilde{\boldsymbol{s}}$. Finally, the sampled received signal $\boldsymbol{s}$ is obtained from $\boldsymbol{S}$ as

$$\boldsymbol{s} \xleftarrow{\mathfrak{F}^{-1}} \boldsymbol{S}, \tag{1.10}$$

where $\mathfrak{F}^{-1}$ is the Inverse Fast Fourier Transform (IFFT).

The multipath radio channel model based on the sum of individual MPCs [Tur72] was obtained from a long-distance propagation multipath model [Tur56]. However, it must be added that the model assumes that the transmitted signal waveform is narrowband compared to the carrier frequency. This work uses an Ultra-WideBand (UWB) signal with a bandwidth less than 15% of the carrier frequency. The experiment in [CWM02] shows that the channel frequency selectivity can be assumed to be the same for all MPCs in an office-like environment. Hence the presented multipath radio model fits this work.

Additionally, the used multipath radio model ignores some negligible or unobservable parameters in this work. The Doppler shift is negligible in a case of a slowly walking pedestrian, which is the case assumed throughout this work. The Angle of Arrival (AoA) and Angle of Departure (AoD) are unobservable when a single antenna transmitter and receiver antenna are used.

1.2 Multipath Radio Channel Parameter Estimation

A sparse set of delay (1.7) and complex-valued amplitude (1.8) parameters parametrize the multipath radio channel (1.2). Except for the channel modeling, the channel parameters $\boldsymbol{\tau}$ can be utilized, e.g., for ToA or TDoA positioning, and channel parameters $\boldsymbol{\alpha}$ can be used for Received Signal Strength Indication (RSSI) based positioning.

The multipath radio channel parameters cannot be obtained perfectly because the received signal is band-limited and corrupted by noise. The multipath radio channel parameters can be estimated using an estimator. Before introducing some well-known channel parameter estimators, I must introduce the theoretical limits of achievable precision known as the Cramer-Rao Lower Bound (CRLB). Our study focuses on sufficiently large Signal-to-Noise Ratios (SNRs), where it is possible to achieve the CRLB [Cra46]. If the SNR is close to or lower than 0 dB, the Ziv-Zakai bound [IWW83; Ian82] must be used because the CRLB is no longer achievable.

The CRLB gives the minimum variance any unbiased estimator can achieve. The derivation for a delay estimation is well studied and can be found, e.g., in [FF95; SJL97; Ric05]. According to [Ric05], the minimum variance that any nonlinear channel parameter estimator can achieve is given as

$$CRB_{\hat{\tau}} = \frac{\sigma_n^2}{\|\alpha\|^2} \frac{6}{M\left(M^2-1\right)}, \tag{1.11}$$

while the minimal variance of the linear parameter magnitude estimator is

$$CRB_{\|\hat{\alpha}\|} = \frac{\sigma_n^2}{2M}, \tag{1.12}$$

and the minimal variance of the linear parameter phase estimator is

$$CRB_{\angle\hat{\alpha}} = \frac{\sigma_n^2}{\|\alpha\|^2} \frac{1}{2M}. \tag{1.13}$$

The CRLB definition drops the MPC coefficient L for clarity.

For a sufficiently large SNR, the CRLB performance is achievable by a Maximum Likelihood (ML) estimator. The likelihood function of the multipath radio channel model defined as

$$\mathrm{p}\left(\boldsymbol{S} \mid \boldsymbol{\tau}, \boldsymbol{\alpha}, \sigma_n^2\right) = \frac{1}{\left(\pi\sigma_n^2\right)^M} \exp\left(-\frac{1}{2\sigma_n^2}\left(\boldsymbol{S} - \tilde{\boldsymbol{S}}\left(\boldsymbol{\tau}, \boldsymbol{\alpha}\right)\right)^{\mathrm{H}}\left(\boldsymbol{S} - \tilde{\boldsymbol{S}}\left(\boldsymbol{\tau}, \boldsymbol{\alpha}\right)\right)\right), \tag{1.14}$$

where H marks conjugate transpose, can be used to derive ML estimators $\hat{\boldsymbol{\tau}}$ and $\hat{\boldsymbol{\alpha}}$ as

$$\hat{\boldsymbol{\tau}} = \arg\max_{\boldsymbol{\tau}} \boldsymbol{S}^{\mathrm{H}} \boldsymbol{B}\left(\boldsymbol{\tau}\right) \hat{\boldsymbol{\alpha}}, \tag{1.15}$$

where the $\arg\max_{\boldsymbol{\tau}}$ returns $\boldsymbol{\tau}$, for which the function $\boldsymbol{S}^{\mathrm{H}} \boldsymbol{B}\left(\boldsymbol{\tau}\right) \hat{\boldsymbol{\alpha}}$ is maximized, and

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{B}^{\mathrm{H}}\left(\boldsymbol{\tau}\right) \boldsymbol{B}\left(\boldsymbol{\tau}\right)\right)^{-1} \boldsymbol{B}^{\mathrm{H}}\left(\boldsymbol{\tau}\right) \boldsymbol{S}, \tag{1.16}$$

where (1.16) is a Best Linear Unbiased Estimator (BLUE) of linear parameters $\boldsymbol{\alpha}$ [Ric05].

Solving (1.15) is a multidimensional optimization problem, which is generally hard to solve. Without any additional assumptions about $\boldsymbol{B}(\boldsymbol{\tau})$, the problem could be solved only by an exhaustive search over the space of $\boldsymbol{\tau}$, which has exponential complexity in the number of dimensions of $\boldsymbol{\tau}$.

The estimation and classification of a signal mixture as a subject of research gained importance with the start of the radar technique. Many approaches to the radio channel parameter estimation problem emerged during that time. The historical approach to performing the estimation of the frequency and wavenumber spectrum [Cap69] showed that it is possible to efficiently estimate the desired spectrum with higher precision than the spacing given by the sampling of measured data.

The significant improvement of the Capon's method [Cap69] was introduced by the MUltiple SIgnal Classification (MUSIC) algorithm [Sch86], where an effective method of classification based on generalized eigenvalue decomposition was proposed.

The main drawbacks of the MUSIC algorithm were computational complexity, a requirement for known sensor position, and careful calibration of the array. The Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm partly overcomes those drawbacks [RK89], reducing computational complexity via constraining the array structure. It requires pairs of identical elements, reducing the search space, and thus reducing the computational complexity. Moreover, it is calculating components directly rather than by optimization. The ESPRIT complexity is cubic in the number of sources, while MUSIC yields exponential complexity.

The algorithm family above assumes narrowband sources of signal and calibrated antenna arrays. In a case when the wideband measurement is required, the Expectation-Maximization (EM) algorithms family [DLR77] is applicable. The family of EM algorithms aims to estimate the deterministic parameters via alternating between two steps assuring convergence of the algorithm with respect to Maximum A Posteriori (MAP) or ML criterion of the estimated parameters. Many fields of science are still using the original algorithm. However, several variations of the EM algorithm are improving the convergence of the optimization process.

One of the variations is a Space-Alternating Generalized Expectation-maximization (SAGE) algorithm, reducing the complexity by estimating the parameters sequentially in small groups [FH94]. This method is advantageous for an estimation of the MPCs' parameters since the individual MPCs can be modeled as statistically independent, which is given by an uncorrelated scattering assumption [Rap96]. Hence, it is possible to estimate the parameters of each MPC independently using the SAGE [Fle+99].

Though, when we desire to track the parameters of MPCs' in a multipath radio channel characterized by a time-variant CIR for a longer period of time, the SAGE algorithm becomes computationally extensive and suboptimal. The suboptimality comes from the fact that the consecutive snapshots of CIR are statistically dependent, but SAGE estimates the parameters for each snapshot independently. The correlation between the individual snapshots of CIR yields additional information that can be used for smoothing the measurements using Bayesian filtering methods. An example of this approach

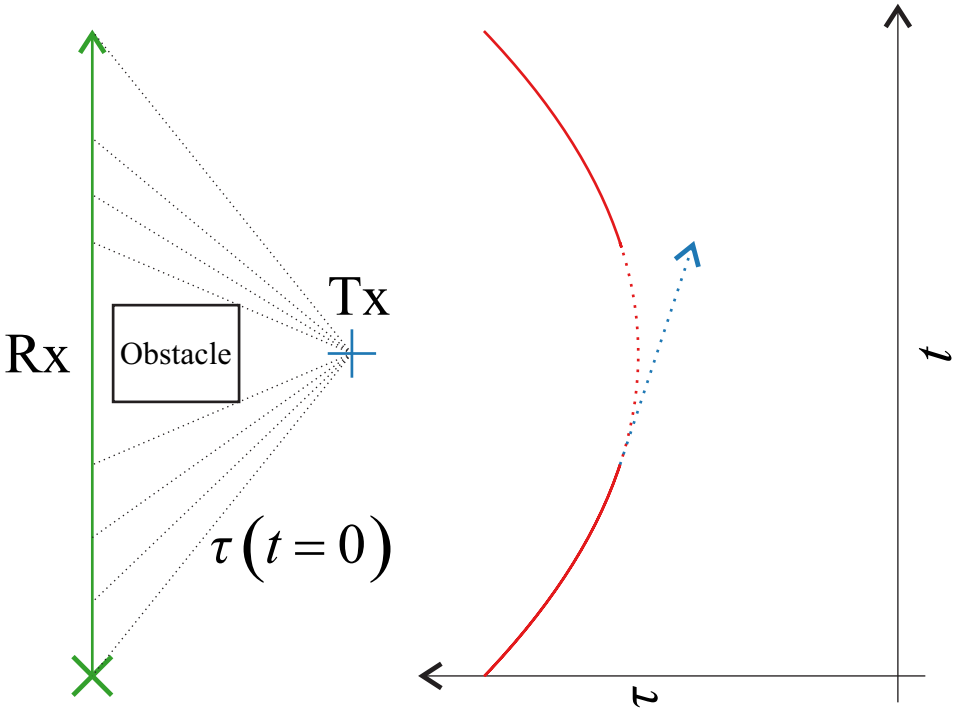


Figure 2: Illustration of the data association problem during the radio channel estimation.

is the Kalman Enhanced Super Resolution Tracking (KEST) algorithm [Jos+12], which combines SAGE with Kalman Filter (KF) to track MPCs over time.

However, the MPC parameters are a potentially nonlinear function of time when a mobile receiver observes the multipath radio channel. Moreover, outages in MPC observation occur when the received signal strength becomes too low due to, e.g., shadowing or large signal propagation distance. When a longer outage in MPC observation occurs, the MPC track is lost, and KEST cannot associate it with a previously observed track when reoccurring. Hence, an optimal multipath radio channel estimation algorithm requires a data association algorithm to associate MPCs to its previously observed tracks.

A simple example can illustrate that a sequential channel estimator is not sufficient for the MPC association. Fig. 2 shows a scenario where a receiver (Rx) moves with a constant speed along a straight trajectory. The green line shows the Rx trajectory, and the red line shows the corresponding delay. At some point, the transmitter (Tx) is behind an obstacle, which causes an outage in the delay τ measurement. The red dotted line shows the true but unobserved delay. The blue dotted line in Fig. 2 shows the delay predicted by a linear model of KF after an outage. After the Rx passed the obstacle, the difference between the observed and predicted delay is significant. Associating the reoccurring observation with the corresponding prediction causes a high error due to model mismatch. Moreover, it cannot be guaranteed that the reoccurring observation is not another MPC. The only possible way to overcome this issue is for the data association algorithm to be aware of the Rx and Tx position. Then, based on the estimated Rx and Tx position, the delay can be correctly predicted and associated with the reoccurring

MPC observation.

The Rx and Tx position estimation uncertainty can be significant compared to the delay estimation. Also, multiple closely separated MPCs can be present in the radio channel. Hence, the data association algorithm needs to allow uncertainty in the data association.

The drawback of the SAGE algorithm is that it requires prior knowledge of the model order. The estimator needs to be augmented with a model order estimator when the model order is not known. Some commonly used model order estimation techniques are Akaike's Information Criterion (AIC) [Boz87], Minimum Description Length (MDL) [Ris78], and Bayesian Information Criterion (BIC) derived in [Sch78]. The problem is that the model order estimation is independent of the MPC parameter estimation and tends to overestimate or underestimate the model order. The model order overestimation is stronger when estimating a multipath radio channel with low SNR.

The logical step forward is to estimate the model order and MPC parameters jointly One of the most recent approaches to this is the Superfast Line Spectral Estimation (SLSE) algorithm [HFR18]. The SLSE algorithm uses a Bayesian approach to estimate MPC parameters jointly with the model order and variance of the noise. The combination of the multiple ideas from recent line spectral estimation publications together with highly optimal techniques, like the superfast Toeplitz inversion algorithm [AG87] and a quasi-Newton optimization algorithm "limited memory Broyden-Fletcher-Goldfarb-Shanno," makes the SLSE the state-of-the-art method for a radio channel parameter estimation with superior performance and evaluation speed compared to the similar approaches.

Despite all the advances in the multipath radio channel estimation and line spectral estimation literature, there still is not a universal algorithm for all situations. Most multipath radio channel estimation algorithms ignore the DMCs and are snapshot-based. This leads to high order models with spurious MPCs changing between consecutive measurements making multipath assisted positioning extremely challenging. Also, the computational load is prohibitively large for real-time multipath assisted positioning algorithms.

1.3 Multipath Assisted Positioning

The NLoS MPCs present an interference for a classical positioning algorithm and can significantly decrease the precision of the estimated position [CS95; SR96]. Hence, the classical positioning algorithms aim to recognize and mitigate the NLoS MPCs [WH96; Che99]. Even more challenging is mitigation of the overlapping MPCs and DMCs during positioning [SW08; JDW08].

In recent years the approach to the NLoS MPCs treatment changed fundamentally. Before the NLoS propagation presented a problem that had to be mitigated, it now becomes a feature to exploit. It shows that the correct treatment of the NLoS MPCs provides additional information that does not decrease the precision of the position estimation. Moreover, it can be utilized to improve it further.

The first attempts to utilize the multipath propagation of the radio signal used a database of the multipath measurements obtained at a given position in a given environment. The position is then obtained by matching the measured multipath radio channel with the data in the database. There is a number of methods matching the multipath radio channel information to a position. E.g., the work in [Tri+06] creates a database of Power Delay Profiles (PDPs) and corresponding position pairs. Then, the measured PDP is compared with the PDPs in a database, and the one minimizing the Least Squares (LS) error is selected. The associated position is then the position estimate. Another method published in [KWC13] is using the delay information and the AoA. The method leverages the subspace approach to prioritize strong MPCs in a rich multipath environment. This family of algorithms is called fingerprinting.

However, some multipath assisted positioning algorithms do not require a fingerprinting database, e.g., the authors of [DT09] show that a non-static UWB radar can be used to estimate the surroundings. The extension in [DT10] shows that the extracted features from the environment can be further utilized for positioning using SLAM. Further multipath assisted positioning algorithms are presented in [SW09; Fro+13; Lei+15; Lei+16; MLW14; WM12], where the authors exploited the MPCs parameters for positioning using a transmitter mirroring, together with a known room geometry and transmitter position.

However, all the previous algorithms require some prior knowledge–either a fingerprinting database or a known floor plan. A different approach is taken by a Channel-SLAM originally presented in [GJ13] and further improved in [Gen+16b; Gen+16a; Gen+17]. The Channel-SLAM does not need any prior information. Furthermore, it requires only one transmitter and a multipath environment for positioning. The Channel-SLAM treats reflections and scatterings of the electromagnetic field as static VTs. During the receiver movement the Channel-SLAM can simultaneously localize the receiver, transmitter and all VTs.

Fig. 3 visualize the principle of VTs. In this figure, the receiver moves along the red dotted line and the LoS between the physical transmitter VT_1 placed on coordinates given in $\mathbf{x}_{\mathrm{Tx},1\,t}$ shown by the solid purple line. Then if we mirror the physical transmitter along the reflecting wall-like surface, we obtain the VT_2 placed on $\mathbf{x}_{\mathrm{Tx},2\,t}$. The important property is that in every possible position of the receiver is the reflection on the reflecting surface interpretable as an LoS propagation from a static $\mathbf{x}_{\mathrm{Tx},2\,t}$ to receiver position $\mathbf{x}_{\mathrm{Rx},t}$ as shown by blue lines. Additionally, the multipath assisted positioning algorithms can exploit the scenario where the signal is scattered in every direction, as shown by the orange dash-dotted and green dashed lines. Since, from the algorithm point of view, there is no difference between the physical transmitter and VTs, they shall be all referred to as VTs throughout this work.

Most multipath assisted positioning algorithms are based on sequential Bayesian filtering techniques like KF [Kal60] and its modifications, Monte Carlo methods like Particle Filter (PF) [GSS93; Aru+02], and sum-product algorithm on factor graphs [KFL01]. All of the mentioned techniques use the Markov chain principle. The Markov chain is a property of a stochastic sequential system. The definition of the Markov chain is that the system state is dependent only on a system observation in a current step

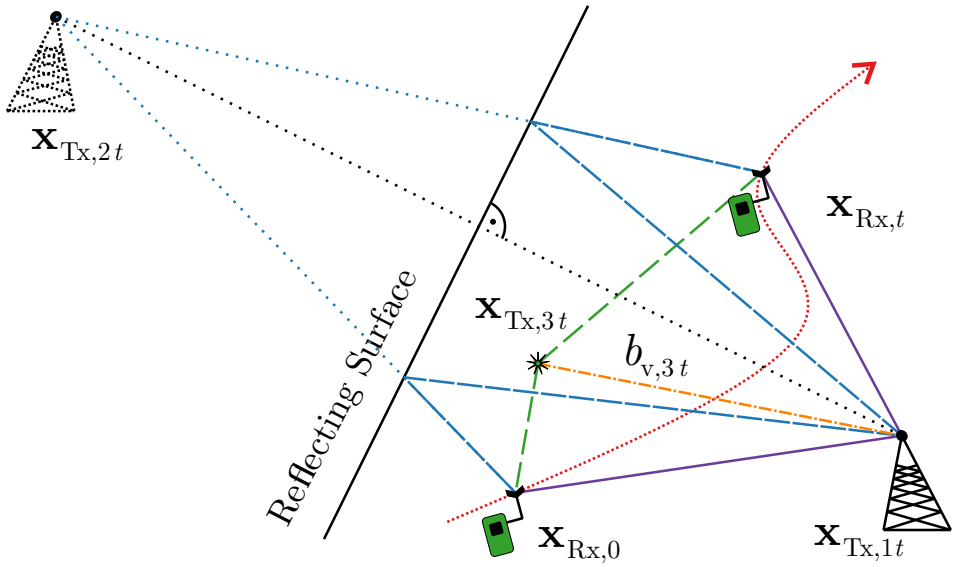


Figure 3: The concept of Virtual Transmitters. The LoS between the physical transmitter $\mathbf{x}_{\mathrm{Tx},1\,t}$ and system trajectory in two time instances $\mathbf{x}_{\mathrm{Rx},0}$ and $\mathbf{x}_{\mathrm{Rx},t}$ is shown by solid purple lines. The MPCs caused by reflection on a wall-like obstacle can be interpreted as an LoS signal propagated from a VT_2 at position $\mathbf{x}_{\mathrm{Tx},2\,t}$. The position of $\mathbf{x}_{\mathrm{Tx},2\,t}$ is obtained by mirroring the physical transmitter position along the reflecting surface. A dashed blue line shows the real ray-like propagation, while a blue dotted line shows the LoS interpretation. Additionally, the transmitted signal is scattered at a scatterer, where VT_3 defines the position of the scatterer $\mathbf{x}_{\mathrm{Tx},3\,t}$ and the corresponding MPC delay is defined as an LoS shown by green dashed lines, with an additional propagation length $b_{\mathrm{v},3\,t}$ shown by orange dash-dotted line.

and a system state in the previous step but is conditionally independent of all states and observations before. Assuming the Markov chain property allows storing only the last system observations and states. The separability of the system observation and evolution models is another valuable property leveraged in multipath assisted positioning algorithms.

The Channel-SLAM uses a Markov Chain Monte Carlo (MCMC) based PF [Aru+02] to obtain the approximation of posterior Probability Density Function (PDF) of the receiver and VTs positions. The algorithm is based on an idea of SLAM [Thr+04]; thus, the name of the algorithm is Channel-SLAM [GJ13]. Channel-SLAM is using the KEST algorithm [Jos+12] to obtain estimates of the MPC parameters and a Rao-Blackwellized Particle Filter (RBPF) [Dou+00] estimating the posterior PDF.

Another multipath assisted positioning algorithm that does not require any prior knowledge about the environment is JPDA-MINT. The JPDA-MINT, based on a Belief Propagation (BP) algorithm, calculates the receiver position's posterior PDF using factor graphs [Lei+16; Lei+17]. The JPDA-MINT system is modeled as a factor graph using the sum-product algorithm described in [KFL01]. The MPC parameters are estimated using the LS algorithm. Then, the MPCs are associated with the individual VTs using the Joint Probabilistic Data Association (JPDA). The latest version of the JPDA-MINT algorithm uses the SLSE algorithm [HFR18] for MPC parameter estimation [LGW21].

1.4 Machine Learning

Machine learning and especially Artificial Neural Networks (ANNs) are a fast-developing field of research. It was already shown that it could be used for classification tasks, where it can even outperform humans. Moreover, the current development in the ANN using deep learning allows for performing more general tasks and approximating basically an arbitrary function and using this approximation to extrapolate for previously unknown data points. Among many others, the ANN can even be used to solve differential equations [PSW19].

With this generality of application, it is not surprising that ANN found its application even in positioning. The ANNs were successfully utilized to perform a room-level positioning, where the positioning was defined as a simple classification problem [NW17]. Closer to a classical approach is the method deployed in [Wan+16; XBM17], where authors proposed positioning systems based on a fingerprinting database. However, thanks to the extrapolation done by the ANN, the proposed positioning algorithms outperform original fingerprinting techniques [BP00].

However, the machine learning techniques described above are bounded by the same environment where the training data were collected. Hence, it is impossible to use the ANN trained on a database obtained in one place for positioning anywhere else. It is desirable to address this problem and propose methods that can improve the current state-of-the-art multipath assisted positioning approaches but are independent of the current environment.

Gaussian Mixture Model

The Gaussian Mixture Model is a technique widely used in machine learning, e.g., for anomaly detection [Li+16], language detection [TRD02], reinforcement learning [AC10], and more.

In signal processing, the main motivation for this type of approximation is to make posterior PDF estimation tractable for nonlinear models with possibly non-Gaussian noise [AS72]. Especially for cases where the Extended Kalman Filter (EKF) approximation is insufficient [BB03]. When the linearization of the nonlinear model causes a large error over the update period or when the posterior PDF is multimodal.

The GMM aims to approximate an arbitrary PDF $\mathrm{p}(X)$ of a random variable X using a sum of Gaussian PDFs $\mathcal{N}(\cdot,\cdot)$ as

$$\mathrm{p}(X) \approx \sum_{k=1}^{K} w_k \mathcal{N}\left(X - X_k, \mathbf{\Sigma}_k\right), \tag{1.17}$$

where the covariance matrix $\mathbf{\Sigma}_k$ is in some cases simplified by a scaled identity matrix $\lambda\mathbf{I}$, and the weights w_k must satisfy $\sum_{k=1}^{K} w_k = 1$.

The parameters of the GMM are usually obtained by an EM algorithm [DLR77]. The work in [AS72] shows that Bayesian filtration techniques like EKF can track the GMM parameters in dynamic scenarios.

However, the calculation load of the GMM update step can be significant, especially when the model noise covariance is large or non-Gaussian. In such a case, also the model noise PDF $\mathrm{p}(v_t)$ have to be approximated by a sum of Gaussian PDFs to obtain the predictive PDF based on the nonlinear dynamic model

$$X_{t+1} = f(X_t) + v_t. \tag{1.18}$$

The GMM model order is KN without any pruning, where K is the state GMM order (1.17), and N is the noise GMM order. Several methods aim to reduce the number of mixture components to decrease the computational load [Sal90; Pao94; Li+18].

Artificial Neural Networks

Although the idea of ANN occurred already in the 1940s [MP43], the real boom started with the success of a deep learning architecture AlexNet in 2012 [KSH17]. This breakthrough established a new standard for solving classification problems. Since that, the ANN has spread to many other fields of science, e.g., into many areas of computer graphics simulations [San+20; Kar+21], reinforcement learning [Sil+16] [Vin+19], and biochemistry [Jum+21].

In general, an ANN approximates some unknown and possibly intractable nonlinear function by another nonlinear parametric function. In other words, if we have a dataset $\left\{\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)}\right\}$ of observation $\boldsymbol{x}^{(i)}$ and target $\boldsymbol{t}^{(i)}$ variable pairs, we desire to find a relation between observation and target variables as a function

$$\boldsymbol{t} = f(\boldsymbol{x}). \tag{1.19}$$

The dataset members are indexed by i. Since the function $f(\cdot)$ is generally unknown, we can try to find its parametric approximation as

$$\boldsymbol{t} \approx q(\boldsymbol{W}, \boldsymbol{x}), \tag{1.20}$$

where the nonlinear function $q(\cdot, \cdot)$ has to be selected, and the weights $\boldsymbol{W}$ parametrizing the function $q(\cdot, \cdot)$ have to be found.

Selecting the $q(\cdot, \cdot)$ architecture is based on the core idea of the ANN, which is to chain simple linear functions with well-defined nonlinear activation functions. A simple example of such an ANN is

$$q(\boldsymbol{W}, \boldsymbol{x}) = \boldsymbol{W}_1 \max(0, \boldsymbol{W}_0 \boldsymbol{x}), \tag{1.21}$$

where the $\max(\cdot)$ returns the maximum value, $\max(0, \cdot)$ is known as the Rectified Linear Unit (ReLU) activation function that replaces all negative values from the vector $\boldsymbol{W}_0 \boldsymbol{x}$ with zero. The linear function $\boldsymbol{W}_0 \boldsymbol{x}$ is referred to as a fully connected layer. The two-layer architecture in (1.21) can be proven to approximate arbitrary nonlinear function. This property is also known as the universal approximation theorem [HSW89].

Another linear function often used in ANN is a convolution. The convolutional layer implements convolution between the input data $\boldsymbol{x}$ and a set of filters parameterized by filter coefficients $\boldsymbol{W}$. Similarly to (1.21),

$$q\left(\boldsymbol{W}, \boldsymbol{x}\right) = \boldsymbol{W}_1 * \max\left(0, \boldsymbol{W}_0 * \boldsymbol{x}\right) \tag{1.22}$$

defines a simple Convolutional Neural Network (CNN) architecture, where $*$ marks a discrete convolution. The CNN is sometimes accompanied by a max-pooling layer which decreases the number of parameters in the CNN. The principle of the max-pooling layer is to cluster closely spaced input variables and select the maximum value from each cluster as an output.

Now, when a simple ANN is designed, we need to find the set of weights $\boldsymbol{W}$ to fit (1.20). The process of finding weights is also called ANN training. The locally optimal weights are usually obtained using gradient approaches, where a backpropagation method can efficiently evaluate the gradient of any feed-forward ANN architecture [Bis06].

The idea of gradient-based parameter learning is that the weights are iteratively updated in the direction given by a gradient ∇ of a loss function $L\left(\cdot, \cdot\right)$ as

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^{k} - \frac{\eta}{I} \sum_{i=0}^{I} \nabla L\left(\boldsymbol{t}^{(i)}, q\left(\boldsymbol{W}^{k}, \boldsymbol{x}^{(i)}\right)\right), \tag{1.23}$$

where the learning rate η gives a small step in the gradient direction in an iteration $k+1$. In some cases, it is not possible to evaluate the gradient of loss function for the whole dataset, e.g., the dataset is obtained sequentially, or the whole dataset might not fit the computer memory. This leads to an approach known as Stochastic Gradient Descent (SGD), where the dataset is divided into smaller batches $\left\{\boldsymbol{t}^{(j)}, \boldsymbol{x}^{(j)}\right\}$ and weights are updated sequentially over the batches as

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^{k} - \frac{\eta}{J} \sum_{j=0}^{J} \nabla L\left(\boldsymbol{t}^{(j)}, q\left(\boldsymbol{W}^{k}, \boldsymbol{x}^{(j)}\right)\right). \tag{1.24}$$

The cardinality $|\cdot|$ of the batch is referred to as a batch size.

The practice and experiments show that using SGD achieves better performance than using all available data directly (1.23). The intuition is that the higher noise of the gradient estimated from a smaller batch pushes the learning process to prefer broader, less sensitive local optima over sharp ones. Such ANN better generalizes for data not present in a training dataset. More details about the batch size influence on the ANN learning can be found in [Kes+17].

Another important aspect of the ANN training is selecting the learning rate parameter η. Selecting the learning rate too low leads to an unsatisfactory training duration. On the other hand, using the learning rate too large can cause the training not to converge or even diverge. The learning rate is one of the ANN hyperparameters usually obtained by experimenting with different values, which can be time-demanding. In

[Sut+13; DHS11; KB17; Ama98], the SGD was revisited to address this and other issues and provide alternative training schedules with better and more robust convergence even without complicated learning rate scheduling. E.g., adding momentum to speed up the training process [Sut+13], adaptive methods [DHS11; KB17], natural gradient descent [Ama98].

Practically, the most used training method is the Adam optimizer [KB17]. Compared to the SGD, the Adam optimizer also calculates the exponential moving averages of past squared gradients, which was shown to be modeling a diagonal approximation of the Fisher Information Matrix (FIM) [RKK19; Mar20]. The diagonal approximation of the FIM provides a significant calculation boost compared with the full FIM calculation in a natural gradient descent method [Ama98] while still providing a superior convergence rate compared with non-adaptive methods like SGD.

In some cases, a more complex ANN architecture consisting of chaining, forking, and merging many linear and nonlinear layers can provide a significantly better convergence rate. A famous example of such architecture is a CNN AlexNet initially presented in the ImageNet Large Scale Visual Recognition Challenge in 2012 [KSH17]. Its idea was to chain several convolutional layers with ReLU activation, max-pooling layers, and fully connected layers.

Almost arbitrary architecture, without loops, is achievable using generic ANN libraries like TensorFlow [Aba+15], Caffe [Jia+14], and PyTorch [Pas+19]. These libraries can automatically obtain the gradient of the designed ANN and use one of the standard optimizers to find the weights fitting the training dataset.

1.5 Goals of Thesis and Contributions

This thesis focuses on several open problems of modern multipath assisted positioning methods. Specifically, the goals of this thesis are:

- Research the possibility of positioning in Single Frequency Network and show the equivalence with multipath assisted positioning (Chapter 2),
- propose and test a novel pedestrian transition model for Channel-SLAM using inertial sensor fusion (Chapter 3),
- propose and verify a stochastic data association method for tracking and associating multipath components with the Virtual Transmitters in Channel-SLAM, (Chapter 4),
- research the possibility of using a machine learning-based methods for Virtual Transmitter estimation in Channel-SLAM, (Chapter 5),
- investigate the possibility of using Artificial Neural Networks for radio channel parameter estimation and compare the proposed method with the theoretical limits and benchmark algorithms. (Chapter 6).

Chapter 2

Positioning in Single Frequency Network

The chapter studies a special transmission mode where multiple transmitters synchronously broadcast the same signal at the same carrier frequency. Such a set of transmitters is said to form a SFN [Ric07]. The SFN is often utilized in Digital Video Broadcasting - Terrestrial (DVB-T) to cover a large area [ETS06]. The signal from multiple transmitters are summed at the receiver antenna, causing interference. However, since the transmitters are synchronized, a standard channel equalizer can easily suppress this interference. The synchronization causes the received signal consists of correlated replicas of the transmitted signal received with different delays.

The idea of SFN allows to cover large areas using multiple transmitters efficiently and subsequently allows to perform positioning. Like the multipath radio channel, the received signal in SFN consists of delayed replicas of the transmitted signal. This observation allows us to use a multipath assisted positioning technique for positioning in an SFN. Specifically, I model the received signal as a transmitted signal propagating through a multipath radio channel. Estimating the multipath radio channel parameters provides a delay of individual MPCs usable for positioning using the TDoA method. The findings presented in this chapter are already published in [KP15; NK16; NKV16; KV18], and I use citations from these publications.

2.1 Signal Synchronization and Delay Estimation

The DVB-T is an Orthogonal Frequency Division Multiplexing (OFDM)-modulated broadcasting service defined in [ETS15]. Usually, the 8k mode, with 8192 subcarriers, is used. Practically speaking, only 6817 of them contain non-zero values. The duration of one DVB-T symbol is 896 μs. Each OFDM subcarrier is modulated using Binary-Phase Shift Keying (BPSK) or Quadrature Amplitude Modulation (QAM) if the subcarrier contains pilot or payload data, respectively. If more than one transmitter in the site is

transmitting with the same carrier frequency, the SFN is applied according to [ETS06]. A so-called Mega-Frame Initialization Packet (MIP) inserter is used to ensure standard fulfillment.

I utilize only the pilot sequence, signals inserted for synchronization, for positioning because it is predefined and does not change, unlike the random payload data. For this reason, I limit the detailed signal description to pilot signal definition. The pilot signal consists of selected OFDM subcarriers that are BPSK modulated. The Pseudo-Random Binary Sequence (PRBS) provides a binary sequence used to modulate the pilot subcarriers. The PRBS cyclic autocorrelation function is 2047 chips long with a correlation maximum of 2047 and -1 elsewhere.

The pilot sequence consists of three parts defined by [ETS15]. The Transmission Parameter Signalling (TPS) chips are mapped to the same predefined subcarriers in each OFDM symbol and contain information related to the transmission scheme. The Continual Pilot (CP) chips are also mapped to predefined subcarriers in every OFDM symbol but carry no additional information. Finally, the Scattered Pilot (SP) chips are not mapped to the same subcarriers in every OFDM symbol. One SP chip is inserted into every 12-th subcarrier according to

$$\begin{aligned} \boldsymbol{SP} &= 3 \times (l \bmod (4)) + 12\boldsymbol{p}, \\ \boldsymbol{p} &= [0, 1, \ldots, 567], \end{aligned} \tag{2.1}$$

where vector $\boldsymbol{SP}$ consists of SP subcarrier numbers and l is the OFDM symbol number. Since $\boldsymbol{SP}$ repeats itself every four OFDM symbols, it is not necessary to know the absolute value of l to obtain the $\boldsymbol{SP}$ sequence. The current l mod (4) value suffices to create the pilot signal replica [CYY06; Che+15]. This algorithm can be described by

$$l = \arg\max_{m} \left\| \sum_{p=0}^{567} \left(R_{l,12p+3m} P_{12p+3m}\right) \left(R^*_{l-1,12p+3(m-1)} P^*_{12p+3(m-1)}\right) \right\|, \tag{2.2}$$

where $R_{l,12p+3m}$ is the value at the $12p + 3m$-th subcarrier of the l-th OFDM symbol, and $\cdot^*$ marks complex conjugation. The P_{12p+3m} contains the $(12p + 3m)$-th value of the PRBS. The obtained l is then applied to generate the SP sequence according to (2.1). This algorithm tests all four possible patterns of the SP sequence and compares values at those subcarriers in the actual and the previous OFDM symbol.

The product of the actual SP spectrum with its complex conjugated replica provides the cyclic AutoCorrelation Function (ACF). Since the SP sequence consists only of values 1 and -1, multiplying the signal spectrum turns all the pilot subcarriers to 1. Ideally, this leads to a sequence of ones that have a Dirac delta Discrete Fourier Transform (DFT) identity. The DVB-T symbol duration is 896 μs, but only every twelfth subcarrier contains the SP symbol that provides the ACF estimation. Thus the resulting impulse response duration is $74.\bar{6}\,\mu$s.

The duration of the SP sequence leads to limitations of the DVB-T-based positioning system. The signal's propagation time from one transmitter to any other transmitter in SFN has to be less than half of the autocorrelation repetition period of $74.\bar{6}\,\mu$s. Otherwise, ambiguities occur.

The first necessary step is to choose an appropriate sampling frequency. The symbol duration is $T_U = 896\,\mu$s, and $1/T_U$ defines the subcarrier frequency spacing. Notice that any sampling frequency assuring an integer number of samples per symbol duration period assures identical frequency spacing as the subcarrier frequency spacing. E.g., in DVB-T, if the sampling frequency is an integer MHz, an integer number of subcarriers is obtained. However, aliasing occurs if the sampling frequency does not cover all nonzero subcarriers. The sampling frequency of 8 MHz is used to sample 7168 DVB-T subcarriers, covering all 6817 nonzero subcarriers present in 8k mode.

The time and frequency synchronization of the DVB-T signal is performed using the ML method for OFDM symbol synchronization as described in [BSB97]. However, the OFDM modulation uses a so-called guard interval to ensure the cyclic character of the transmitted signal $s(t)$ and suppress the Inter-Symbol Interference (ISI). Since its duration can vary in DVB-T, it is necessary to estimate guard interval duration (T_Δ) to separate the sampled signal into individual OFDM symbols. The method for guard interval duration estimation I proposed in [NK16] is defined by

$$\begin{aligned} \Lambda(\tau) &= \int_0^{T_{\Delta\mathrm{MAX}}} s(t-\tau)\cdot s^*(t-\tau-T_U)\,\mathrm{d}t, \\ \Lambda'(\tau) &= \frac{\mathrm{d}\Lambda(\tau)}{\mathrm{d}\tau}, \\ T_\Delta &= \max_\tau \left(\Lambda'(\tau)\right), \end{aligned} \tag{2.3}$$

where $\Lambda(\tau)$ is a log-likelihood function. Similarly to [BSB97], the log-likelihood function aims to correlate the end of the OFDM symbol with its replica contained in the guard interval. The integration interval $T_{\Delta MAX}$ is set to the maximal possible guard interval duration. Then, the maximum of the log-likelihood function derivative estimates the T_Δ using the normalized signal. The guard interval duration in the SFN does not change often but cannot be ruled out.

When the guard interval duration is known, the ML estimation of time offset θ_{ML} and frequency offset $\varepsilon_{\mathrm{ML}}$ is obtained according to [BSB97] as

$$\begin{aligned} \theta_{\mathrm{ML}} &= \arg\max_\tau \Lambda(\tau), \\ \varepsilon_{\mathrm{ML}} &= \angle \max_\tau \Lambda(\tau), \end{aligned} \tag{2.4}$$

where the log-likelihood function $\Lambda(\tau)$ is defined by

$$\Lambda(\tau) = \int_0^{T_\Delta} s(t-\tau)\cdot s^*(t-\tau-T_U)\,\mathrm{d}t. \tag{2.5}$$

The received DVB-T signal is equalized using θ_{ML} and $\varepsilon_{\mathrm{ML}}$ obtained from (2.4). Then, the equalized DVB-T signal is separated into the individual OFDM symbols. In the following step, the CIR (1.3) is estimated using a mismatched filter. Note that in

SFN, the CIR includes MPCs generated by multiple transmitters. The filter impulse response is the ideal pilot signal used in the current OFDM symbol. However, the pilot signal's spectrum is multiplied by a window function to obtain the mismatched filter impulse response. The properties of different window functions are described in [Pra13]. I used a Taylor Window with two nearly constant-level sidelobes and a peak sidelobe level of -23 dB. The mismatched filter improves the peak detection performance since the sidelobes of the estimated CIR are suppressed. On the other hand, the main lobe becomes wider, which decreases the time estimation precision.

Assuming static measurements, or low dynamics of the receiver movement, it is possible to average multiple OFDM symbols to improve the SNR. The proposed algorithm uses averaging of the absolute value of the mismatched filter outputs. This technique is also known as a noncoherent integrator [Sko08].

The critical step for the DVB-T-based positioning algorithm is to extract the TDoA values from the estimated CIR. The TDoAs are defined as a time difference between the signal arrival from different transmitters. This is the same as the delay between the individual peaks in the averaged mismatched filter output. The problem is that not all peaks in the CIR correspond to a physical DVB-T transmitter. Hence, the algorithm needs to detect the mismatched filter output peaks caused by a DVB-T transmitter and not by noise or other influences.

The Cell Averaging - Constant False Alarm Rate (CA-CFAR) is a standard detection algorithm that provides satisfactory results in many real cases [Sko08]. The CA-CFAR algorithm automatically chooses the threshold (γ_{th}), according to the actual signal level, to maximize the detection probability (P_{D}) with a constant false alarm probability (P_{FA}) using the Neyman-Pearson theorem [Kay98] given as

$$L\left(x\right) = \frac{p\left(x;\mathcal{H}_1\right)}{p\left(x;\mathcal{H}_0\right)} > \gamma_{\mathrm{th}}, \tag{2.6}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are the null and alternative hypotheses, respectively, and γ_{th} can be obtained from

$$P_{FA} = \int_{x:L(x)>\gamma_{\mathrm{th}}} p\left(x;\mathcal{H}_0\right)\mathrm{d}x. \tag{2.7}$$

The CA-CFAR algorithm is a shift register divided into three windows with different lengths. The content of the left and right windows is added together to obtain the signal's mean level, which is proportional to γ_{th}. Multiplying the signal's mean level by the threshold constant gives the γ_{th}. The left and right window lengths are long enough to provide a sufficient estimation of γ_{th}. The middle window represents the guard interval and is omitted from the threshold level estimation. The number of cells should be long enough to cover the main lobe of the mismatched filter output. If the value in the guard window exceeds the estimated γ_{th}, detection occurs. The peak delays detected by the CA-CFAR algorithm are used for positioning. The proposed detection algorithm principle is visualized in Fig. 4.

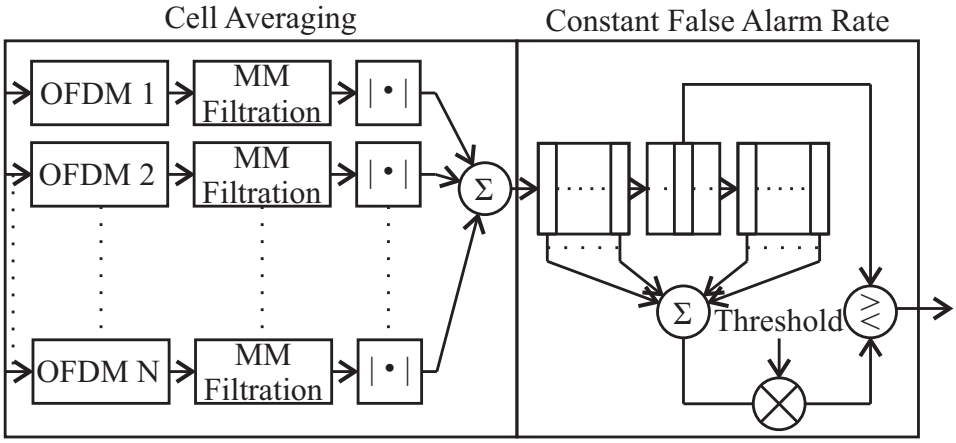


Figure 4: Principle of the CA-CFAR algorithm used for detection of the delayed DVB-T pilot signal replicas.

2.2 The Single Frequency Network Offset Estimation

The last problem to solve before positioning is the estimation of the offset in the SFN. This problem was detected during the measurements of the DVB-T network in Prague (Czech Republic). The obtained TDoA estimations at a known position would not correspond to the measurement's geometrical layout even if the $74.\bar{6}\,\mu$s cyclic character of the pilot sequence was assumed. The observed misalignment does not change with time or with measurement location. Hence, the MIP must insert a constant offset between the synchronized transmitter time and the actual transmission start into the SFN. The SFN offset causes the transmitter to appear further from the receiver than it is. This section presents a method for SFN offset estimation.

Fig. 5 shows an SFN with two receivers given by vectors $\mathbf{Rx}_n$ and two transmitters placed at $\mathbf{Tx}_o$. The signal propagation time multiplied by the speed of light (ct_{no}) gives the physical distance between the transmitter $\mathbf{Tx}_o$ and receiver $\mathbf{Rx}_n$. The virtual prolonging of the receiver to transmitter distance caused by the time offset Δ_o is $c\Delta_o$.

It is more important to show how this phenomenon influences the detections, as illustrated in Fig. 6.

$$D_{npo} = -D_{nop} = \tau_{np} - \tau_{no} \tag{2.8}$$

defines the TDoA measurements as a matrix $\mathbf{D}$ with elements D_{npo}, where the receivers are indexed by n and transmitters by o and p. Thus, e.g., τ_{np} indicates the pseudorange between receiver n and transmitter p. The position of a detected peak in the mismatched filter output gives the pseudorange τ_{np}.

If the time offsets Δ_o are known, it is possible to evaluate the signal propagation time t_{no} from pseudorange τ_{no} as $\tau_{no} = t_{no} + \Delta_o$ and obtain the ideal time differences given by matrix $\mathbf{T}$ with the elements

$$T_{npo} = -T_{nop} = t_{np} - t_{no}. \tag{2.9}$$

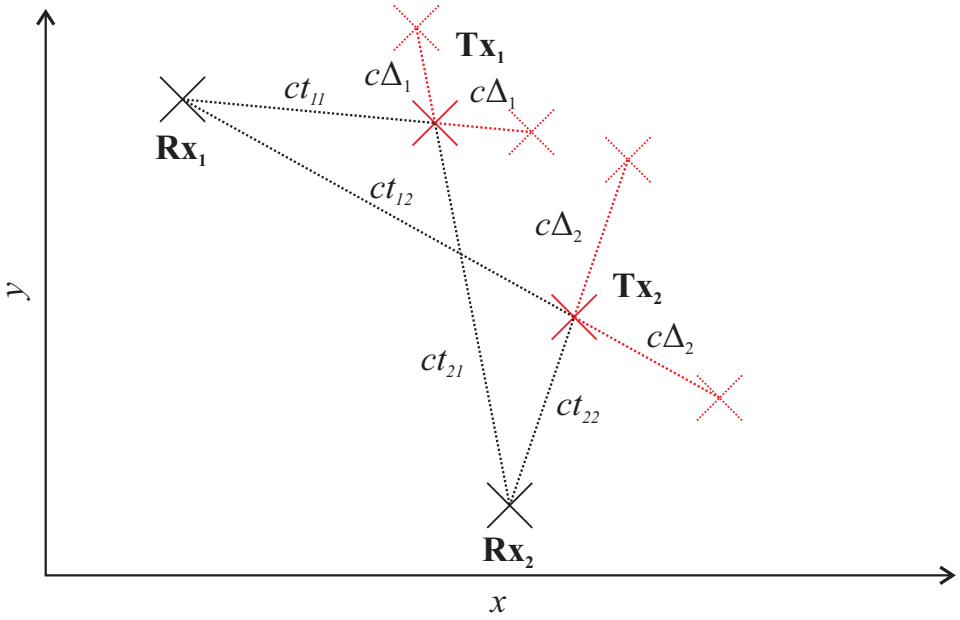


Figure 5: The SFN time offset effect illustration.

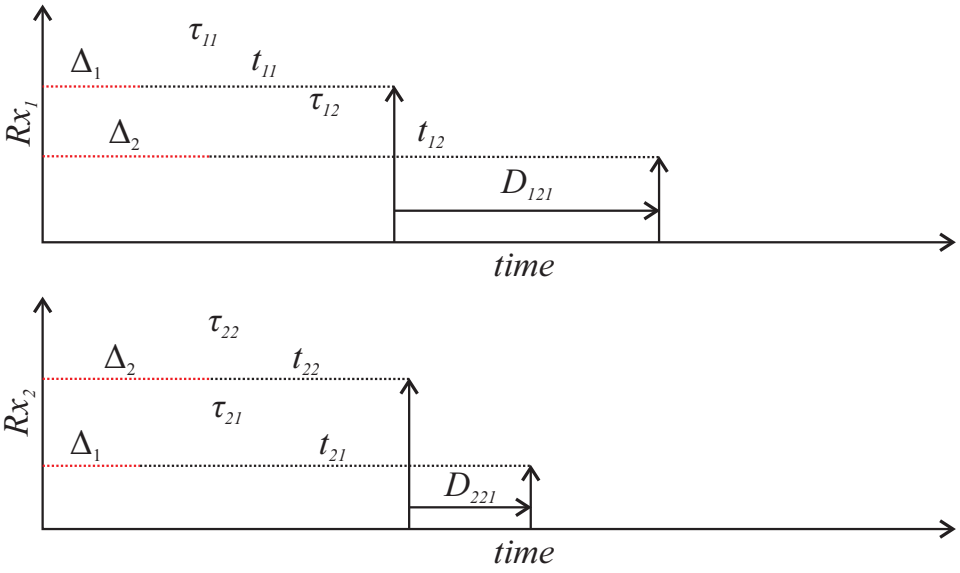


Figure 6: The SFN time offset effect illustration in filtered data.

Since the receiver time is relative, only the transmitter time offset differences are available. The time offset differences are organized in a matrix $\mathbf{O}$ with elements

$$O_{po} = -O_{op} = \Delta_p - \Delta_o. \tag{2.10}$$

This definition of the problem makes the offset estimation more transparent. For a given SFN with known transmitter positions, the matrix $\mathbf{T}$ is identifiable using measurements at known positions. The detection matrix is obtained from (2.8). Then, the offset matrix is evaluated as $\mathbf{O} = \mathbf{D} - \mathbf{T}$. However, suppose only one receiver position is used. In that case, it is not possible to select the correct order of detections and assign elements to a submatrix $\mathbf{D}_{po}$ to correspond to the $\mathbf{T}_{po}$. Adding another dimension to matrix $\mathbf{O}$, including all permutations of detection to transmitter assignments, solves the ambiguity. The elements of $\mathbf{O}$ are now given by

$$O_{nkpo} = D_{nkpo} - T_{npo}, \tag{2.11}$$

where k is a transmitter permutation index, and $\mathbf{D}$ includes all permutations of detected TDoA values similarly to $\mathbf{O}$.

If multiple measurements at different locations are performed, only one permutation k results in the same offset values in all submatrices $\mathbf{O}_n$. This permutation gives us relative offset values in the measured SFN and matches them to the individual transmitters. The relative offsets slightly differ in the individual measurements because of many influences, e.g., noise and multipath propagation. The resulting relative offset values from the individual measurements are averaged to reduce measurement inaccuracies. The offset estimates are used to correct the position estimation in the SFN.

2.3 Position Estimation

As noted earlier, the detection algorithm provides the TDoA measurements. For the TDoA positioning scenario, the Newton-Raphson iterative algorithm, similar to the ToA method described in [ME01], is used. I published the derivation of the Newton-Raphson iterative algorithm for the TDoA positioning in [NK16].

The automatic detection algorithm described in Section 2.1 provides a local time of the individual detections. Only the time differences between the detections are necessary for the TDoA positioning method. Hence, the time offset between the local and global timing in the SFN is irrelevant.

The first step of the TDoA algorithm is a conversion of the arrival time differences to the distance differences by multiplying by the speed of light. The distance differences are organized into the matrix with coefficients given by

$$d_{ij} = \|\mathbf{x}_{\mathrm{Rx}} - \mathbf{x}_{\mathrm{Tx},i}\| - \|\mathbf{x}_{\mathrm{Rx}} - \mathbf{x}_{\mathrm{Tx},j}\| + \epsilon_{12}, \tag{2.12}$$

where $\mathbf{x}_{\mathrm{Rx}}$ is a vector containing receiver coordinates, $\mathbf{x}_{\mathrm{Tx},i}$ is a vector containing coordinates of the i-th transmitter, and the additive error ϵ_{12} is considered in the measured time differences.

For the iterative solver, the variables are separated into a predictive ($\mathbf{x}_{\mathrm{Rx,p}}$) and corrective ($\mathbf{x}_{\mathrm{Rx,c}}$) part. Applying the Taylor approximation of the 1st order to distances between the receiver and transmitters results in

$$\|\mathbf{x}_{\mathrm{Rx,p}} + \mathbf{x}_{\mathrm{Rx,c}} - \mathbf{x}_{\mathrm{Tx},i}\| \approx \|\mathbf{x}_{\mathrm{Rx,p}} - \mathbf{x}_{\mathrm{Tx},i}\| + \frac{(\mathbf{x}_{\mathrm{Rx,p}} - \mathbf{x}_{\mathrm{Tx},i})}{\|\mathbf{x}_{\mathrm{Rx,p}} - \mathbf{x}_{\mathrm{Tx},i}\|}\mathbf{x}_{\mathrm{Rx,c}}, \tag{2.13}$$

where the second part of the equation contains the unitary direction vector defined as

$$\mathbf{1}_i \equiv \frac{(\mathbf{x}_{\mathrm{Rx,p}} - \mathbf{x}_{\mathrm{Tx},i})}{\|\mathbf{x}_{\mathrm{Rx,p}} - \mathbf{x}_{\mathrm{Tx},i}\|}. \tag{2.14}$$

Now, using the direction vector (2.14) and the corrective part of the difference matrix $\mathbf{d}$ given by (2.12), it is possible to write the corrective part as

$$\mathbf{d}_{\mathrm{c}} = \begin{bmatrix} \mathbf{1}_i - \mathbf{1}_j \\ \vdots \\ \mathbf{1}_i - \mathbf{1}_j \end{bmatrix} \mathbf{x}_{\mathrm{Rx,c}} = \mathbf{G}\mathbf{x}_{\mathrm{Rx,c}}. \tag{2.15}$$

The geometry matrix $\mathbf{G}$ is generally not a square matrix. For that reason, corrective part estimated from

$$\mathbf{x}_{\mathrm{Rx,c}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}\mathbf{d}_{\mathrm{c}}, \tag{2.16}$$

requires a pseudo-inverse of $\mathbf{G}^T\mathbf{G}$. In the beginning, an arbitrary initial point in the vicinity of the network is used as a position prediction $\mathbf{x}_{\mathrm{Rx,p}}$. Then, the correction $\mathbf{x}_{\mathrm{Rx,c}}$ is computed using (2.16), and the predicted position is corrected. The algorithm is repeated until the Euclidean norm of the correction vector is lower than a given threshold. This algorithm quickly converges to the correct solution after several iterations. Typically, the Euclidean norm of the correction vector is in order of 0.1 m after seven iterations and after eight iterations in order of 1×10^{-7} m. This method estimates the receiver position iteratively while minimizing the least square error.

2.4 Dilution of Precision

The Dilution of Precision (DoP) serves to quantify the quality of the geometry of the transmitter-receiver constellation. The DoP coefficients are scalar measures that relate the positioning parameter estimator variance and the variance of the estimated position. Our study of the DoP for multiple SFNs located in Prague is published in [NKV16].

The DoP for the TDoA positioning can be obtained similarly to [ME01] as

$$\mathrm{cov}(\mathbf{d}) = \mathbf{I}\sigma_d^2 \tag{2.17}$$

$$\mathrm{cov}(\mathbf{x}_{\mathrm{Rx}}) = (\mathbf{G}^T\mathbf{G})^{-1}\sigma_d^2 = \mathbf{H}\sigma_d^2, \tag{2.18}$$

where σ_d^2 is a variance of the TDoA estimator and $\mathbf{G}$ is the geometry matrix defined in (2.15). The values on the main diagonal of the matrix $\mathbf{H}$ are the squared DoP

coefficients for individual axis directions. Using the East, North, Up (ENU) coordinate system allows for defining the East-West Dilution of Precision (EDoP) as

$$EDOP = \sqrt{H_{11}}, \tag{2.19}$$

where the H_{11} denotes the first term of the matrix diagonal. Similarly, the H_{22} and H_{33} define the North-South Dilution of Precision (NDoP) and Vertical Dilution of Precision (VDoP), respectively. In addition to these, Horizontal Dilution of Precision (HDoP) and General Dilution of Precision (GDoP) are defined as

$$HDOP = \sqrt{H_{11} + H_{22}} \tag{2.20}$$

$$GDOP = \sqrt{H_{11} + H_{22} + H_{33}}\,. \tag{2.21}$$

The expected variance of the 3D position is obtained from

$$\sigma^2_{\text{pos}} = GDOP^2 \sigma^2_d. \tag{2.22}$$

A similar equation is valid for all other DoP coefficients, e.g., HDoP for variance in the horizontal plane, etc. Practically, a lower value of DoP brings a more accurate solution.

To visualize the system coverage, I show the analysis of the HDoP using two SFNs available in Prague and its surroundings for for three positioning scenarios.

1. *2D solution with three transmitters on DVB-T channel 42.* In Fig. 7 can be seen that the HDoP performance is rather poor in the vicinity of the transmitter constellation. All three transmitters are distributed almost in one line causing poor geometry. Therefore the intersecting hyperbolas - lines of constant TDoA - are nearly perpendicular in the east-west direction. Thus the NDoP is relatively low, but EDoP is extremely high.

2. *2D solution with 4 transmitters on DVB-T channels 42 and 46.* Suppose the TDoA from DVB-T channel 46 is included (see the new transmitter in the left part of Fig. 8.) In that case, the overall HDoP performance significantly improves, especially in the area between the DVB-T channel 42 transmitters.

3. *3D solution with 4 transmitters on DVB-T channels 42 and 46,* see Fig. 9. With the introduction of the new transmitter also the 3D positioning is available. The area of excellent HDoP value is not as large as in the second case.

The 3D positioning case shows the HDoP to compare it with the 2D positioning solutions. Moreover, the GDoP value is generally high in the whole area for the 3D positioning case because the transmitter heights are almost the same, especially when compared to the horizontal distances. The above analyses were performed in the MATLAB software and plotted in GoogleEarth using [Dav12].

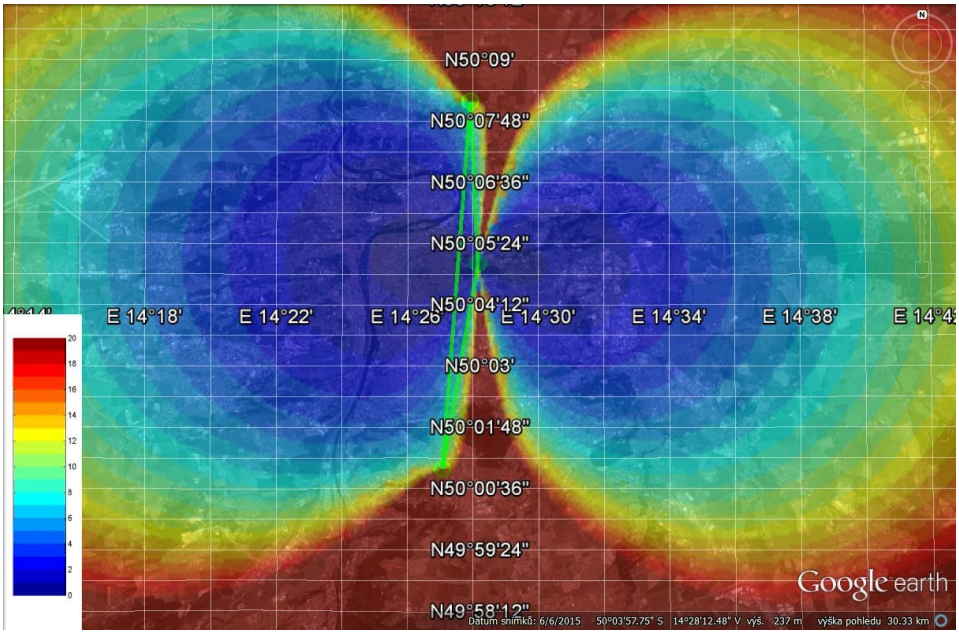


Figure 7: HDoP for 2D solution with 3 transmitters.

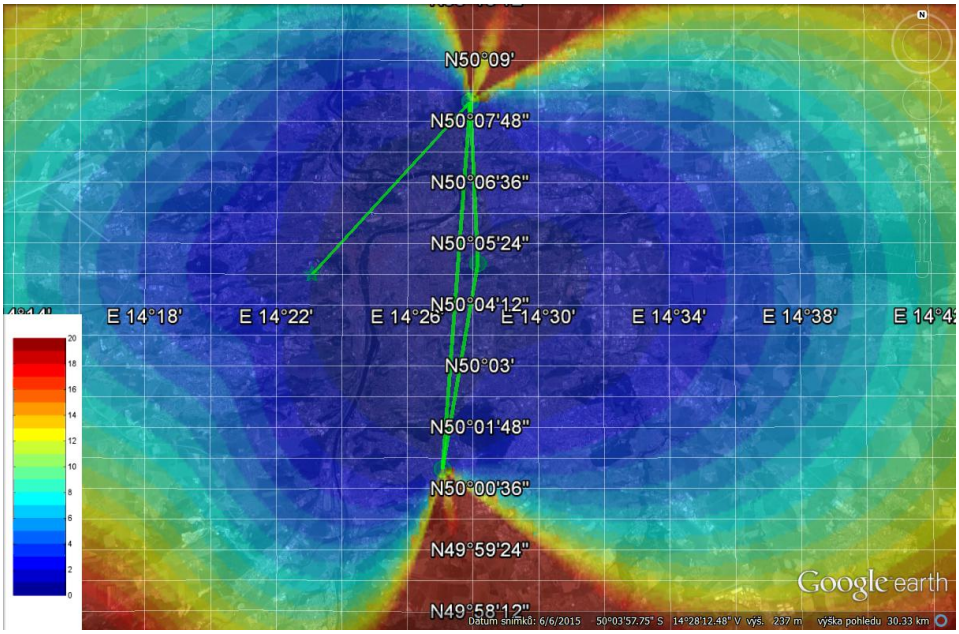


Figure 8: HDoP for 2D solution with 4 transmitters.

2.5 Experimental Results

The method for utilizing a DVB-T network for positioning was verified by the experimental measurement of the real signal in Prague (Czech Republic). Channel 42, with a center frequency of 642 MHz and consisting of three transmitters working in the SFN, is an ideal choice to demonstrate the designed positioning system. Transmitters in this

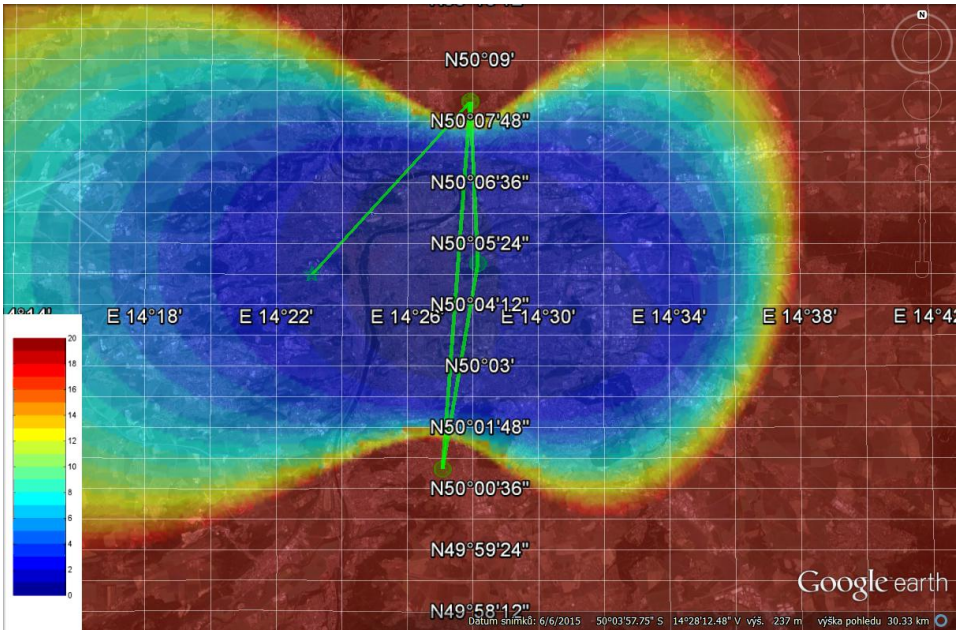


Figure 9: HDoP for 3D solution with 4 transmitters.

network are deployed nearly in one line across the city. At the north, the Ládví ($\mathbf{Tx}_1$) transmitter transmits with 20 kW Effective Radiated Power (ERP). Further south from $\mathbf{Tx}_1$ is the Olšanská transmitter ($\mathbf{Tx}_2$) with 10 kW ERP, and the Novodvorská transmitter ($\mathbf{Tx}_3$) covers the greater southern area with 5 kW ERP. The Global Positioning System (GPS) coordinates of the individual transmitters are shown in Table 2.1.

The measured points in this experiment are designed to minimize the influence of signal propagation, such as diffraction and multipath, to lower transmitter offset estimation error. The positions with a free line of sight from the receiver to most transmitters are selected for this experiment. Their coordinates are shown in Table 2.1. Fig. 11 illustrates the overview of the experimental scenario. Red points mark the position of the transmitters, and the measurement spots are blue.

The Software-Define Radio (SDR) ETTUS USRP N210, featuring a dual 14-bit Analog-to-Digital Converter (ADC), with a sampling frequency of 100 MS/s and a daughterboard UBX 10-6000 MHz Rx/Tx, is used as the receiver. It is possible to connect it directly with MATLAB to download the sampled data. A marine omnidirectional Glomex TV antenna serves as a receiver antenna. The block diagram of the experimental receiver is shown in Fig. 10.

Two measurements were performed at every position and time with a sampling frequency of 10 MHz, with the duration of the record corresponding to 10 OFDM symbols. The experiment aims to estimate the offset in the SFN and use it to evaluate the correct position of the receiver.

The measured data are processed with the presented algorithm that first finds the duration of the guard interval and then the beginning of the OFDM symbol. Then, the actual SP sequence is found and is used to choose the corresponding mismatched

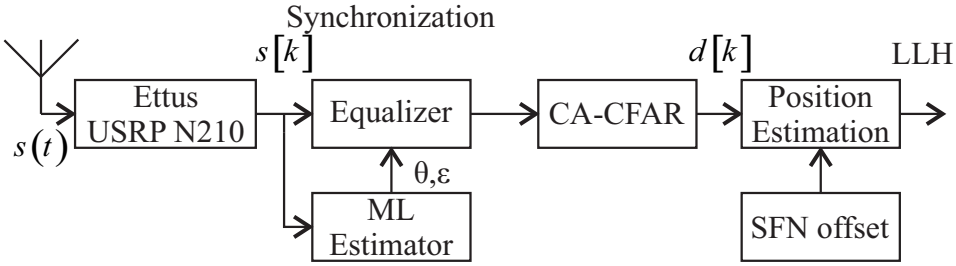


Figure 10: The block diagram of the experimental receiver.

Table 2.1: The coordinates of the DVB-T transmitters and the measured positions.

Transmitter	Latitude °	Longitude °	Altitude [m a.s.l.]
$\mathbf{Tx}_1$ (Ládví)	50.13642	14.46528	359
$\mathbf{Tx}_2$ (Olšanská)	50.08352	14.46910	250
$\mathbf{Tx}_3$ (Novodvorská)	50.01638	14.45111	305
Receiver	Latitude °	Longitude °	Altitude [m a.s.l.]
$\mathbf{Rx}_1$ (Parukářka)	50.08510	14.46022	300
$\mathbf{Rx}_2$ (Strahov)	50.08048	14.39566	330
$\mathbf{Rx}_3$ (Suchdol)	50.13175	14.38581	275

filter. Next, the signal is separated into individual OFDM symbols, and the guard interval is removed. Symbols are filtered using a corresponding mismatched filter and added together using the noncoherent integrator. The filtration is performed only on the interval corresponding to the SP sequence duration. At this point, the pre-processed data sequence is interpolated and processed with the CA-CFAR algorithm to obtain detections.

The individual detections are assigned to the corresponding transmitters using additional information from the GPS receiver. Since the DVB-T positioning system is intended as a supplementary system in hard conditions, it is possible to assume that the initial position is roughly known. When an outage occurs, the last known position is used to associate the detections with the DVB-T transmitters.

A visualization of a CA-CFAR algorithm for one experimental measurement on every position is shown in Fig. 12, Fig. 13, and Fig. 14. The red line shows filtered data, and the threshold estimated by the CA-CFAR is blue. It is visible that many peaks caused by multipath and noise are below the estimated threshold and are omitted, keeping the false alert rate low.

Next, the algorithm described in Section 2.2 estimates the SFN offset. All permutations of the TDoA are evaluated and stored in matrix $\mathbf{O}$. Table 2.2 shows three submatrices of $\mathbf{O}_n$, containing all offset permutations for the n-th experiment. Each $\mathbf{O}_n$, corresponding to the n-th experiment, includes a single permutation k with similar

Figure 11: The overview of the experiment scenario.

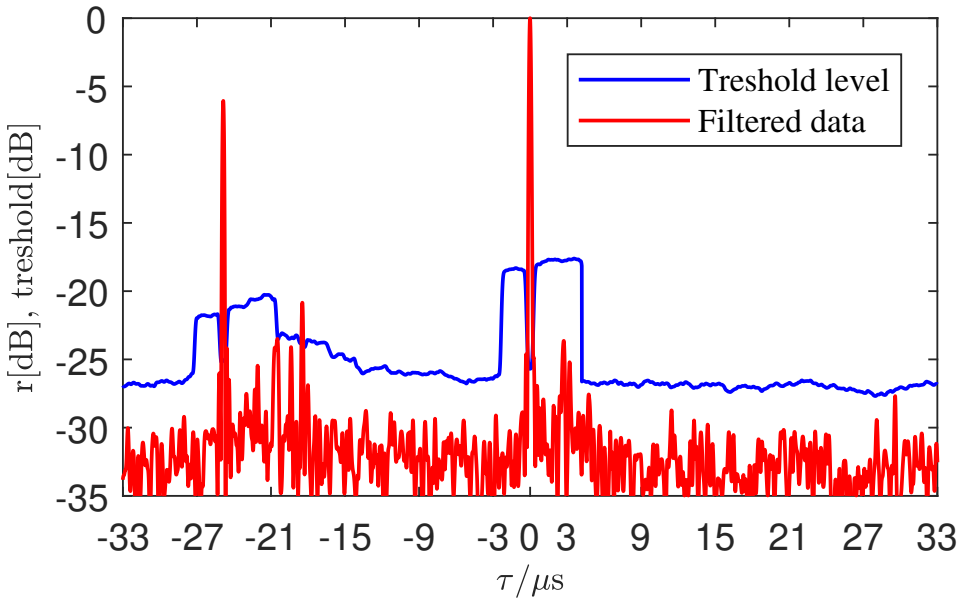


Figure 12: The output of the CA-CFAR algorithm for the experiment at $\mathbf{Rx}_1$.

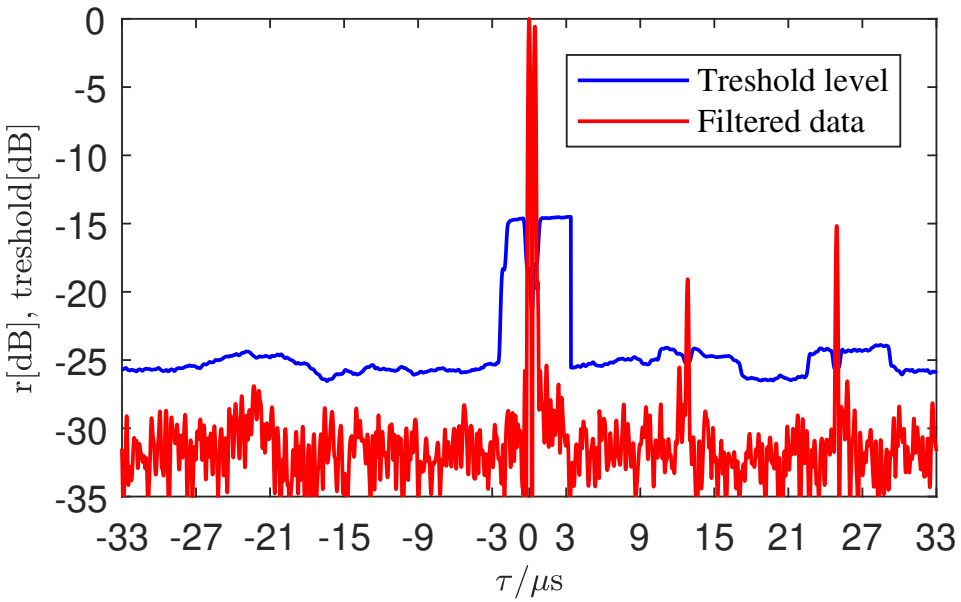


Figure 13: The output of the CA-CFAR algorithm for the experiment at $\mathbf{Rx}_2$.

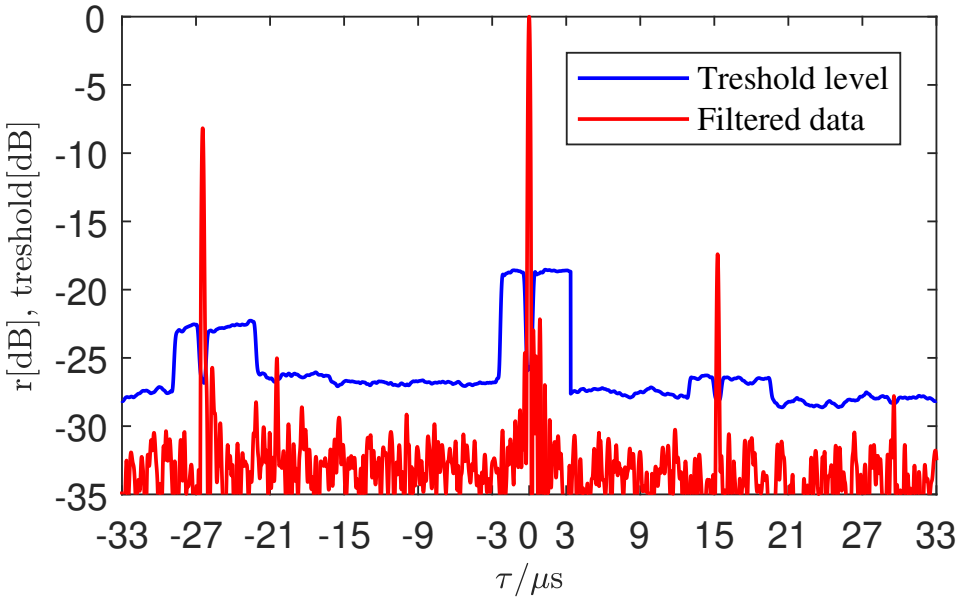


Figure 14: The output of the CA-CFAR algorithm for the experiment at $\mathbf{Rx}_3$.

Table 2.2: The evaluated offset values in submatrices $\mathbf{O}_n$.

k	$\mathbf{O}_1$ ($\mathbf{Rx}_1$) [μs]		
1	0	12.9066	48.2440
2	0	00.1066	41.8440
3	0	29.7840	31.3666
4	0	18.3534	23.2700
5	0	16.9840	24.9666
6	0	10.4774	11.9534

k	$\mathbf{O}_2$ ($\mathbf{Rx}_2$) [μs]		
1	0	01.0540	42.3130
2	0	00.1140	41.8430
3	0	10.0864	33.2806
4	0	09.5024	32.1127
5	0	09.1464	32.8106
6	0	08.5624	31.6427

k	$\mathbf{O}_3$ ($\mathbf{Rx}_3$) [μs]		
1	0	52.9687	68.3195
2	0	00.0487	41.8595
3	0	45.2728	76.0153
4	0	18.7641	22.9980
5	0	07.6472	57.2025
6	0	30.6939	34.1559

Table 2.3: The coordinates obtained from measurements.

Receiver	Latitude °	Longitude °	HDoP	Err [m]
$\mathbf{Rx}_1$ I	50.08507	14.46034	0.8	9
$\mathbf{Rx}_1$ II	50.08510	14.46018	0.8	3
$\mathbf{Rx}_2$ I	50.08034	14.39604	1.7	30
$\mathbf{Rx}_2$ II	50.08031	14.39587	1.7	22
$\mathbf{Rx}_3$ I	50.13175	14.38594	3.8	9
$\mathbf{Rx}_3$ II	50.13123	14.38721	3.7	115

values for all six experiments. In Table 2.2, this corresponds to the second line. No other line consists of similar values. Hence, it is the most likely estimate of the SFN offsets. The offsets obtained from the experiments are averaged and used as an offset correction. The estimated offsets are 0.0897 μs and 41.8472 μs for the $\mathbf{Tx}_1$ and $\mathbf{Tx}_2$, respectively. Those values are relative to the $\mathbf{Tx}_3$ since this method cannot obtain the absolute transmission delay of the individual transmitters.

With the known SFN offsets, it is possible to start positioning from the data in this SFN. The output of the positioning algorithm using the SFN offset correction is shown in Table 2.3. The fourth column shows the HDoP estimation. The geometry matrix $\mathbf{G}$ defined in (2.15) provides the HDoP estimates as

$$HDOP = \sqrt{\mathrm{tr}\left((\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}\right)}. \tag{2.23}$$

This experiment aimed to confirm the assumption that constant time offsets are applied to DVB-T transmitters. It is possible to estimate the relative values of these offsets and use them to evaluate the correct position. Even in this small sample, it is possible to determine that the HDoP significantly influences the positioning precision as it changes rapidly within the area covered by the SFN.

2.6 Outlook of the Method

The experimental results presented in this chapter show that the proposed SFN positioning algorithm can be used for position estimation in a real-life scenario where the individual transmitters in the DVB-T network are synchronized but delayed by an unknown offset. The similarity of the SFN principle with the modern multipath assisted positioning approach can provide several possible improvements. First of all, the detection algorithm based on mismatched filtering is a statistically suboptimal method. When the MPCs caused by the transmitters in the SFN are well separated, the difference between the proposed method and the optimal channel estimation approaches is negligible. However, closely separated MPCs require using an ML or an MAP method for optimal delay estimation. The SAGE [Fle+99] algorithm can provide good ML estimates of the MPC parameters. However, this method is also suboptimal if the number of MPCs is unknown. To jointly estimate the number of MPCs and their parameters requires using the MAP estimation algorithm such as the recently proposed SLSE [HFR18]. Using SLSE provides statistically optimal estimates of MPC delays necessary for obtaining the TDoA values required for positioning. Moreover, the synchronization part can be omitted because the knowledge of the signal structure is the only necessary information for MPC parameter estimation. The original method is still needed to initialize the guard interval duration and the current SP sequence.

Another improvement of the proposed method is using a sequential positioning algorithm. The proposed method is a one-shot approach that estimates position but does not utilize the information of the previous position estimates. A PF-based approach can provide a sequential positioning algorithm outperforming the proposed one-shot method. The PF-based approach can provide multiple advantages compared with the initial study performed in this section. The simple data association algorithm used in this chapter can cause ambiguities. Based on my following research [KG20], the Multiple Hypothesis Tracking (MHT) provides an optimal data association technique, which could also be utilized for the SFN-based positioning. The PF can jointly estimate the SFN offset and position similarly to [KG20]. Hence, using the proposed SFN offset estimation algorithm would no longer be necessary. Another advantage is that even if the offset values are changed in the SFN, no measurements in a specified location are required prior to positioning.

Chapter 3

Pedestrian Transition Model for Channel-SLAM

This chapter introduces the general principle of the system evolution model, also known as a movement model or transition model. Next, it presents a more general approach where some of the system observation variables, also called measurements, can be directly integrated into the transition model to improve its properties. This direct combination of measurements and transition model is also known as a tight fusion of sensors. I propose a tightly fused transition model to predict pedestrian movement using a handheld Inertial Measurement Unit (IMU).

The proposed pedestrian transition model fuses the general random walk transition model with measurements obtained from a handheld IMU using a Bayesian framework. It is not possible to double integrate the body movement acceleration to calculate position considering a handheld device. While walking, the pedestrian accelerates the sensor in many directions, which influences the measurement more than the actual pedestrian's movement. This problem is addressed, e.g., in [Lee+13] and [KE17], where step detection and step duration are used for positioning. This approach requires the creation of a step length estimator. When using a handheld IMU, the step length is not directly observable. However, it is observable when the step length changes. Moreover, combining the proposed pedestrian transition model with the Channel-SLAM allows for better tracking of the changes in the step length. I published this work in [KG20], and I use citations from this article throughout this chapter.

3.1 Inertial Sensor Measurements as a Control Signal

In the Bayesian framework, the Joint Probability Density Function (JPDF) of the system state and observation is factorized into conditionally independent factors. The transition model is usually one of the factors and is defined by a conditional PDF

$$\mathrm{p}(\mathbf{x}_{\mathrm{u},t} \mid \mathbf{x}_{\mathrm{u},t-1}), \tag{3.1}$$

where $\mathbf{x}_{\mathrm{u},t}$ defines the system state at time step t. Using a Markov chain property, $\mathbf{x}_{\mathrm{u},t}$ is conditionally dependent only on the most recent system state $\mathbf{x}_{\mathrm{u},t-1}$.

However, the proposed pedestrian transition model uses the idea of control signals usually used to describe the transition models of controlled robotic systems. The controlled transition model can be defined as

$$\mathrm{p}(\mathbf{x}_{\mathrm{u},t} \mid \mathbf{x}_{\mathrm{u},t-1}, \mathbf{u}_t), \tag{3.2}$$

where $\mathbf{u}_t$ represents the IMU-based control signals for the transition model.

The presented approach utilizes raw IMU measurements of angular velocities and accelerations, performs sensor calibration, extracts yaw rate $\dot{\Psi}$, and estimates a Standard Deviation (STD) of acceleration β. The time derivation of β can be utilized to detect changes in walking style, especially if the pedestrian is standing still, increasing or decreasing the walking pace. Finally, the pedestrian step frequency ω based on a peak/valley step detector [KE17] is used with a step length γ to calculate the distance walked.

The novel idea presented in this work is to approximate the position state vector $\mathbf{x}_{\mathrm{u},t}$ and the control signals $\mathbf{u}_t$ by a set of particles sampled from its prior PDF. This idea allows us to incorporate the IMU measurements directly into the movement model as stochastic control signals and better approximate the time evolution PDF. This idea presents a general approach allowing us to use a highly nonlinear movement model without linearization and approximation by Gaussian distributions. Incorporating $\mathbf{u}_t$ does not change the algorithm's computational complexity scaling. Hence, the algorithm's computational complexity remains almost the same. At the same time, a better estimate of the $\mathbf{x}_{\mathrm{u},t}$ conditional PDF greatly improves the performance of the whole positioning system.

3.2 Complete Pedestrian Transition Model for Channel-SLAM

The state vector of the pedestrian transition model with handheld IMU is defined as

$$\mathbf{x}_{\mathrm{u},t} = \left[x_t, y_t, v_t, \Psi_t, b_{\dot{\Psi},t}, \gamma_t, b_{\mathrm{u},t}\right]^{\mathrm{T}}, \tag{3.3}$$

where x_t and y_t are coordinates of a pedestrian in a 2D Cartesian coordinate system, v_t is the magnitude of a pedestrian velocity vector in m/s, Ψ_t is the yaw of a system in radians, $b_{\dot{\Psi},t}$ is a yaw rate bias in rad/s, γ_t is a step length in meters, and the receiver clock bias after multiplying by the speed of light is in $b_{\mathrm{u},t}$. Note that the receiver clock bias is not an observable variable when using only the IMU. The receiver clock bias is important when combining the proposed pedestrian transition model with Channel-SLAM.

The control signal vector is defined as

$$\mathbf{u}_t = \left[\dot{\Psi}_t, \omega_t, \dot{\beta}_t, \xi_t, \varepsilon_t\right]^{\mathrm{T}}, \tag{3.4}$$

where $\dot{\Psi}_t$ is the yaw rate in rad/s, ω_t is the stepping frequency in Hz, $\dot{\beta}_t$ is a time derivation of β, ξ_t is an *ismoving* flag providing a binary estimation of whether the pedestrian is moving (True) or standing still (False). The *ismoving* flag is estimated according to the STD of accelerometer measurements. The last of the control signals, ε_t, is a flag indicating the validity of the control signals.

The transition model (3.2) is defined as a nonlinear vector function that separates into a set of equations corresponding to one variable from a state vector (3.3). The set of state equations is defined as follows

$$x_t = x_{t-1} + \delta_t v_{t-1} \cos\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\frac{\delta_t^3}{3}}\mathcal{N}\left(0, \sigma_{w_x}^2\right), \tag{3.5}$$

$$y_t = y_{t-1} + \delta_t v_{t-1} \sin\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\frac{\delta_t^3}{3}}\mathcal{N}\left(0, \sigma_{w_y}^2\right), \tag{3.6}$$

$$\begin{aligned} v_t = (1-\varepsilon_t)&\sqrt{\left[v_{t-1}\cos\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_{w_x}^2\right)\right]^2} \\ &\overline{+\left[v_{t-1}\sin\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_{w_y}^2\right)\right]^2} \\ +\varepsilon_t&\sqrt{\left[\xi_t\omega_t\gamma_{t-1}\cos\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_{w_x}^2\right)\right]^2} \\ &\overline{+\left[\xi_t\omega_t\gamma_{t-1}\sin\left(\Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right)\right) + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_{w_y}^2\right)\right]^2}, \end{aligned} \tag{3.7}$$

$$\Psi_t = \Psi_{t-1} + \delta_t\left(\dot{\Psi}_t + b_{\dot{\Psi},t-1}\right), \tag{3.8}$$

$$b_{\dot{\Psi},t} = b_{\dot{\Psi},t-1} + \mathcal{N}\left(0, \sigma_{\mathrm{BS}}^2\right), \tag{3.9}$$

$$\gamma_t = \gamma_{t-1} + c_\beta \delta_t \dot{\beta}_t + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_\gamma^2\right), \tag{3.10}$$

$$b_{\mathrm{u},t} = b_{\mathrm{u},t-1} + \sqrt{\delta_t}\mathcal{N}\left(0, \sigma_{\mathrm{Rx}}^2\right). \tag{3.11}$$

The model used for the position (3.5), (3.6), and velocity (3.7) is, essentially, the White Noise Acceleration (WNA) model [BLK01] in 2D, with the only difference being that velocity is modeled as a magnitude of velocity and angle in the direction of movement. In the beginning, the white noise samples for every particle are obtained from the Gaussian distributions with variances $\sigma^2_{w_x}$ and $\sigma^2_{w_y}$ for the x and y axes, respectively. Then, the generated values serve as a point-wise approximation of a Gaussian distribution. The variances are obtained from the expected dynamics of a system and sampling period using the relation

$$\Delta v \approx \sqrt{\delta_t \sigma^2_{w_x}}, \tag{3.12}$$

where Δv is the maximal expected velocity difference between two measurements, and δ_t is the duration of a time step. The particle-based approximation performs a stochastic process simulation. Hence, the particle weights automatically provide the expected covariance of a WNA model.

Next, the velocity (3.7) is calculated by two possible relations separated by the validity flag ε_t. Suppose the data provided by a step detector are not valid, then the term $(1 - \varepsilon_t) = 1$ and the previous velocity, and the random acceleration give the future velocity. In the second case, when the pedometer data are valid, i.e., $\varepsilon_t = 1$, the step length γ_{t-1}, pedestrian step frequency ω_t, the *ismoving* flag ξ_t, and acceleration noise give the future velocity. Note that the measured values provided in the control signals are also represented as samples from their prior PDFs, similar to the acceleration noise in the movement model. This doubling of the velocity update equation allows us to predict the PDF of velocity even if the data provided by the IMU are not valid.

The prediction of yaw is given by (3.8). It is based on a Micro-Electro-Mechanical System (MEMS) gyroscope measurement model using Angle Random Walk (ARW) and Bias Stability (BS) [Woo07] to sample from its prior PDF. The ARW represents the STD of angle noise after the integration, usually in $^\circ/\sqrt{\text{hr}}$. The manufacturer of the sensor provides this value. Then the prior PDF of the angular velocity measurement $\dot{\Psi}_t$ in (3.8) is given by

$$\dot{\Psi}_t = \check{\dot{\Psi}}_t + \frac{\pi}{180\sqrt{3600}} \frac{1}{\sqrt{\delta_t}} \mathcal{N}\left(0, \sigma^2_{\text{ARW}}\right), \tag{3.13}$$

where the $\check{\dot{\Psi}}_t$ is the measured yaw rate value in rad/s obtained from a gyroscope after the sensor calibration. The check symbol "ˇ" is used to mark values provided by the IMU after calibration. The second parameter of the gyroscope measurement model, the BS, is included in a yaw rate bias estimator in (3.9). Suppose the calibration took place at the beginning of a measurement. In that case, the assumption that the calibrated bias of a yaw rate is a zero-mean random number with Gaussian distribution and the STD σ_{BS} in rad/s holds.

The estimation of step length is provided in (3.10), where the prediction of the step length in the following time step is modeled as a random variable given by the old value γ_{t-1} and Gaussian zero-mean distributed noise with the STD $\sqrt{\delta_t}\sigma_\gamma$. The last part of the step update equation is linked with the IMU measurement and walking pace estimation

[KE17], where $\dot{\beta}_t$ is obtained as

$$\dot{\beta}_t = \check{\dot{\beta}}_t + \delta_t \mathcal{N}\left(0, \sigma^2_{\ddot{\beta}}\right), \tag{3.14}$$

where $\sigma_{\ddot{\beta}}$ is chosen high enough to cover the errors of $\check{\dot{\beta}}_t$, which is the estimated derivative of $\check{\beta}_t$ provided by the step counting algorithm as

$$\check{\dot{\beta}}_t = \frac{\check{\beta}_t - \check{\beta}_{t-1}}{\delta_t}, \tag{3.15}$$

where function $\check{\beta}_t$ is an estimation of the acceleration STD obtained by filtering the measured acceleration by a low pass Butterworth filter.

(3.10) models the change of step length as a value proportional to the change of acceleration STD with proportionality constant c_β. This simplification of an unknown dependence between the accelerometer measurement and the actual change of the step length is possible when assuming a random distribution of the error of such mapping, as shown in (3.14).

Finally, the time bias of a receiver is predicted by (3.11). The receiver bias is modeled as a Wiener process with an STD $\sqrt{\delta_t}\sigma_{\mathrm{Rx}}$.

The velocity update (3.7) used information extracted from the IMU measurement, but its PDFs had not yet been provided. The measured pedestrian step frequency ω_t is a Gaussian distributed random number defined as

$$\omega_t = \check{\omega}_t + \mathcal{N}\left(0, \sigma^2_\omega\right), \tag{3.16}$$

where $\check{\omega}_t$ is a pedestrian step frequency value provided by the step counter algorithm.

The Probability Mass Function (PMF) of a validity flag is binomial because there is a nonzero probability that an error in a step detector algorithm was made. The flag should be inverted with probability p_ε. Then, this PMF can be expressed as

$$\varepsilon_t = \|(u < \mathrm{p}_\varepsilon) - \check{\varepsilon}_t\|; \quad u \sim \mathcal{U}(0,1), \tag{3.17}$$

where $\check{\varepsilon}$ is the validity flag provided by the step counting algorithm, u is a uniformly distributed random number on the interval from zero to one.

Finally, the *ismoving* flag PMF is obtained as

$$\xi_t = \left\|(u < \mathrm{p}_\zeta) - \check{\zeta}_t\right\| \Big|\Big| \ (!\varepsilon_t); \quad u \sim \mathcal{U}(0,1), \tag{3.18}$$

where $\check{\zeta}_t$ is a step detection flag provided by the step counter algorithm, $\Big|\Big|$ is the logical *or* operator, and ! is the logical *not* operator. $\check{\zeta}_t$ is equal to one when the actual set of peaks and valleys detected in the measured acceleration was forming a sequence peak-zero-valley, which marks a valid step. However, the $\check{\zeta}_t$ flag is not valid when the pedestrian is standing still. For this reason, ξ_t is given as a binomial distribution of $\check{\zeta}$ with a probability of error p_ζ only when the control signal is valid ($\varepsilon_t = 1$). The *ismoving* flag is raised otherwise.

3.3 Performance of the Proposed Pedestrian Transition Model

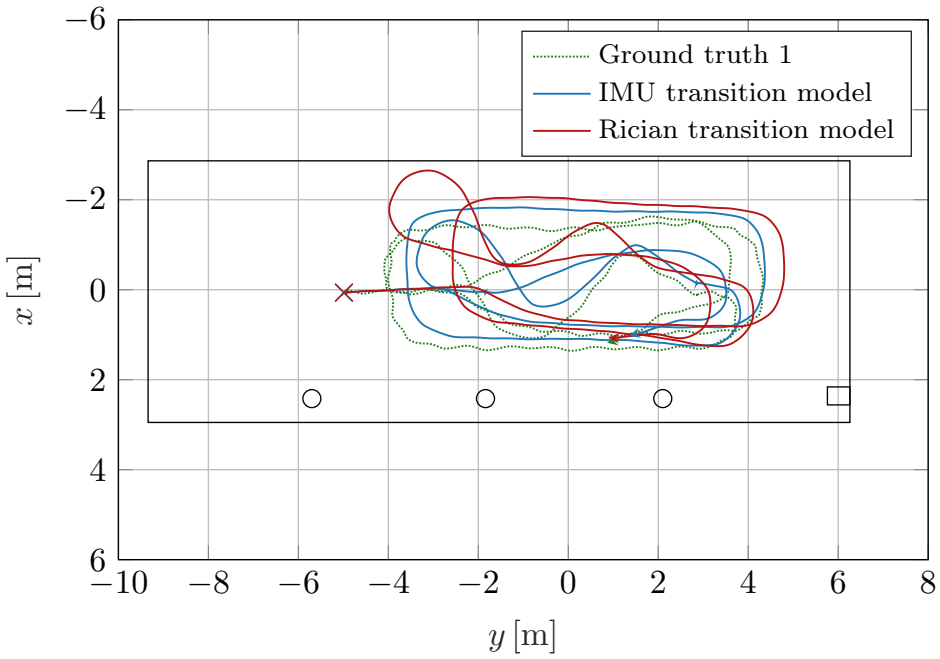
This section compares the proposed pedestrian transition model performance with the Rician transition model originally proposed in [Gen+17]. The new pedestrian transition model is based on the Rician transition model. However, it incorporates more information available from the IMU measurements. The main advantage is that more dynamic pedestrian movement is possible without the need for unreasonably high transition model variance. The extraction of the *ismoving* flag from the IMU measurements allows the pedestrian to stop suddenly and then continue walking. Since the pedestrian transition model estimates the step length, the walking speed can be inferred when pedestrians resume walking. The pedestrian step length is likely similar during the walk. The more precise transition model provides a more accurate position estimation of the multipath assisted positioning algorithm.

We used a Vicon motion capturer to track the movement of the pedestrians, hence, obtaining the ground truth of the pedestrian movement. The Vicon motion capture system can track the motion of the Vicon reflective marker in a room with a ground area of approximately 7 m by 4 m with an accuracy below 1 cm. The Vicon reflective marker, which the Vicon motion capturer tracked, was attached to the hand-held device.

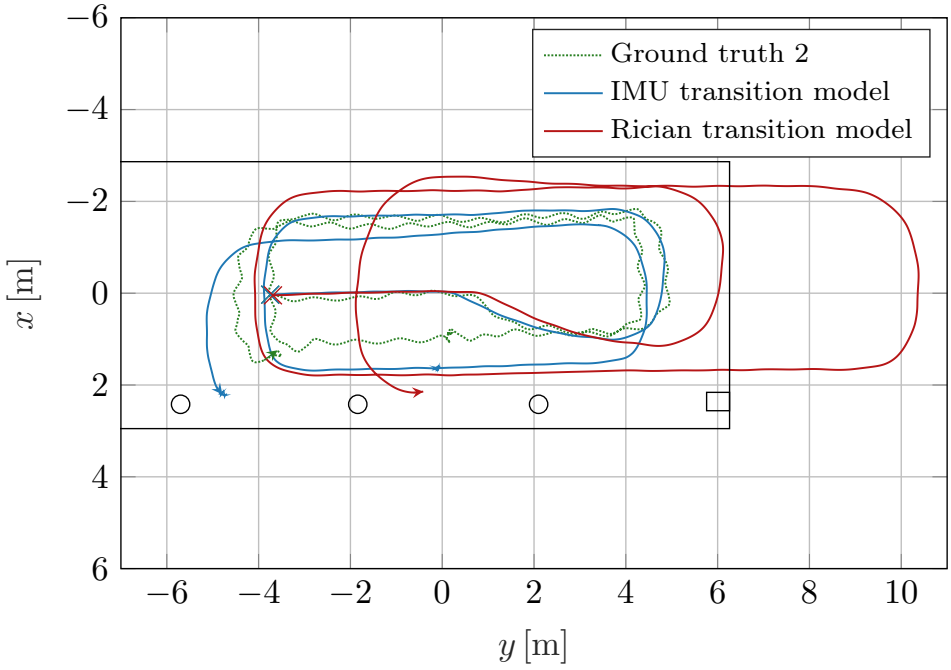
Two scenarios are evaluated to compare the transition models. Two volunteers with different heights and weights walked inside a room, handholding an Xsense (MTI-G-700) IMU while the Vicon motion capturer tracked their ground truth position. The IMU provides the calibration data, and the calibrated IMU measurements are used to evaluate position using Rician and IMU transition models.

Fig. 15 shows the room's layout where the experiments took place, the ground truth trajectories of the walking pedestrian, and the expectation of the predicted position PDF estimated based on the Rician and IMU transition models. A cross marks the start of the trajectory, and the arrow marks the end. The first trajectory in Fig. 15a contains two short walking interruptions. It is visible that the IMU transition model follows the actual trajectory closely, while the Rician transition model overshoots the actual trajectory already in the first turn. The measured data shows that both volunteers slow down before every turn. From the movement model, it seems that both volunteers keep the pedestrian step frequency similar but shorten their steps. The velocity estimation based on step length is not part of the formal Rician transition model. Hence, it cannot recognize the sudden change in velocity and overshoots the actual trajectory. The volunteer increased his walking pace right after the second turn, which is again successfully recognized by the new transition model. On the other hand, the Rician transition model cannot track the velocity change itself. This mismatch between the true and transition model velocity causes the Rician transition model to undershoot the actual trajectory. In contrast, the new IMU transition model keeps with the ground truth trajectory.

Similar behavior is observed in the second trajectory. Moreover, the pedestrian stopped at position $[1.0, 0.1]$ during this walk and waited for approximately 10 seconds before resuming the walk. Compared to the new IMU transition model, the Rician



(a) Scenario number one.



(b) Scenario number two.

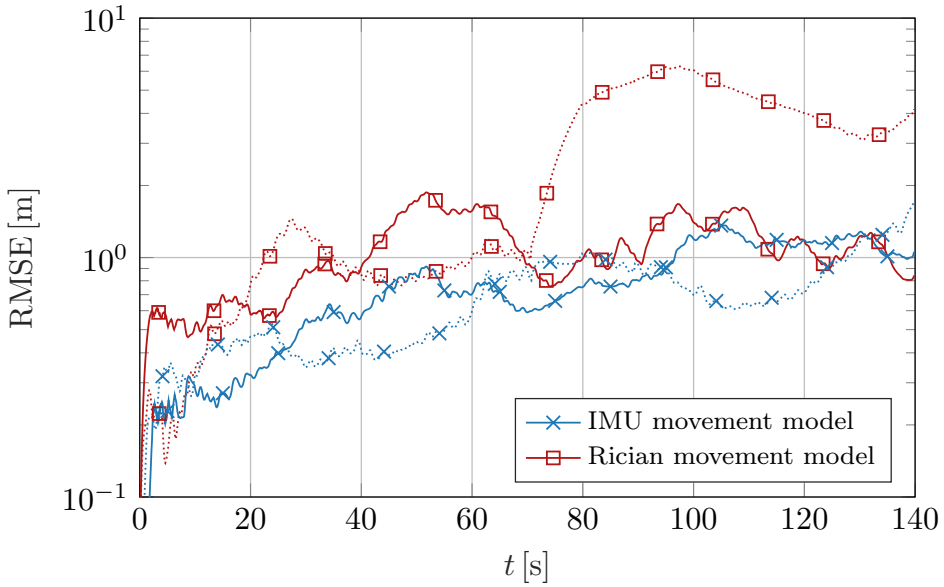
Figure 15: The comparison of the IMU and Rician transition models for two different pedestrian walking scenarios inside a room.

transition model cannot recognize that the pedestrian stopped. Thus, a huge error is accumulated during the standing still period.

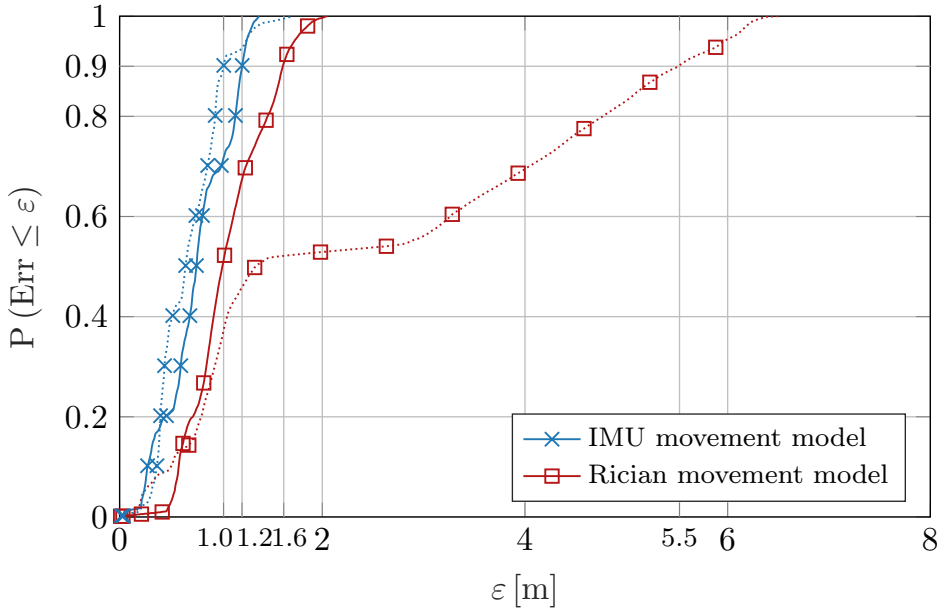
Fig. 16 evaluates the performance of the compared transition models. The solid and dotted lines illustrate the first and the second scenarios, respectively. Fig. 16a shows the evolution of Root-Mean-Square Error (RMSE) during the walk. Note that the RMSE axis has a logarithmic scale. The RMSE for the proposed IMU transition model significantly outperforms the Rician transition model or both scenarios. The first scenario has a similar RMSE towards the end of the walk. The RMSE of the Rician transition model visibly decreases after 65 seconds of the experiment. After the second loop, the Rician transition model position is ahead of the true position, but then the pedestrian increases his pace for a short time. Since the Rician transition model does not observe the speed increase, the ground truth trajectory catches up with the Rician transition model, canceling its high RMSE. The IMU transition model outperforms the Rician transition model over the whole period for the second trajectory. Moreover, a significant RMSE is accumulated between 70 and 80 seconds. The IMU transition model recognized that the pedestrian was standing still while the Rician transition model continued with the same speed.

Fig. 16b shows the overall performance of both transition models using an empirical Cumulative Distribution Function (CDF). Extra tics in the plot highlight the 90% confidence intervals. Note that the transition model variance causes the RMSE to increase over time. Hence, the 90% confidence interval increases with the experiment duration. This is not an issue since the duration of both experiments was the same.

The expected position provided by the new IMU transition model follows the ground truth better than for the Rician transition model. Hence, setting a lower WNA variance in (3.5) and (3.6) decreases the RMSE of the transition model. The smaller transition model variance consequently improves the precision of the Bayesian positioning algorithm combining the transition model prediction with the system observation, which is the aim of the proposed IMU transition model.



(a) The time evolution of the Root-Mean-Square Error (RMSE) of the estimated position.



(b) The Cumulative Distribution Function (CDF) of the estimated position error.

Figure 16: The evaluation of the IMU transition model performance compared to the Rician transition model. The solid and dotted lines illustrate the first and the second scenarios, respectively.

Chapter 4

Stochastic Data Association for Channel-SLAM

The main idea of multipath assisted positioning algorithms is that VTs are assumed to be static while the receiver is moving. Hence, it is a typical example of a SLAM, where the algorithm tracks the receiver position and simultaneously builds a map of VTs. However, the multipath assisted positioning algorithms require every observed radio channel MPC associated with a VT. The association of the observed MPCs with the corresponding VTs is a challenging problem. Most channel parameter estimators are snapshot-based algorithms, e.g. [Fle+99; HFR18]; thus, MPCs are not associated with previous observations of VTs. Using snapshot-based methods requires associating the observed MPCs with corresponding VTs in each time step. Channel estimation approaches like [Jos+12] track the MPCs over time but can suffer from outages. Hence, the stochastic data association is required even if the approach in [Jos+12] is utilized.

This chapter presents a stochastic data association method for multipath assisted positioning. The proposed stochastic data association method is based on the MHT [BL95], which allows uncertainty in the associations using a PF-based approximation of the data association PMF. Not only that the soft data association approach can recover from a wrong association, but it can also recognize an association error caused by the sequential radio channel estimator algorithm.

The derivation and results presented in this chapter are based on the article [KG20] I published. This chapter uses citations from [KG20].

4.1 Derivation of the Algorithm

This section outlines the essential steps of the proposed algorithm derivation, whose goal is to estimate the system's position $\mathbf{x}_{\mathrm{Rx},0:t}$. The concatenation of the system's position and other state parameters $\boldsymbol{\theta}$ defines the system's state. $\boldsymbol{\theta}$ contains parameters of the transition model. The pedestrian transition model described in Chapter 3 can be

assumed as a transition model. So, the system state is

$$\mathbf{x}_{\mathrm{u},0:t} = \left[\mathbf{x}_{\mathrm{Rx},0:t}^{\mathrm{T}}, \boldsymbol{\theta}_{0:t}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{4.1}$$

The VT positions and the constant propagation length offsets must be estimated simultaneously to benefit from a multipath radio channel. Thus, the VTs' system state

$$\mathbf{x}_{\mathrm{v},1:L\,0:t} = \left[\mathbf{x}_{\mathrm{Tx},1:L\,0:t}^{\mathrm{T}}, b_{\mathrm{v},1:L\,0:t}\right]^{\mathrm{T}} \tag{4.2}$$

concatenates the position of L VTs with constant propagation length offsets, where $\mathbf{x}_{\mathrm{Tx},1:L\,0:t}$ gives the cartesian position coordinates of all VTs for the whole duration until t, and $b_{\mathrm{v},1:L\,0:t}$ defines the additional propagation distances as in Section 1.3.

The whole system is the concatenation of a receiver state with all VTs states denoted as

$$\mathbf{x}_{0:t} = \left[\mathbf{x}_{\mathrm{u},0:t}^{\mathrm{T}}, \mathbf{x}_{\mathrm{v},1:L\,0:t}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{4.3}$$

The current state $\mathbf{x}_t$ is dependent on the previous states $\mathbf{x}_{0:t-1}$ and collected evidence. This work assumes two sources of evidence: the parameters of the MPCs estimated by KEST from $\mathbf{s}_{1:t}$ concatenated in measurement vector

$$\mathbf{z}_{1:t} = \left[\tau_{0\,1:t}, \ldots, \tau_{m\,1:t}, \ldots, \tau_{M-1\,1:t}\right]^{\mathrm{T}}, \tag{4.4}$$

and the output of the IMU after calibration and processing, which is then used as a control signal $\mathbf{u}_{1:t}$ as described in Chapter 3. Furthermore, a data association vector $\mathbf{n}_{1:t}$ is used, which matches the individual MPCs with the corresponding VTs in the system state. Otherwise stated, the data association vector n_{it} is an integer linking measurement z_{it} with $\mathbf{x}_{\mathrm{v},n_{it}\,t}$. Assume, for example, that the measurement vector consists of three delays $\mathbf{z}_t = [1.4, 2.5, 4.4]^{\mathrm{T}}$, and the Channel-SLAM currently consists of five VTs. If the first delay is associated with the second VT in $\mathbf{x}_{\mathrm{v},t}$, the second delay with the fifth VT, and the third delay with the first VT, the data association vector would be $\mathbf{n}_t = [2, 5, 1]^{\mathrm{T}}$.

Estimating the PDF of state $\mathbf{x}_{0:t}$ at the same time as the data association results in calculating a JPDF conditioned on the collected evidence

$$\mathrm{p}(\mathbf{x}_{0:t}, \mathbf{n}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}), \tag{4.5}$$

sometimes referred to as a posterior or belief function $\mathrm{bel}\,(\mathbf{x}_{0:t}, \mathbf{n}_{1:t})$.

The derivation of the proposed positioning algorithm is based on the previous work [Gen+16a; Gen+17]. In contrast, this work uses $\mathbf{n}_{1:t}$ directly during the whole derivation, which is necessary for the data association algorithm. This approach results in a slightly different derivation, but the main ideas of Channel-SLAM are preserved.

The first step during the derivation is to factorize the state posterior defined in (4.5) as

$$\begin{aligned}\overbrace{\mathrm{p}(\mathbf{n}_{1:t} \mid \mathbf{x}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}^{\text{data association posterior}} \overbrace{\mathrm{p}(\mathbf{x}_{0:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}^{\text{state posterior}} = \\ \overbrace{\mathrm{p}(\mathbf{n}_{1:t} \mid \mathbf{x}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}^{\text{data association posterior}} \overbrace{\mathrm{p}(\mathbf{x}_{\mathrm{u},0:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}^{\text{system state posterior}} \\ \prod_{\ell} \underbrace{\mathrm{p}(\mathbf{x}_{\mathrm{v},\ell\,0:t} \mid \mathbf{x}_{\mathrm{u},0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}_{\ell\text{-th VT state posterior}}.\end{aligned} \tag{4.6}$$

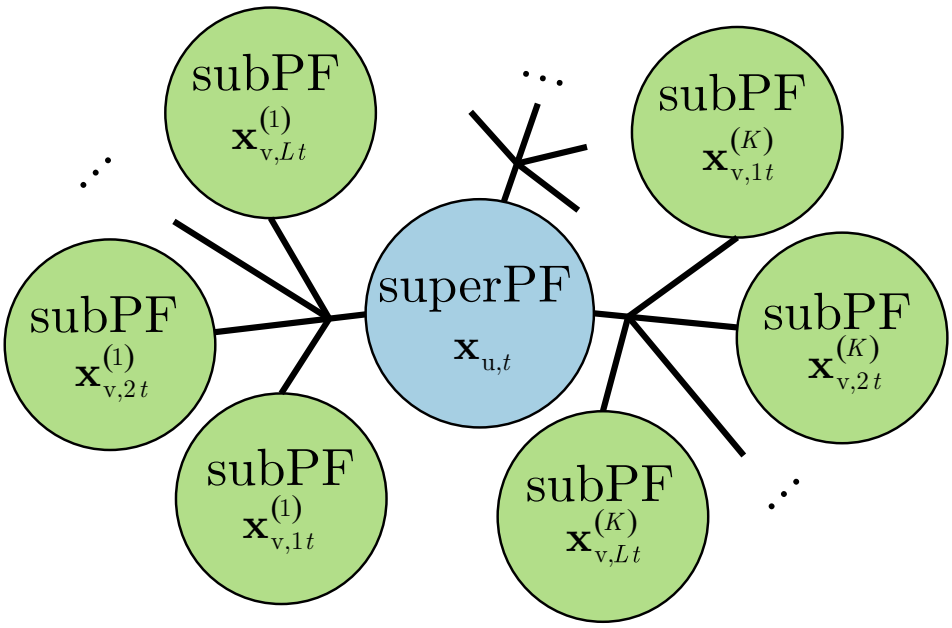


Figure 17: Graphical representation of the Rao-Blackwellized Particle Filter, where system state $\mathbf{x}_{\mathrm{u},t}$ is represented by superordinate PF, while for each system state particle $\mathbf{x}^{(k)}_{\mathrm{u},t}$, there is a set of independent subordinate PFs representing the state of the individual VTs. The state of the p-th particle of the ℓ-th VT linked with the k-th system particle is given by $\mathbf{x}^{(k\,p)}_{\mathrm{v},\ell\,t}$.

In the beginning, the posterior is factorized into the data association posterior and the state posterior using the product rule. The latter posterior is modified using the same rule to separate the state posterior into the system's $\mathbf{x}_{\mathrm{u},0:t}$ and VTs' $\mathbf{x}_{\mathrm{v},1:L\,0:t}$ posteriors. The last product term in (4.6) is obtained using the independence between the individual MPCs in the wireless channel. We can exploit this conditional independence to represent the system's state by a superordinate PF and each VT by the set of subordinate PFs. This concept is shown in Fig. 17, where each of the K particles representing $\mathbf{x}_{\mathrm{u},t}$ is assigned with the set of subordinate PFs representing L VTs. This separation into the set of conditionally independent posteriors is also known as a RBPF. For further information about the RBPF in Channel-SLAM, see [Gen+16a; Gen+17].

A closer examination of the first two terms on the r.h.s. of (4.6) reveals that it is a product of a data association and system state beliefs. Bayes' and product rule rewrite the term above as

$$\begin{aligned}\mathrm{bel}\left(\mathbf{n}_{1:t}\right)\mathrm{bel}\left(\mathbf{x}_{\mathrm{u},0:t}\right) =& \eta \mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t)\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}) \\ & \mathrm{p}(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}, \mathbf{z}_{t-1})\mathrm{p}(\mathbf{x}_{\mathrm{u},t} \mid \mathbf{x}_{\mathrm{u},t-1}, \mathbf{u}_t) \\ & \mathrm{bel}\left(\mathbf{n}_{1:t-1}\right)\mathrm{bel}\left(\mathbf{x}_{\mathrm{u},0:t-1}\right),\end{aligned} \tag{4.7}$$

where η is a normalizing constant given by the measurement prior. It is sufficient to perform normalization to obtain η. The derivation assumes that the current measurement vector $\mathbf{z}_t$ is conditionally independent of all previous system states $\mathbf{x}_{0:t-1}$ and data associations $\mathbf{n}_{0:t-1}$. If the first-order Hidden Markov Model (HMM) assumption holds,

the current state $\mathbf{x}_{\mathrm{u},t}$ depends only on $\mathbf{x}_{\mathrm{u},t-1}$ and current control signals $\mathbf{u}_t$. Similarly, the actual data association $\mathbf{n}_t$ only depends on the previous association $\mathbf{n}_{t-1}$. These assumptions allow rewriting (4.7) to a sequential form

$$\mathrm{bel}\left(\mathbf{x}_{\mathrm{u},0:t}\right) = \eta \mathrm{p}(\mathbf{x}_{\mathrm{u},0}) \prod_{i=1:t} \mathrm{p}(\mathbf{x}_{\mathrm{u},i} \mid \mathbf{x}_{\mathrm{u},i-1}, \mathbf{u}_i) \mathrm{p}(\mathbf{z}_i \mid \mathbf{x}_{\mathrm{u},0:i}, \mathbf{z}_{1:i-1}), \tag{4.8}$$

where the belief of the whole trajectory $\mathrm{bel}\left(\mathbf{x}_{\mathrm{u},0:t}\right)$ is obtained as a product of beliefs in all time steps.

Usually, it is not possible to calculate the most recent belief exactly. Still, according to [Aru+02], it is possible to approximate the desired posterior PDF function by a posterior PMF composed of samples from a PDF with assigned weights

$$\mathrm{bel}\left(\mathbf{x}_{\mathrm{u},t}\right) \approx \sum_k w_t^{(k)} \delta\left(\mathbf{x}_{\mathrm{u},t} - \mathbf{x}_{\mathrm{u},t}^{(k)}\right), \tag{4.9}$$

where $\delta\left(\cdot\right)$ is a Dirac delta function. The weights $w_t^{(k)}$ can be obtained as a division of the target PDF and importance PDF

$$w_t^{(k)} = \frac{\pi\left(\mathbf{x}_{\mathrm{u},t}^{(k)} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{n}_{1:t}\right)}{\mathrm{q}\left(\mathbf{x}_{\mathrm{u},t}^{(k)} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}, \mathbf{n}_{1:t}\right)}, \tag{4.10}$$

where the nominator is the target PDF, which ought to be approximated by particles. The denominator is the importance PDF from which the particles are sampled.

Based on the generic particle filter [Aru+02], the algorithm shows how to calculate the position and data association beliefs jointly. Using (4.7) as a target PDF and

$$\underbrace{\mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t) \overbrace{\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1})}^{\text{MPC tracking}}}_{\text{Data association PDF}} \underbrace{\mathrm{p}(\mathbf{x}_{\mathrm{u},t} \mid \mathbf{x}_{\mathrm{u},t-1}, \mathbf{u}_t)}_{\text{system state evolution PDF}} \tag{4.11}$$

as an importance PDF, the weight at time t is proportional to

$$w_{t|t-1}^{(k)} = w_t^{(k)} w_{t-1|t-2}^{(k)}, \tag{4.12}$$

where

$$w_t^{(k)} \propto \mathrm{p}\Big(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{z}_{t-1}\Big) \tag{4.13}$$

and

$$w_{t-1|t-2}^{(k)} \propto \mathrm{bel}\left(\mathbf{n}_{1:t-1}\right) \mathrm{bel}\left(\mathbf{x}_{\mathrm{u},0:t-1}\right). \tag{4.14}$$

The likelihood in (4.13) can be interpreted as a marginal PDF over transmitters $\mathbf{x}_{\mathrm{v},1:L\,t}$. By directly using the independence between the individual MPCs and the product rule, the marginal PDF can be written as

$$\mathrm{p}\Big(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{z}_{t-1}\Big) \propto \prod_\ell \int \mathrm{p}\Big(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{x}_{\mathrm{v},\ell\,t}^{(k)}, \mathbf{z}_{t-1}\Big) \mathrm{p}\Big(\mathbf{x}_{\mathrm{v},\ell\,t}^{(k)} \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{z}_{t-1}\Big) \mathrm{d}\mathbf{x}_{\mathrm{v},\ell\,t}^{(k)}. \tag{4.15}$$

As in [Gen+16a], (4.15) results in

$$w_{t|t-1}^{(k)} \propto \prod_{\ell} \sum_{p} w_{\ell\, t-1|t-2}^{(k\,p)} \mathrm{p}\Big(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{x}_{\mathrm{v},\ell\, t}^{(k\,p)}\Big), \tag{4.16}$$

where the measurement likelihood function is calculated using samples approximating the transmitter position $\mathbf{x}_{\mathrm{v},\ell\, t}^{(k\,p)}$ weighted by $w_{\ell\, t-1}^{(k\,p)}$ according to the multivariate PDF with a specified kernel function as in the regularized PF [Aru+02].

4.2 Pruned Multiple Hypothesis Tracking Filter

The previous section derives the PF-based approximation of the position and data association JPDF. However, it must be stressed that this approximation only fits for sampling from the importance PDF defined by (4.11), where the last term is the transition model described in Chapter 3. This section focuses on the first two terms, $\mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t)$, and $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1})$, and proposes a method for sampling from

$$\mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t)\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}). \tag{4.17}$$

The terminology used in this work is related to [BL95], where the reader can find a comprehensive description of the data association problem, including practical examples. A brief introduction of the terminology used in this section is in order.

The measurement is assumed to be information provided by a detector when a detection condition is fulfilled. Thus, it is not the whole set of measured values, only its subset which fulfills certain conditions, e.g., the strength of the signal or the number of occurrences in the number of consecutive measurements (m-of-n). In this case, the KEST algorithm serves as a detector and tracker providing us with estimations of MPC parameters. We use measurement vector $\mathbf{z}_t$ defined in (4.4) as measurements.

Even after being processed by the detector, some values in $\mathbf{z}_t$ can be caused by undesirable effects such as the thermal noise of the receiver or interference with other systems operating in the same radio frequency band. Such values of $\mathbf{z}_t$ are called false alarms. A value in $\mathbf{z}_t$, which is not a false alarm, is called detection. In other words, detection is a measurement caused by the target. The target is observed as an MPC caused by an LoS signal, a wall reflection, a scatterer, or any combination of possibly multiple signal reflections and scatterings. The continuous observation of the target is called a track. Finally, the set of false alarm measurements is called clutter.

The proposed algorithm is based on the MHT data association algorithm, and its derivation can be found in [BL95]. The premise of the MHT is to assume data association as a random variable and filter it using Bayesian filtration techniques. However, it requires an exhaustive search over the entire space of possible associations. The number of possible associations increases exponentially with the number of targets, which renders the MHT computationally infeasible in general. The MHT algorithm derivation is outlined to demonstrate how the data association algorithm is similar to MHT, how the reduction of computational complexity is achieved, and how much it costs.

First, I define several statistics based on data association $\mathbf{n}_t$, which will be advantageous for further derivations:

- The track indicator

$$\tau_i = \begin{cases} 1; & \text{if } z_{i\,t} \text{ is caused by a tracked MPC} \\ 0; & \text{otherwise} \end{cases} \tag{4.18}$$

  indicates that the i-th element of the measurement vector $\mathbf{z}_t$ originates from one of the already observed MPCs.

- The new target indicator

$$\nu_i = \begin{cases} 1; & \text{if } z_{i\,t} \text{ is caused by a new MPC} \\ 0; & \text{otherwise} \end{cases} \tag{4.19}$$

  indicates that $z_{i\,t}$ originates from an MPC, which was not yet observed, and the algorithm must create a new track.

- The detection indicator

$$\delta_\ell = \begin{cases} 1; & \text{if } \mathbf{x}_{\mathrm{v},\ell\,t} \text{ is detected at } t \\ 0; & \text{otherwise} \end{cases} \tag{4.20}$$

  tells whether the MPC linked with VT_ℓ is in the current measurement vector $\mathbf{z}_t$.

Note that sums $\tau = \sum_i \tau_i$, and $\nu = \sum_i \nu_i$ provide the number of targets in $\mathbf{z}_t$. Then, the number of false alarms is denoted by

$$\phi = |\mathbf{z}_t| - \tau - \nu, \tag{4.21}$$

where $|\cdot|$ is cardinality.

The product rule is utilized to derive the data association posterior PDF from (4.6) as

$$\mathrm{p}(\mathbf{n}_{1:t} \mid \mathbf{x}_{0:t}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \eta \mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t) \mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}) \mathrm{bel}\left(\mathbf{n}_{1:t-1}\right), \tag{4.22}$$

where the independent variables are dropped assuming first-order HMM. Notice that (4.22) is equivalent to the data association belief $\mathrm{bel}\left(\mathbf{n}_{1:t}\right)$ from (4.7).

The first PDF on the r.h.s. of (4.22) is the likelihood function and can be evaluated using indicators (4.18), (4.19), and (4.20) as

$$\mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t) = V^{-\phi} \prod_{i=1}^{|\mathbf{z}_t|} \mathrm{p}(z_{i\,t})^{\nu_i} \left[\mathrm{p}(z_{i\,t} \mid \mathbf{x}_{\mathrm{u},t}, \mathbf{x}_{\mathrm{v},n_{i\,t}\,t})\right]^{\tau_i}, \tag{4.23}$$

where $n_{i\,t}$ index the ℓ-th VT. In cases when $z_{i\,t}$ is not associated with the ℓ-th VT, the likelihood is given by a uniform PDF in a volume of interest V. Thus, for an unassociated element of $\mathbf{z}_t$, the likelihood is V^{-1}.

The second term $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1})$ in (4.22), which can be interpreted as the evolution of a data association state, is assumed to be independent of the previous step $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}) = \mathrm{p}(\mathbf{n}_t)$ when the KEST labels are unavailable. In this case, the data association algorithm can calculate only the prior $\mathrm{p}(\mathbf{n}_t)$ of data association.

When the data association algorithm obtains a measurement vector of cardinality $|\mathbf{z}_t|$, it needs to label each delay in $\mathbf{z}_t$ as either one of the previously observed VTs, or newly observed VT, or a false alarm.

The possible label assignments, how to assign $|\mathbf{z}_t|$ measurements with $|\mathbf{z}_t| - \phi$ VTs[1], are equally probable. However, one must evaluate how many unique label assignments can be composed to calculate the value of $\mathrm{p}(\mathbf{n}_t)$.

Using the summed indicator functions τ, ν, and ϕ, and the fact they are fully defined by $\mathbf{n}_t$, according to Section 6.3 in [BL95], it is possible to write

$$\begin{aligned} \mathrm{p}(\mathbf{n}_t) =& \mathrm{p}(\mathbf{n}_t, \tau(\mathbf{n}_t), \nu(\mathbf{n}_t), \phi(\mathbf{n}_t)) \\ =& \mathrm{p}(\mathbf{n}_t \mid \tau(\mathbf{n}_t), \nu(\mathbf{n}_t), \phi(\mathbf{n}_t)) \mathrm{p}(\tau(\mathbf{n}_t), \nu(\mathbf{n}_t), \phi(\mathbf{n}_t)), \end{aligned} \tag{4.24}$$

where the conditional term in (4.24) is calculated based on combinatorics. The number of possibilities of how to assign $|\mathbf{z}_t|$ measurements with τ previously observed targets is the number of permutations. The remaining possibilities are the combinations of how to label the remaining $\nu + \phi$ measurements as either a new track or a false alarm. Finally, the last term in (4.24) is modeled using clutter models.

However, the KEST provides its label estimate yielding additional information

$$\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}), \tag{4.25}$$

that benefits the data association algorithm. The idea is to use the label of the MPC provided by the KEST algorithm as a simple hard decision data association and use the soft decision data association algorithm only to resolve the remaining MPCs. This approach reduces the calculational complexity of the data association algorithm.

The individual MPCs might not always be visible, causing outages to occur. A tracking algorithm resolves a short outage, but a more extended outage causes the MPC to disappear during the tracking. The KEST algorithm initializes a new track when a new MPC appears and assigns a unique label. The KEST does not assume re-tracking. Hence, the KEST removes a tracked MPC from its state when it disappears for several consecutive radio channel measurements. We refer to removing the MPC from a KEST state as the KEST outage. The most common cause of the KEST outage is shadowing the signal path or the signal propagation distance exceeding the transmitter's range. For details on how the KEST algorithm decides which MPC to remove, see [Jos+12].

It is not possible to use only the KEST labels directly as a data association for two reasons:

- After an outage in KEST, the label of MPC is dropped, and when the MPC originating from the same VT reappears, it is assigned a new label. Hence, the

[1]Note that if the number of false alarms in measurement vector $\mathbf{z}_t$ is ϕ, then the number of VTs in $\mathbf{z}_t$ is $|\mathbf{z}_t| - \phi$.

Channel-SLAM would initialize a new VT even though the same VT has already been observed.

- Due to the "natural" data association used in KEST [Jos+12], the measurement can be assigned an incorrect KF causing an association error. Every error in KEST's association would cause a hard decision error leading to the divergence of the positioning algorithm.

Hence, if available, the proposed approach uses a KEST-based association to shape the data association PMF as $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1})$. Otherwise, it remains independent of history $\mathbf{n}_{t-1}$ [BL95]. When the label of the i-th MPC provided by KEST is the same as the label previously linked with any of the VTs, the algorithm links it again with the same VT. This is assured by setting $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}) = 0$ otherwise. We illustrate this in an example: Vector $\mathbf{z}_{t-1}$ consists of three measurements with labels $L_{t-1} = [11, 22, 33]$. The measurements were assigned as VT 2, 3, and 1; $\mathbf{n}_{t-1} = [2, 3, 1]$. In the following time step, $\mathbf{z}_t$ consists of four measurements with $L_t = [11, 44, 55, 22]$. Since labels 11 and 22 were observed in the previous time step and are still observed in the current time step, the data association trusts the KEST algorithm and sets $n_{1\,t} = 2$ and $n_{4\,t} = 3$. The probability of any $\mathbf{n}_t$ violating the previous association is assumed to be zero. Hence, (4.22) does not need to be evaluated for associations $n_{1\,t} \neq 2$ and $n_{4\,t} \neq 3$, which significantly reduces computational complexity.

Using the KEST labels significantly decreases the number of possible $\mathbf{n}_t$ for which (4.22) needs to be calculated. Only the newly occurring labels need to be handled. At this moment, the data association algorithm becomes suboptimal because the small probability that KEST is tracking an errant path is approximated by zero. An algorithm resolving possible association errors in the KEST is proposed at the end of this section.

Finally, the remaining unassociated measurements in a subset of $\mathbf{z}_t$, indicated by $\mathbf{z}'_t$, must be associated with the corresponding VTs. As shown by the derivation in Section 4.1, sampling from (4.11) allows estimation of the system state JPDF. Therefore one needs to obtain samples from

$$\mathrm{p}(\mathbf{z}_t \mid \mathbf{n}_t, \mathbf{x}_t)\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}). \tag{4.26}$$

Now, assuming all elements of $\mathbf{z}_t$ labeled with the same labels as the previously observed VTs are already assigned together, then (4.26) is proportional to

$$\mathrm{p}(\mathbf{z}'_t \mid \mathbf{n}'_t, \mathbf{x}'_t)\mathrm{p}(\mathbf{n}'_t), \tag{4.27}$$

where the VTs associated with elements of $\mathbf{z}_t$ are omitted from $\mathbf{x}_t$ to obtain subsets, marked by an apostrophe, excluding already assigned data. Because of this, the remaining associations are independent of history, and $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1})$ becomes an association prior $\mathrm{p}(\mathbf{n}'_t)$.

Essentially, the number of elements in $\mathbf{z}'_t$ is usually zero, and rarely is it greater than one. It means that, in most cases, data association is resolved solely by the KEST algorithm. When the association according to (4.27) is needed, its complexity usually

reduces to assigning one measurement with one of the currently unassigned VTs from $\mathbf{x}'_t$, with a new VT while creating $\mathbf{x}_{\mathrm{v},L+1\,t}$, or with a false alarm. However, when $|\mathbf{z}'_t| > 1$, the algorithm resolves its elements one by one, decreasing the space's size to search for each resolved element of $\mathbf{z}'_t$.

In other words, evaluating the PMF using (4.22) provides a subset of the most likely $\mathbf{n}_t$ while setting $\mathrm{p}(\mathbf{n}_t) = 0$ for all others $\mathbf{n}_t$. The association is sampled from the obtained PMF. Since the association is sampled for each superordinate particle separately, the algorithm is a soft decision data association algorithm related to the MHT filter. Omitting the unlikely combinations of $\mathbf{n}_t$ is also known as pruning [BL95].

The algorithm associates a single element of $z'_{i\,t}$ at a time. Hence, the sampling from (4.27) is done as follows. First, calculate

$$\mathrm{p}(z'_{i\,t} \mid n'_{i\,t}, \mathbf{x}'_t)\mathrm{p}(n'_{i\,t}) \tag{4.28}$$

for every possible value of $n'_{i\,t} = 1 : L'$, where L' is the number of unassociated VTs. Then, sample the desired number of particles from this PMF using a systematic resampling scheme [KLZ13]. Finally, remove $z'_{i\,t}$ and $\mathbf{x}'_{\mathrm{v},n'_{i\,t}\,t}$ from $\mathbf{z}'_t$ and $\mathbf{x}'_t$. This process is repeated while $|\mathbf{z}'_t| > 0$.

To evaluate (4.28) still needs a solution for the prior PMF $\mathrm{p}(n'_{i\,t})$, e.g., in [BL95]. However, the proposed data association algorithm does not calculate associations jointly. Hence, the prior PMF is uniform for association with one of the older, currently unassociated VTs. Then, the probability that the $z'_{i\,t}$ is a false alarm is given by a constant probability of false alarm P_{FA}. Similarly, the PMF of a new target occurrence is the probability of a new target P_{N}. This can be written as

$$\mathrm{p}(n'_{i\,t}) = \begin{cases} P_{\mathrm{FA}}; & i = -1 \\ P_{\mathrm{N}}; & i = 0 \\ \dfrac{1 - P_{\mathrm{FA}} - P_{\mathrm{N}}}{|\mathbf{z}'_t|}; & \text{else.} \end{cases} \tag{4.29}$$

Joint data association requires modeling the false alarm and the new target occurrence as Poisson distributed random variables. Also, the number of possible associations would increase exponentially with the number of elements of $\mathbf{z}'_t$, rendering the joint data association into an extremely computationally demanding algorithm.

The final part addresses the situation where the KEST algorithm can make an incorrect data association. A KEST association error occurs when different targets cause almost identical observations. This situation typically occurs when the pedestrian moves close to a wall, corners, or scatterer. Then the LoS delay is almost identical to the delay of MPC caused by the reflection from a nearby obstacle. This problem is especially severe if AoA information is unavailable. Using AoA information, the algorithms [Ulm+17] and [Thr+04] can resolve overlapping MPCs delays. For these methods, the data association provided by KEST labeling can be sufficient since the approximation, $\mathrm{p}(\mathbf{n}_t \mid \mathbf{n}_{t-1}) = 0$, is nearly true.

However, the proposed algorithm uses low-cost hardware without the AoA capability. Therefore, the data association algorithm cannot ignore the KEST association error. The

data association algorithm performs stochastic tests to keep or discard the associated tracks based on the association error probability

$$\mathrm{p}_{\mathrm{err}}^{(kp)} = 1 - \frac{1}{P\sqrt{2\pi Q}} \sum_{p=1}^{P} \mathrm{p}\Big(\mathbf{z}_t \mid \mathbf{x}_{\mathrm{u},t}^{(k)}, \mathbf{x}_{\mathrm{v},\ell\,t}^{(k\,p)}\Big), \tag{4.30}$$

where P is the number of particles used to approximate the ℓ-th transmitter $\mathbf{x}_{\mathrm{v},\ell\,t}$, and Q is the delay measurement variance. Note that this solution uses only the current time step likelihood since the RBPF weight normalization does not conserve track likelihood information. Using the current step likelihood results in a pessimistic association error probability. Therefore, the MHT decreases the number of discarded tracks using the power operation in the track discard condition

$$u < \left(\mathrm{p}_{\mathrm{err}}^{(kp)}\right)^{c_{\mathrm{err}}}; \quad u \sim \mathcal{U}\,(0,1)\,, \tag{4.31}$$

where u is a number generated from a uniform distribution $\mathcal{U}\,(0,1)$ on the interval from zero to one. c_{err} is a constant improving the algorithm's robustness without the necessity of increasing the number of particles.

4.3 Performance Using Stochastic Data Association

The measurements were conducted using an UWB system based on the *Decawave DWM1000* anchor using a two-way-ranging method (see [GU17]). The *Decawave DWM-1000* ToA estimation precision is 10 cm. The measurement setup consists of one static UWB anchor and a hand-held device carried by a walking pedestrian. The hand-held device includes a UWB tag, an Xsense IMU (MTI-G-700), and a laptop that stores the IMU and UWB measurement data. The UWB system is configured to a bandwidth of 500 MHz and a carrier frequency of 3.9 GHz for the measurements. The *Decawave DWM1000* anchor can provide the measured CIR in addition to the ranging information.

The measurement took place in a rectangular room with support pillars on one side. Two different volunteers performed the experiment. Both volunteers were changing speed and completely stopped, at least once for a short period during the walk. The duration of each experiment was 140 seconds. Fig. 18 shows the environment layout, the ground truth of the pedestrian movement, the UWB anchor position indicated by VT_1, and the VT positions VT_2–VT_6.

During the experiment, the Vicon motion capture system captured the ground truth of the pedestrian movement, similarly to Section 3. The handheld Xsense (MTI-G-700) captured the IMU measurements. Finally, the *Decawave DWM1000* anchors provided the CIR measurements. The positioning was carried out offline using the collected data.

Evaluations of Precision Based on a Simulation

The performed simulation shows the performance of the proposed data association algorithm. The simulation assumes one physical transmitter, three wall reflections, and two

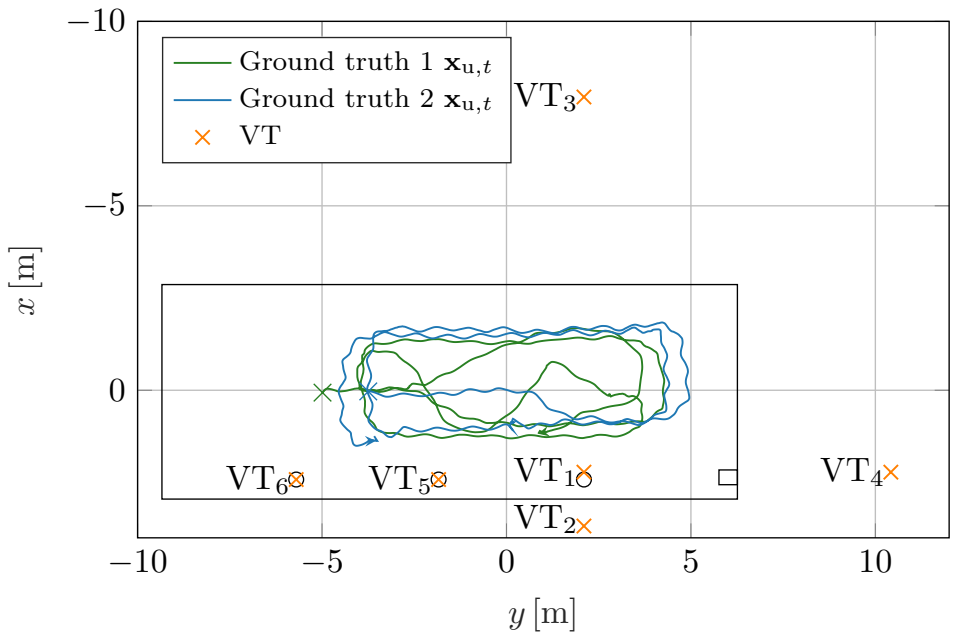


Figure 18: The environment layout with the ground truth of the pedestrian movement during the performed experiments. The cross and arrow mark the beginning and end of the experimental trajectory, respectively. The blue crosses mark the positions of six virtual transmitters. VT_1 is the position of the physical transmitter. VT_2–VT_4 are caused by the reflections from the room walls. VT_5–VT_6 are caused by scattering the transmitted signal on the room support pillars.

scatterers. Based on the geometry, the positions of the VTs are obtained as described in Section 1.3. The measurement vector $\mathbf{z}_{1:t}$ is simulated as a noisy signal propagation distance between the VT and the ground truth position $\mathbf{x}_{\mathrm{u},1:t}^{\mathrm{GT}}$, including the additional propagation length $b_{\mathrm{v},t}$ for scatterers VT_5–VT_6. The assumed noise of $\mathbf{z}_{1:t}$ is normally distributed with STD $\sigma_n = 0.1$ m, corresponding to the precision of the UWB system based on the *Decawave DWM1000* anchor.

As with the real experiment, the simulation assumes outages in the MPCs observation. Similarly as in an actual KEST-based estimation, the simulator generates a new KEST label after an outage. The probability of the MPC outage is selected for each MPC separately to achieve visibility for a desired percentage of the time. The outage duration is generated as a uniformly distributed random number between zero and 10 seconds $\mathcal{U}(0, 10)$ when an outage occurs. The percentage of visibility was selected 0.8, 0.7, 0.6, 0.5, 0.4, and 0.3 for MPC_1–MPC_6 respectively.

The movement model uses the recorded IMU measurements as input because creating an IMU measurement simulator for a walking pedestrian is not part of this work. The IMU measurements serve as control signals for the movement model, see Chapter 3. The trajectory recorded by the Vicon motion capturer serves as the ground truth of the pedestrian movement $\mathbf{x}_{\mathrm{u},1:t}^{\mathrm{GT}}$ for the simulation of $\mathbf{z}_{1:t}$ and the precision evaluation. This approach shows the performance of the proposed Channel-SLAM, including the data

Table 4.1: Parameters used for the evaluation of the simulations.

K	2000	c_P	1000
Q	$0.36\,\mathrm{m}^2$	σ_{w_x}	$0.15\,\mathrm{m/s}^{3/2}$
σ_{w_y}	$0.15\,\mathrm{m/s}^{3/2}$	σ_{BS}	$10\,\mathrm{deg/h}$
σ_γ	$5 \cdot 10^{-3}\,\mathrm{m/s}^{3/2}$	σ_{Rx}	$1 \cdot 10^{-3}\,\mathrm{m/s}^{1/2}$
σ_{ARW}	$10\,^\circ/\sqrt{\mathrm{hr}}$	$\sigma_{\ddot{\beta}}$	$1 \cdot 10^{-4}\,\mathrm{m/s}^4$
σ_ω	$1 \cdot 10^{-4}\,\mathrm{Hz}$	p_ε	$1 \cdot 10^{-3}$
p_ζ	$1 \cdot 10^{-3}$	c_{err}	3.8

association, compared to the state-of-the-art method presented in [Gen+17].

Table 4.1 summarizes the values of the parameters used in the simulation. The parameters of the movement model fit all the measured scenarios.

The STD of the *Decawave DWM1000* anchor ToA estimation is 10 cm. However, the STD of the MPC delay estimate is higher due to a low SNR of the reflections. We set the MPC variance $Q = 0.36\,\mathrm{m}^2$ for the experiments. Therefore, the simulation uses the same value.

The bias stability of the Xsense IMU (MTI-G-700) gyroscope is 10°/h, which is directly used as the movement model parameter σ_{BS}. Similarly, the noise density of the IMU gyroscope is $0.01°/\mathrm{s}/\sqrt{\mathrm{Hz}}$, which corresponds to the angular random walk of 6°/h when a 100 Hz bandwidth is assumed. The parameter tuning showed that the increased value $\sigma_{\mathrm{ARW}} = 10\,^\circ/\sqrt{\mathrm{hr}}$ yields better performance.

The parameters linked with the pedestrian velocity (σ_{w_x}, σ_{w_y}, σ_γ, $\sigma_{\ddot{\beta}}$, and σ_ω) are set to envelop the walking dynamics similar to the WNA model [BLK01]. However, when the chosen parameters are too large, the deviation of the superordinate particles is also large, resulting in a high number of modes of the estimated posterior PDF, which significantly decreases the precision.

Fig. 19 shows an example of the simulated CIRs for recorded scenario number one. The random outages of the noisy MPC observations provide a short time track similar to an actual MPC parameter estimation provided by KEST. The simulation is simplified by using a constant amplitude for all MPCs because Channel-SLAM does not directly use the amplitude information.

Fig. 20 shows the recorded ground truth of the pedestrian movement, one realization of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model, and one realization of the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The position estimators use simulated MPC parameters. The transition models use IMU measurements. Fig. 20a and Fig. 20b show the first and the second scenario, respectively.

Fig. 21 compares the performance of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art approach without the stochastic data association and with the Rician transition model. The solid and dotted lines illustrate the first and second scenarios, respectively.

Fig. 21a shows the RMSE evolution throughout the experiments. In contrast to

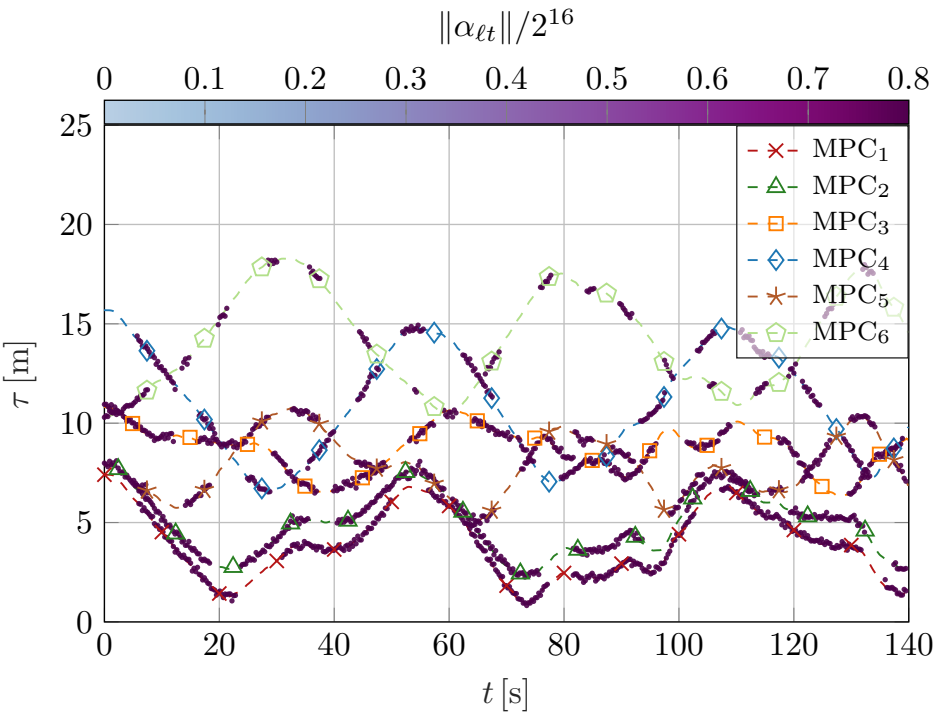
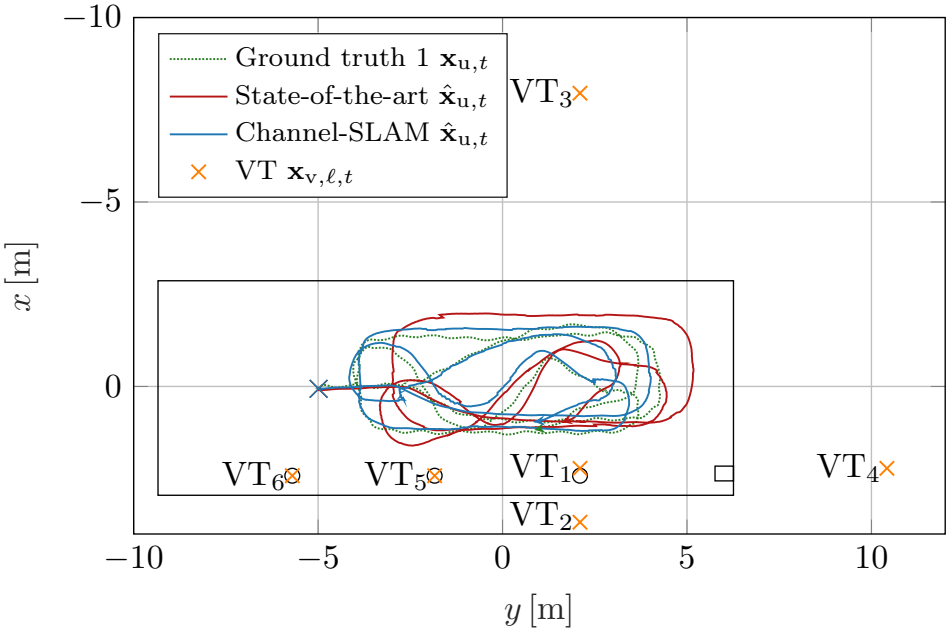


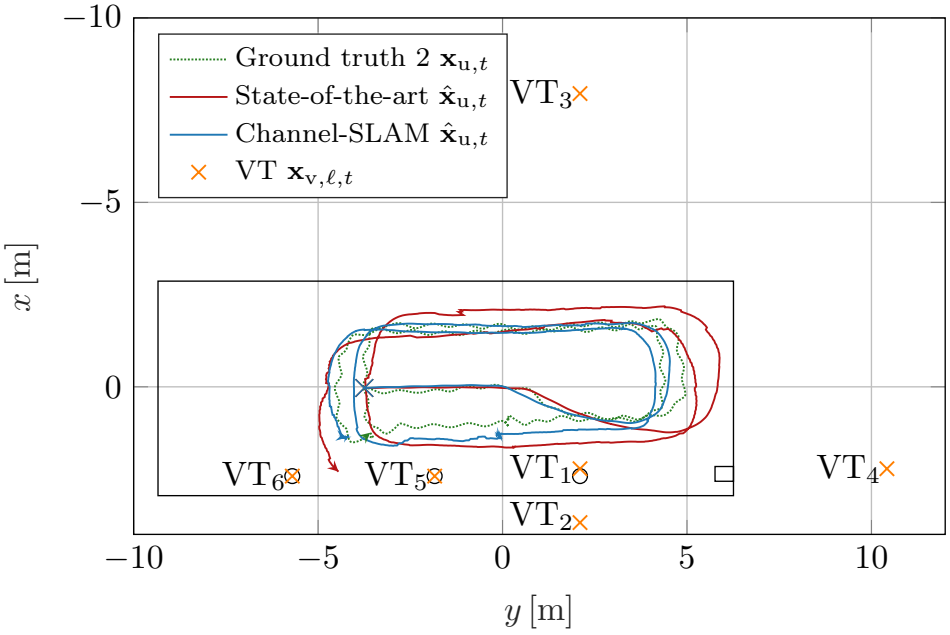
Figure 19: The simulated CIR with ground truth MPCs for scenario number one. The MPC outage is designed to achieve the time percentage of visibility 0.8, 0.7, 0.6, 0.5, 0.4, and 0.3 for MPC$_1$–MPC$_6$, respectively. The simulation does not include the magnitude of the measured impulse response but only the delays.

the formal state-of-the-art, the RMSE of the proposed algorithm never exceeds 1 m for either scenario. The positioning algorithm with the Rician transition model accumulates significant RMSE due to sudden pedestrian speed changes. The RMSE can be more than three times better for the proposed Channel-SLAM. The decrease of the RMSE visible in the state-of-the-art approach after 60 s and 120 s of the first experiment is caused by the significant bias between the actual and estimated speed after a sudden change in the pedestrian's walking style. The velocity bias can cause the movement model to catch up with the actual position causing the unintuitive behavior where the RMSE decreases. This is a severe problem since the model misalignment causes a significant bias between the estimated posterior PDF and the optimal posterior PDF obtained if the positioning algorithm correctly incorporates available information. The result of such a biased posterior PDF is that the positioning algorithm is not aware of its true precision, which could lead to severe damage. The newly proposed approach minimizes the posterior PDF bias by using a pedestrian transition model that models the specific nature of the pedestrian walk better than the Rician transition model.

The empirical CDFs in Fig. 21b show the overall comparison of the proposed Channel-SLAM with the state-of-the-art approach. The precision of the proposed algorithm outperforms the state-of-the-art approach in each simulated experiment. The additional grid lines mark the 90% confidence interval of the RMSE for each plot. For the first sce-

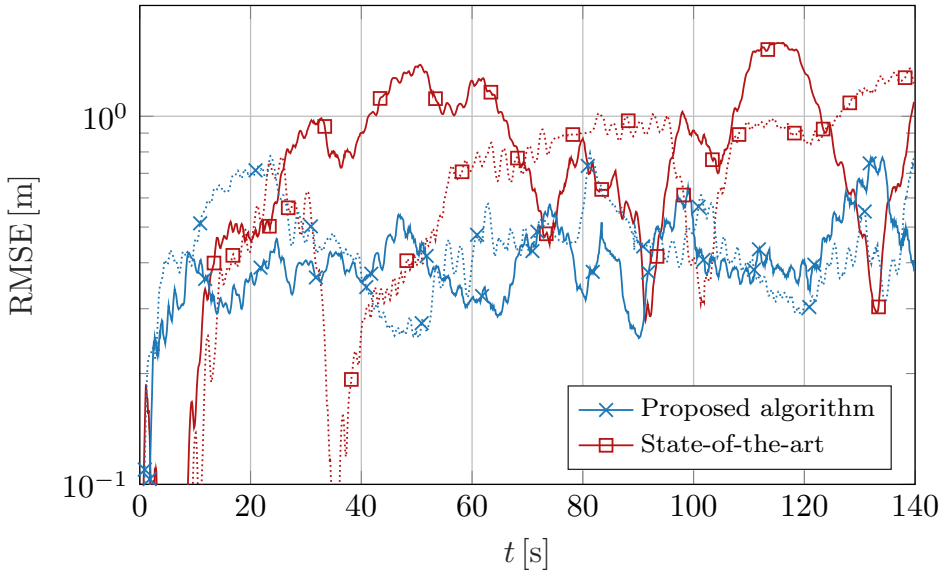


(a) The simulation of scenario number one.

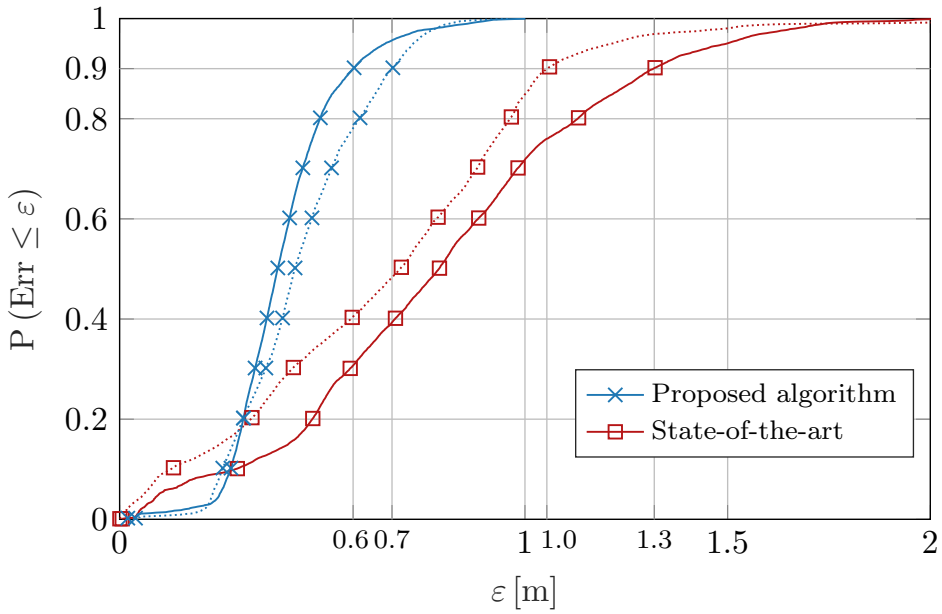


(b) The simulation of scenario number two.

Figure 20: The layout of testing scenarios and the comparison of the performance of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The position estimators use simulated MPC parameters. The transition models use IMU measurements.



(a) The RMSE comparison throughout the experiment duration.



(b) The RMSE empirical CDF over the runs of the simulation. The additional grid lines mark the 90% confidence interval for each plot.

Figure 21: The comparison of the performance of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The solid and dotted lines illustrate the first and the second scenarios, respectively. The position estimators use simulated MPC parameters. The transition models use IMU measurements.

nario, the 90% confidence interval of the proposed algorithm's RMSE is ±0.58 m, while for state-of-the-art, it is ±1.32 m, and ±1.21 m is achieved using only the IMU-based movement model. For the second scenario, the 90% confidence interval of the proposed algorithm's RMSE is ±0.67 m, while for state-of-the-art, it is ±1.05 m, and ±1.03 m is achieved using only the IMU-based movement model.

The algorithms' computational complexity depends not only on the setting of the algorithm parameters but also on the number of the MPCs, values of the MPC parameters, and the number of KEST outages. The algorithms implemented in *MATLAB R2018b* and executed on a desktop computer took, on average, 96.6 min and 91.3 min for the proposed and the state-of-the-art algorithms, respectively, for one run of the first scenario. For the second scenario, 73.4 min and 71.5 min for the proposed and the state-of-the-art algorithm, respectively.

Comparing the proposed Channel-SLAM with the state-of-the-art approach shows that the data association capability and the pedestrian transition model yield significantly better position estimation performance.

Evaluations of Precision Based on Measurement

This section compares the precision achieved by the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model [Gen+17]. The MPCs extracted from the CIR using the KEST algorithm are used to evaluate the position estimation precision. The CIR is measured using one low-cost *Decawave DWM1000* anchor.

Fig. 23 shows the recorded ground truth of the pedestrian movement, one realization of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model, and one realization of the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The position estimators use estimated MPC parameters obtained from measured CIR using one *Decawave DWM1000* anchor placed at position VT_1. Fig. 23a and Fig. 23b show the first and the second scenario, respectively. The parameters used in the experiment are the same as the parameters used for the simulation, given in Table 4.1.

Fig. 22 shows the output of the KEST algorithm estimating the parameters of the MPCs from the recorded CIR during experiment scenario number one. The scatter plot color map indicates the estimated MPC magnitude. Additionally, based on the ground truth of the pedestrian movement, the scatter plot shows the calculated theoretical delay of the MPCs MPC_1–MPC_6 originating from VT_1–VT_6, respectively. The scatter plot shows more than the six MPCs. Those additional MPCs might be caused by multiple reflections of the transmitted signal and by other scatterers. For the sake of plot readability, the plot shows only MPC_1–MPC_6. However, the positioning algorithm utilizes all of the available MPCs.

We can see that the calculated theoretical delay indicated by the dashed lines in Fig. 22 matches the KEST estimate when detectable. However, the MPCs are often distorted and cannot be tracked all the time reliably. Hence, the algorithm must be ro-

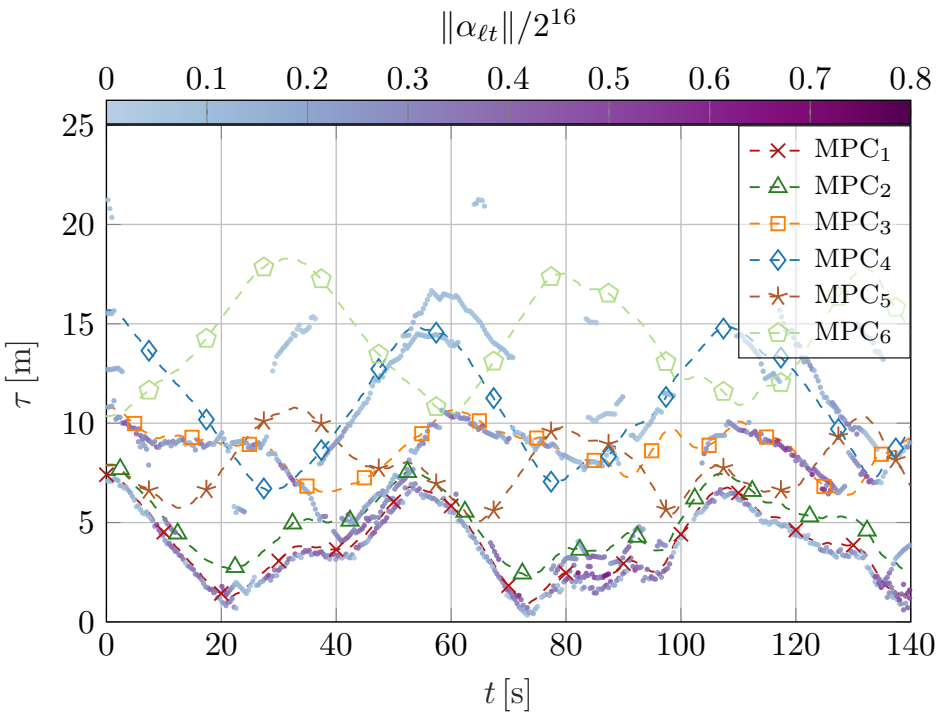


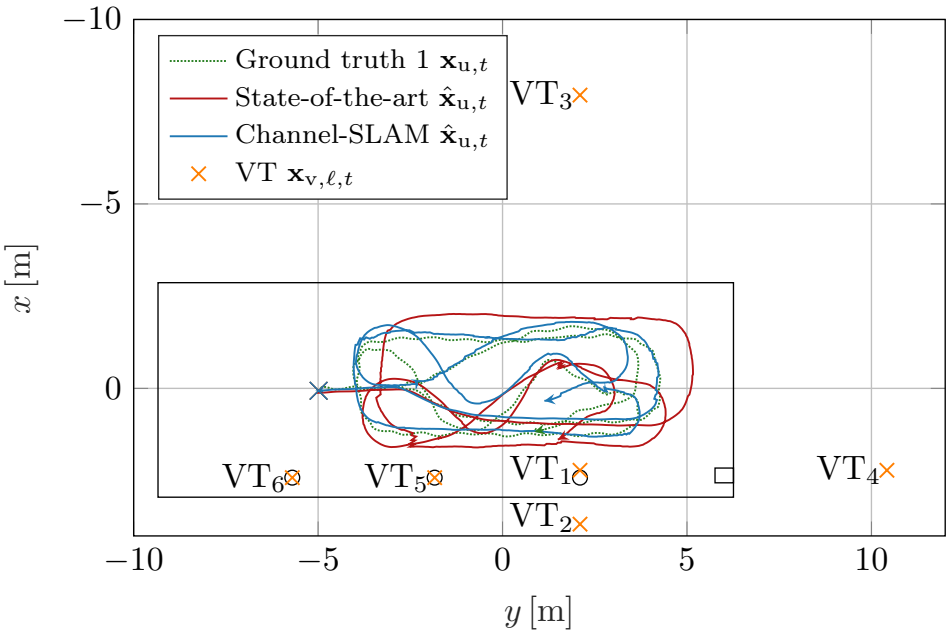
Figure 22: Comparison of the estimated CIR with the ground truth MPCs originating from VTs for scenario number one. The magnitude of the estimated CIR $\|\alpha_{\ell t}\|$ is shown by the scatter plot color. The receiver provides the amplitude measurement as a 16-bit number. The individual MPCs can be easily lost due to shadowing by the pedestrian's body and by other obstacles.

bust to deal with outages occurring even more often than for the professional broadband channel sounder used in [Gen+17] when low-cost hardware is used.

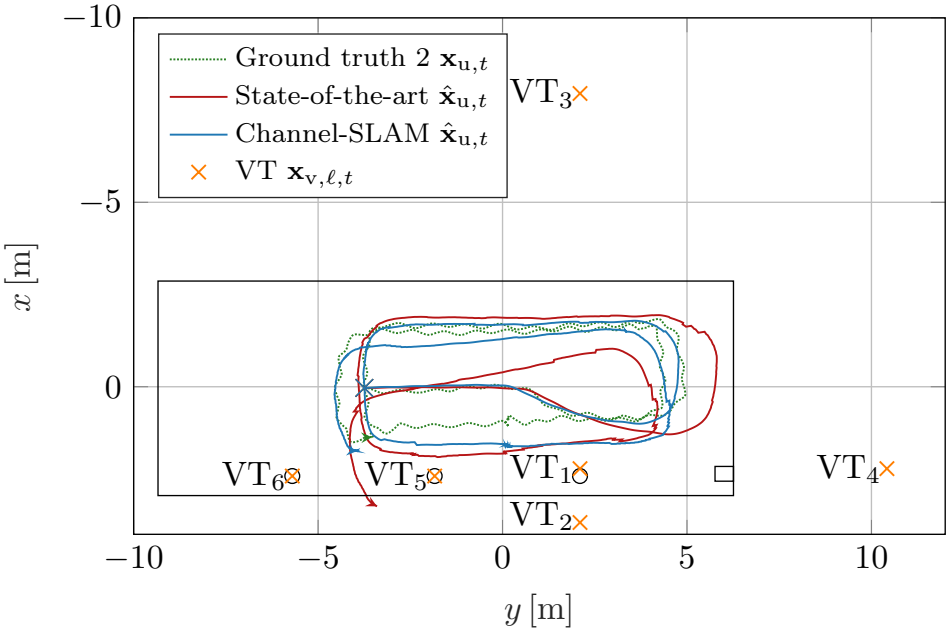
Additionally, KEST may associate incorrect MPCs during the tracking. For example, this is visible at $t = 35\,\mathrm{s}$, where MPC$_3$ is wrongly assigned, and its delay decreases towards MPC$_2$. Since only MPC$_1$ and MPC$_3$ are observed at that point, this faulty assignment would cause a significant positioning error by steering the estimated position sideways.

However, this wrongly tracked MPC is successfully recognized by the KEST error detection algorithm described in Section 4.2, and the KEST-based association is dropped. The effect of this algorithm is apparent in scenario one, marked using solid lines, in Fig. 24a, where the RMSE of the state-of-the-art algorithm starts to increase. In contrast, the RMSE of the proposed data association algorithm starts to decrease shortly after.

The proposed Channel-SLAM improves the precision of the position estimation, as already mentioned in the previous paragraph. This is depicted in Fig. 24a, where the comparison of the state-of-the-art approach with the proposed Channel-SLAM approach is illustrated using the RMSE. The solid and dotted lines illustrate the first and the second scenarios, respectively. The RMSE of the proposed algorithm never exceeds 1 m

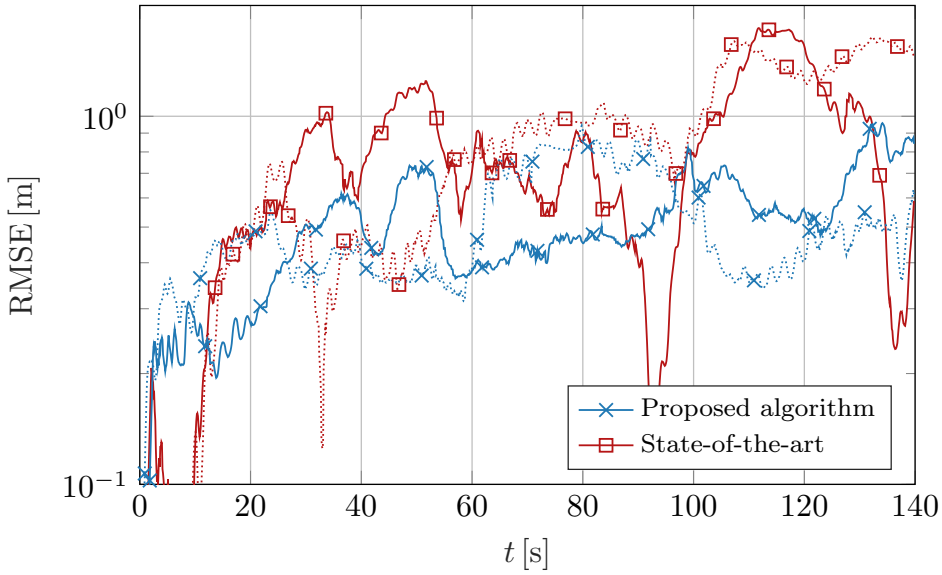


(a) The evaluation of scenario number one.

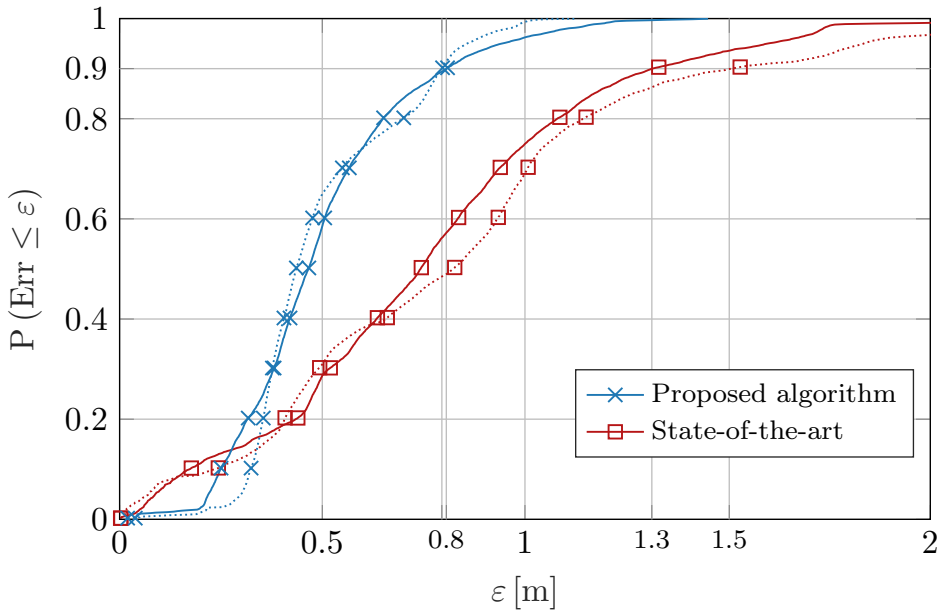


(b) The evaluation of scenario number two.

Figure 23: The layout of testing scenarios and the comparison of the performance of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The position estimators use estimated MPC parameters obtained from measured CIR using one *Decawave DWM1000* anchor placed at position VT$_1$. The transition models use IMU measurements.



(a) The RMSE comparison throughout the experiment duration.



(b) The RMSE empirical CDF over the runs of the simulation. The additional grid lines mark the 90% confidence interval for each plot.

Figure 24: The comparison of the performance of the Channel-SLAM using the proposed stochastic data association algorithm and pedestrian transition model with the state-of-the-art Channel-SLAM without the stochastic data association and with the Rician transition model. The position estimators use estimated MPC parameters obtained from measured CIR. The solid and dotted lines illustrate the first and the second scenarios, respectively.

for both scenarios, unlike the state-of-the-art approach exceeding 1 m in both scenarios.

The empirical CDFs in Fig. 24b shows the overall comparison of the proposed Channel-SLAM with the state-of-the-art approach. The precision of the proposed algorithm outperforms the state-of-the-art approach in each experiment.

The additional grid lines mark the 90% confidence interval of the RMSE for each plot. For the first scenario, the 90% confidence interval of the proposed algorithm's RMSE is ±0.81 m, while for state-of-the-art, it is ±1.31 m. For the second scenario, the 90% confidence interval of the proposed algorithm's RMSE is ±0.80 m, while for state-of-the-art, it is ±1.50 m.

The average duration of one run for the first scenario is 54.5 min for the proposed and 48.1 min for the state-of-the-art algorithm. For the second scenario, 59.2 min and 53.9 min for the proposed and the state-of-the-art algorithm, respectively.

The comparison of the proposed Channel-SLAM with the state-of-the-art approach shows that the data association capability proposed in this work yields significantly better precision in the position estimation.

When the low-cost *Decawave DWM1000* anchor is used, the MPCs are observed for a lower percentage of the time, and the outage can last longer compared to the professional broadband channel sounder used in [Gen+17]. During the KEST outage, the variance of system position increases, and when the MPCs re-occur, the initialization process creates new subordinate particle filters to start estimating the VT again. However, the variance of the system position is reflected in the variance of the VT via the initialization process. Thus, re-occurring VTs cannot improve the position estimation when the state-of-the-art approach is used for positioning. On the other hand, if the data association is performed and some of the system position particles are associated with a previously observed VT, the variance of the position estimation decreases immediately. This process can recover the precision after an MPC outage.

Chapter 5

Gaussian Mixture Model for Virtual Transmitter Estimation

This chapter aims to improve the speed and scalability of the state-of-the-art Channel-SLAM positioning algorithm [KG20]. The Channel-SLAM is based on a RBPF [Dou+00]. Its computational complexity depends linearly on the number of particles. I propose a method that allows decreasing the required number of particles significantly.

I present a novel idea to represent the VTs by a GMM rather than by a PF [Aru+02]. The proposed method learns the GMM parameters online. The efficient learning schedule allows initializing VTs using orderly less GMM components than a PF would require. Moreover, I propose a learning schedule that efficiently releases and reuses computational resources, which further improves the speed of the Channel-SLAM.

This chapter describes the findings I published in [KG21]. Hence, I use citations from this publication throughout this chapter.

5.1 Modeling Virtual Transmitter with a Gaussian Mixture Model

Since the VT posterior PDF is nongaussian and, in most cases, a multimodal, its exact evaluation is impossible. Approximative methods provide a solution to this problem. I propose a GMM-based approximation of the VT posterior PDF in this work. Generally, the GMM allows a tractable approximation of an arbitrary PDF [AS72]. The calculation power and available memory limit the maximum possible GMM order, limiting the PDF approximation precision.

The system state at time step t is obtained from (4.1) as

$$\mathbf{x}_{\mathrm{u},t} = \left[\mathbf{x}_{\mathrm{Rx},t}^{\mathrm{T}}, \boldsymbol{\theta}_t^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{5.1}$$

The state of ℓ-th VT at time step t is obtained from (4.2) as

$$\mathbf{x}_{\mathrm{v},\ell t} = \left[\mathbf{x}_{\mathrm{Tx},\ell t}^{\mathrm{T}}, b_{\mathrm{v},\ell t}\right]^{\mathrm{T}}. \tag{5.2}$$

The GMM approximating the VT's PDF is defined as

$$\mathrm{p}(\mathbf{x}_{\mathrm{v},\ell t}) \approx \sum_{p=1}^{P} w_{\ell t|t-1}^{(p)} \mathcal{N}\left(\mathbf{x}_{\mathrm{v},\ell t}^{(p)}, \Sigma_{\ell t}\right), \tag{5.3}$$

where each mixture component is defined by a Gaussian distribution with a mean $\mathbf{x}_{\mathrm{v},\ell t}^{(p)}$, covariance matrix $\Sigma_{\ell t}$, and mixture weight $w_{\ell t|t-1}^{(p)}$.

The GMM-based VT approximation is a two-step process. In the beginning, the VT's prior PDF is approximated by the GMM. The algorithm uses each new VT observation z_{it} to update the GMM, estimating the posterior PDF.

A simple method to approximate the VT is to draw a set of samples from its prior PDF and use it as the mean values of the GMM components. The covariance matrix of each component is the same and is selected according to desired precision of the approximation. However, the smaller the covariance matrix, the larger the GMM order must be.

Before defining the VT's prior PDF, a short discussion is necessary. Prior the first VT observation, a uniform distribution inside a sphere centered at $\mathbf{x}_{\mathrm{Rx},t}$ would define the prior VT PDF. Since its radius is infinite, it is impossible to approximate such PDF by sampling a finite number of components from it. The transmitter's range reduces this radius. However, the required number of components is cubic in the range defining the prior PDF sphere diameter. Moreover, most of the prior PDF sphere volume would have almost zero likelihood after the first observation. The calculation load of such an approach is infeasible even for today's computers. An alternative method used in this work leverages the fact that the posterior PDF after the first observation has a closed-form solution.

Looking at the observation likelihood

$$\mathrm{p}(z_{it} \mid \mathbf{x}_{\mathrm{u},t}, \mathbf{x}_{\mathrm{v},\ell t}) = \frac{1}{\sqrt{2\pi Q}} \mathrm{e}^{-\frac{\left(z_{it} - \left\|\mathbf{x}_{\mathrm{Rx},t} - \mathbf{x}_{\mathrm{Tx},\ell t}\right\| - b_{\mathrm{v},\ell t}\right)^2}{2Q}}, \tag{5.4}$$

one can notice that a right circular cone defines the equiprobable plane. The right circular cone is given by

$$\left\|\mathbf{x}_{\mathrm{Rx},t} - \mathbf{x}_{\mathrm{Tx},\ell t}\right\|^2 = \left(z_{it} - b_{\mathrm{v},\ell t}\right)^2. \tag{5.5}$$

Assuming the VT's prior PDF as a sphere with a positive radius centered at $\mathbf{x}_{\mathrm{Rx},t}$ gives posterior PDF in the same format as (5.4) but constraining $b_{\mathrm{v},\ell t}$ to be nonnegative. Also, the posterior PDF has a different normalization constant than (5.4).

This observation motivates an efficient VT initialization that directly draws samples from the posterior PDF obtained after the first observation. In detail, the algorithm

draws the required number of samples of $\mathbf{x}_{\mathrm{Tx},\ell t}$ uniformly inside a circle centered at $\mathbf{x}_{\mathrm{Rx},t}$. The first observation defines the circle radius. Since the observation is assumed to be a gaussian distributed random variable

$$z_{i\,t} = \mathcal{N}\left(\mu_{z_{i\,t}}, Q\right) \tag{5.6}$$

with mean $\mu_{z_{i\,t}}$ and variance Q, it is sufficient to draw samples inside a circle with a radius $z_{i\,t} + 5\sqrt{Q}$ to ensure the correct VT position is inside with a probability higher than 99.9999%. Finally, each position sample is appended with the additional propagation distance given by

$$b_{\mathrm{v},\ell\,t} = z_{i\,t} - \|\mathbf{x}_{\mathrm{Rx},t} - \mathbf{x}_{\mathrm{Tx},\ell\,t}\|, \tag{5.7}$$

where for p-th sample of VT position $\mathbf{x}^{(p)}_{\mathrm{Tx},\ell\,t}$ the corresponding sample of the additional propagation distance $b^{(p)}_{\mathrm{v},\ell\,t}$ is obtained using random values of $z_{i\,t}$ drawn from (5.6). This method provides samples of $\mathbf{x}_{\mathrm{v},\ell\,t}$ distributed according to the posterior PDF calculated after the first observation.

Up to this point, the VT approximation is similar to the state-of-the-art approach [KG20]. The main difference is that the Regularized Particle Filter (RPF) [Aru+02] used in [KG20] introduces a Brownian movement to VT particles during the resampling of a PF. Since the state-of-the-art Channel-SLAM requires the stationary VTs, the Brownian movement violates this condition. Setting the RPF kernel sufficiently small that the Brownian movement is negligible compared to the $\mathbf{x}_{\mathrm{Rx},t}$ movement solves this problem. However, the number of required RPF particles grows with $1/\sigma^2$, where σ^2 is the variance of the Brownian movement. Hence, a compromise between the number of particles and the validity of the approximation must be made. In general, the number of required particles is significant and does not allow real-time processing.

On the other hand, the GMM-based approach assumes stationary particles precisely according to the Channel-SLAM derivation. The mean values are sampled as described above. The weight of components is initialized as

$$w^{(p)}_{\ell\,t|t-1} = \frac{1}{P}. \tag{5.8}$$

The initial position covariance $\Sigma_{\ell\,t}$ of each component is constant and is proportional to the sample estimate of the position covariance matrix

$$\hat{\Sigma}_{\mathbf{x}_{\mathrm{Tx},\ell\,t}} = \sum_p w^{(p)}_{\ell\,t|t-1} \left(\mathbf{x}^{(p)}_{\mathrm{Tx},\ell\,t} - \hat{\mu}_{\mathbf{x}_{\mathrm{Tx},\ell\,t}}\right) \left(\mathbf{x}^{(p)}_{\mathrm{Tx},\ell\,t} - \hat{\mu}_{\mathbf{x}_{\mathrm{Tx},\ell\,t}}\right)^{\mathrm{T}}, \tag{5.9}$$

where the estimated mean is given by

$$\hat{\mu}_{\mathbf{x}_{\mathrm{Tx},\ell\,t}} = \sum_p w^{(p)}_{\ell\,t|t-1} \mathbf{x}^{(p)}_{\mathrm{Tx},\ell\,t}. \tag{5.10}$$

The variance of the additional propagation distance $\sigma^2_{b_{\mathrm{v},\ell\,t}}$ is set to zero to reflect the stationarity of VTs.

The Channel-SLAM aims mainly for position estimation using the VT position information. Keeping this goal in mind, after initializing a new VT, it provides minimum information for the position estimation. After more information about the VT's state is collected, the VT posterior PDF starts to support the position estimation. From the positioning point of view, it does not matter if a newly initialized VT's lack of information is encoded by a large number of particles approximating an almost uniform distribution or by a low order GMM with large covariance of its components. However, from a computational complexity point of view, the GMM can be orderly faster since the number of components is small compared to the RPF.

Rather than tracking the posterior PDF as precisely as possible using RPF, the GMM starts with a coarser approximation of the posterior PDF and learns the posterior PDF by sequentially incorporating new observations. The GMM sequential learning process is related to RPF in a sense it calculates posterior weights and performs resampling. However, it uses an exploration technique in the resampling step to learn the posterior PDF sequentially without introducing a Brownian movement to its components.

5.2 Gaussian Mixture Model Resampling

Like [KG20], the systematic resampling scheme [Aru+02] provides a set of new GMM components by drawing the components with the probability corresponding to their weights $w_{\ell t|t-1}^{(p)}$. However, the components are not drawn multiple times during the systematic resampling step. The proposed algorithm draws each component not more than once and assigns it a weight corresponding to the required occurrences. Hence, the PDF obtained by this resampling is equivalent to that obtained by systematic resampling but has fewer components.

The key idea is to reuse the resources released by resampling without repetition for exploration to improve the posterior PDF approximation and reduce the PF's degeneracy phenomenon. Sampling from the GMM with P_{unq} unique components obtained during the resampling without repetition provides P_{add} exploration samples. Assigning the P_{add} samples with a constant weight, appending with P_{unq} components, and normalizing the new set of $P = P_{\text{add}} + P_{\text{unq}}$ weights produces a new GMM approximating the VT posterior PDF.

Setting the p-th GMM component position covariance to (5.9) results in a covariance decreasing as the GMM becomes more sure about the VT position. However, setting the $b_{\text{v},\ell t}^{(p)}$ variance proportional to its sample-based estimate would produce an overly pessimistic model. Rather than that, one can use (5.7) to define the nonlinear model between the VT position and the $b_{\text{v},\ell t}^{(p)}$. In other words, the $b_{\text{v},\ell t}^{(p)}$ and $\mathbf{x}_{\text{Tx},\ell t}^{(p)}$ should reflect that the $\mathbf{x}_{\text{v},\ell t}^{(p)}$ is always close to the surface of the right cone that defines VT's prior PDF as in Section 5.1. Following the VT stationarity given by the Channel-SLAM derivation, the variance $b_{\text{v},\ell t}^{(p)}$ is set to zero.

The P_{add} exploration samples are sampled from the GMM while assuring the exploration samples would be in the vicinity of the VT's prior PDF. The sampling from a GMM consists of three steps. First, a p-th GMM component $\mathbf{x}_{\text{Tx},\ell t}^{(p)}$ is selected with prob-

ability $w^{(p)}_{\ell\, t|t-1}$. Then, one sample $\mathbf{x}^{(new)}_{\mathrm{v},\ell\, t}$ is drawn from a gaussian distribution defined by the parameters of the p-th GMM component. Finally, the $b^{(new)}_{\mathrm{v},\ell\, t}$ value is updated according to

$$b^{(new)}_{\mathrm{v},\ell\, t} = b^{(p)}_{\mathrm{v},\ell\, t} + \left\| \mathbf{x}_{\mathrm{Rx},t} - \mathbf{x}^{(p)}_{\mathrm{Tx},\ell\, t} \right\| - \left\| \mathbf{x}_{\mathrm{Rx},t} - \mathbf{x}^{(new)}_{\mathrm{Tx},\ell\, t} \right\|. \tag{5.11}$$

To demonstrate, sampling a $\mathbf{x}^{(new)}_{\mathrm{v},\ell\, t}$ where, $\sigma^2_{b^{(new)}_{\mathrm{v},\ell\, t}} = 0$ shifts $\mathbf{x}^{(p)}_{\mathrm{v},\ell\, t}$ in a plane of constant $b^{(p)}_{\mathrm{v},\ell\, t}$, which might no longer be a point on the prior PDF right cone's surface given by (5.5) at the time step of the first VT observation. (5.11) shifts $\mathbf{x}^{(new)}_{\mathrm{v},\ell\, t}$ in the $b_{\mathrm{v},\ell}$ direction back on the surface of the prior PDF.

The presented exploration schedule provides new GMM components, which are both close to a current posterior PDF's areas of high probability and near the surface of the right cone, defining the VT's prior PDF from which the initial set of GMM components was sampled.

Thanks to this approach, the new VT can be initialized with a significantly subsampled prior PDF and still find the correct VT position by exploring the prior PDF during the positioning. A disadvantage of this approach is that the new components do not carry information from the previous measurements. Hence, it is not a statistically optimal approach. The optimal approach requires evaluating the likelihood of this new component for all previous measurements that present a high computational load and requires storing the whole measurement history.

The last observation concerns the rate of exploration. The exploration step is done only when resampling is required. When only a handful of components are present after resampling, which manifests the degenerative behavior and poor posterior PDF approximation, the number of released resources for exploration is large. Hence the poorer the posterior PDF representation, the stronger the exploration. The algorithm automatically determines how strong the exploration should be. Interestingly, it follows the intuition that more exploration is necessary when the VT posterior PDF poorly reflects the actual posterior PDF.

5.3 Performance of the Gaussian Mixture Model

To show the advantage of the presented method, I compare it with the method [KG20], which is referenced as a state-of-the-art approach throughout this chapter. I use simulations of a band-limited multipath radio channel. Each simulation assumes one physical transmitter and five reflections. The WNA movement model [BLK01] simulates the system trajectory. I assume that a channel parameter estimator approaches the CRLB when processing the received ranging waveform. The noisy delay estimates serve as an input for the algorithms. Finally, I assume outages of the MPC tracking. The MPC outage occurs when the amplitude of the MPC decreases below some threshold. I use 3 dB above the noise level as a threshold.

The evaluation compares the proposed algorithm with the state-of-the-art approach for different particle volumes initializing the VTs. The simulations are performed for ten

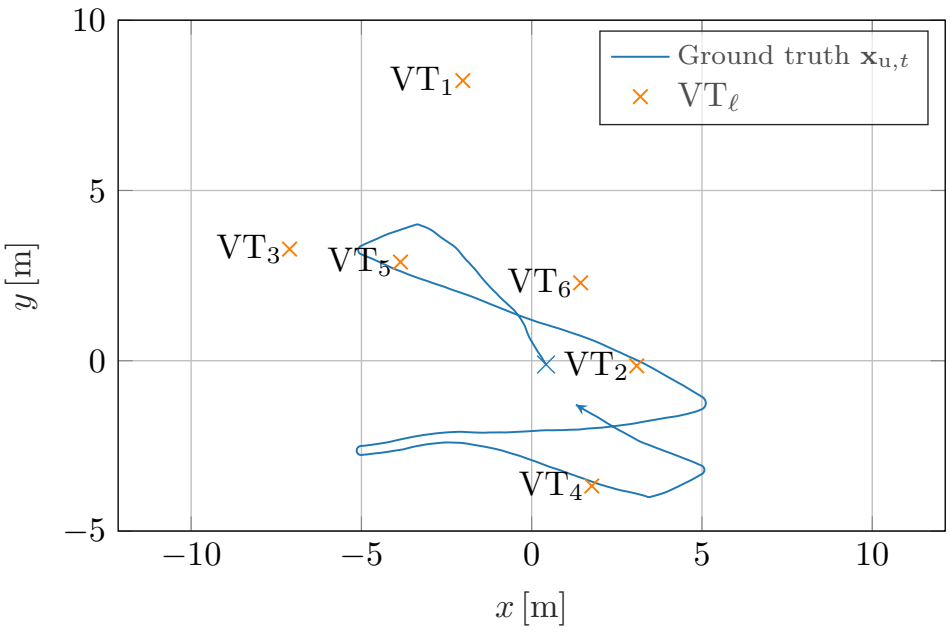


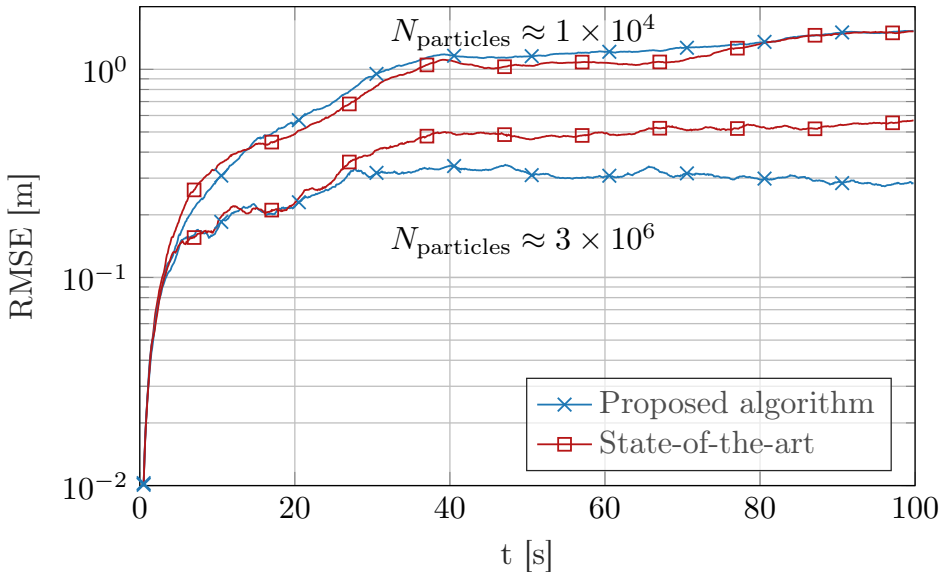
Figure 25: The example of the simulated trajectory is shown in green with the marked beginning and the end. The randomly generated VTs are marked with blue crosses, where the $\mathbf{x}_{\mathrm{Tx},1}$ is the physical transmitter. The figure shows the RMSE curves for two different particle volumes.

randomly generated scenarios. An example of a randomly generated scenario is shown in Fig. 25, where the pedestrian walks inside a rectangular room for 100 seconds. All evaluations are repeated ten times to capture the simulation and evaluation randomness.

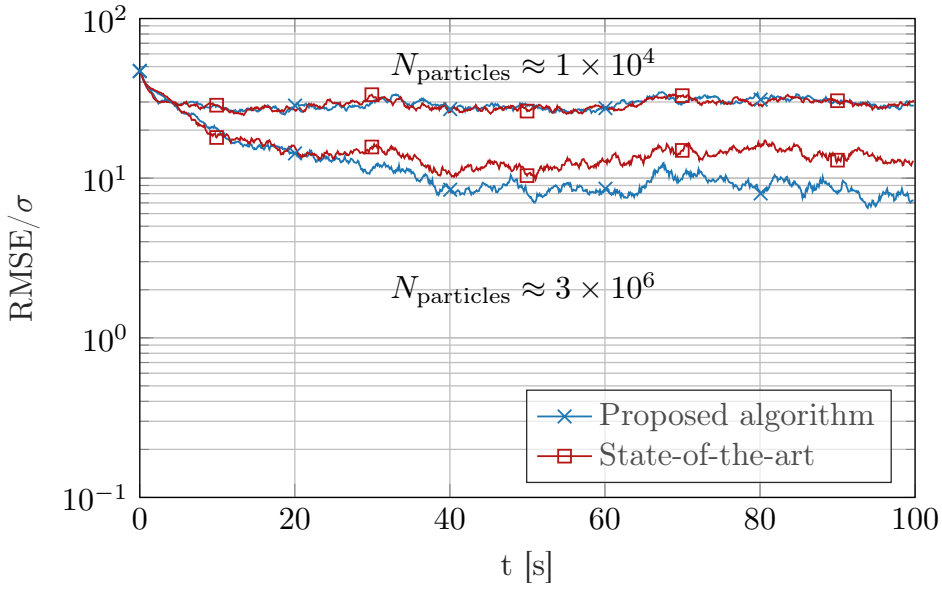
Throughout the evaluations, the RMSE is used to compare the algorithm performance. The delay estimation variances differ significantly since each VT has a different propagation loss The evaluated RMSE for each VT is normalized by its delay STD to show the VT estimator's performance. The system and VT position estimation performance over time is shown in Fig. 26a and Fig. 26b. The plots show two different volumes of simulated particles. The particle volume size is defined by the particles needed to approximate the system state; however, it depends on the MPC delay and variance during the initialization. Hence, the number of particles varies between $1 \times 10^4 < N_{\text{particles}} < 5 \times 10^6$. The different volumes of particles for individual simulations are obtained by changing one multiplicative constant, influencing the number of particles used to initialize VT.

I assume that the initial system state is known. Hence, the initial RMSE in Fig. 26a starts at zero and slowly increases because the VTs are not yet estimated well enough to correct the system state prediction error. The VTs are initialized uniformly within the whole prior PDF given by (5.5). Hence, in Fig. 26b, the VT RMSE is maximal initially and decreases over time.

The plotted CDFs in Fig. 27 show better the performance improvement. The additional dashed plot shows the performance when the number of VT particles is increased ten times than the minimum $N_{\text{particles}} \approx 1 \times 10^4$. While the state-of-the-art method still



(a)



(b)

Figure 26: The RMSE of the system $\mathbf{x}_{\text{Rx},t}$ (a) and VT position (b) was estimated using the proposed PF resampling method marked by a blue line and the state-of-the-art Channel-SLAM shown by a red line. The corresponding STD normalizes each VT allowing one to plot one generalizing RMSE curve for the VTs with different observation covariance. The figures show the RMSE curves for two different particle volumes.

needs more particles to approach its potential, the proposed algorithm already outperforms the best precision achievable by the state-of-the-art algorithm.

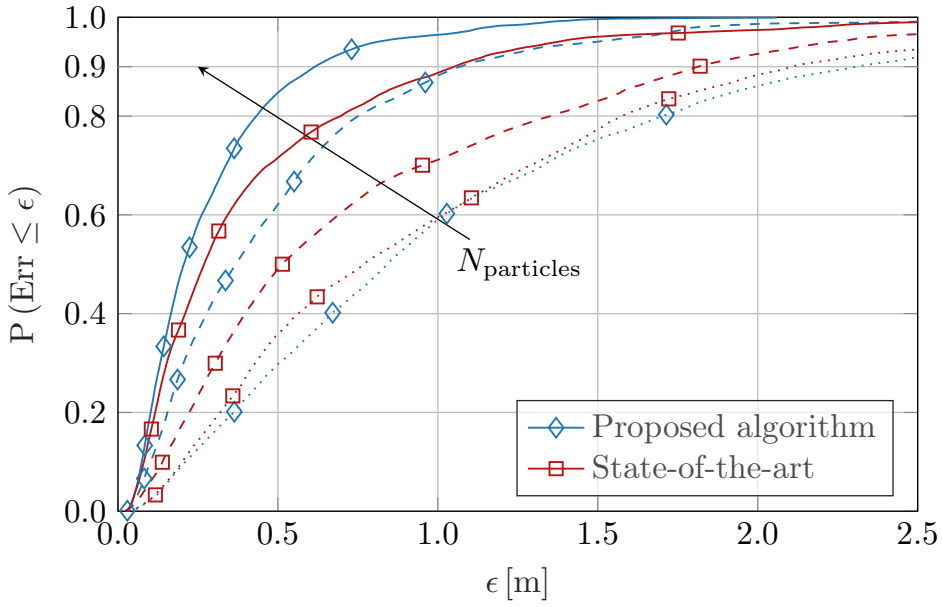
To fully capture the sensitivity of the compared algorithms concerning the number of particles, Fig. 28 shows the 90% confidence interval of the RMSE achievable for a given number of particles. Fig. 28b shows that for an extremely low number of VT particles, the RMSE converges to a common point. The lowest point is using approximately ten particles per VT per system particle.

With the number of particles this low, the probability of converging to a correct VT position is marginal. However, the measurements help estimate the system position even for the lowest number of particles. When tracking the system position using only the movement model, the RMSE is more than twice the RMSE $< 2.5\,\mathrm{m}$ achieved using VT with the lowest number of particles.

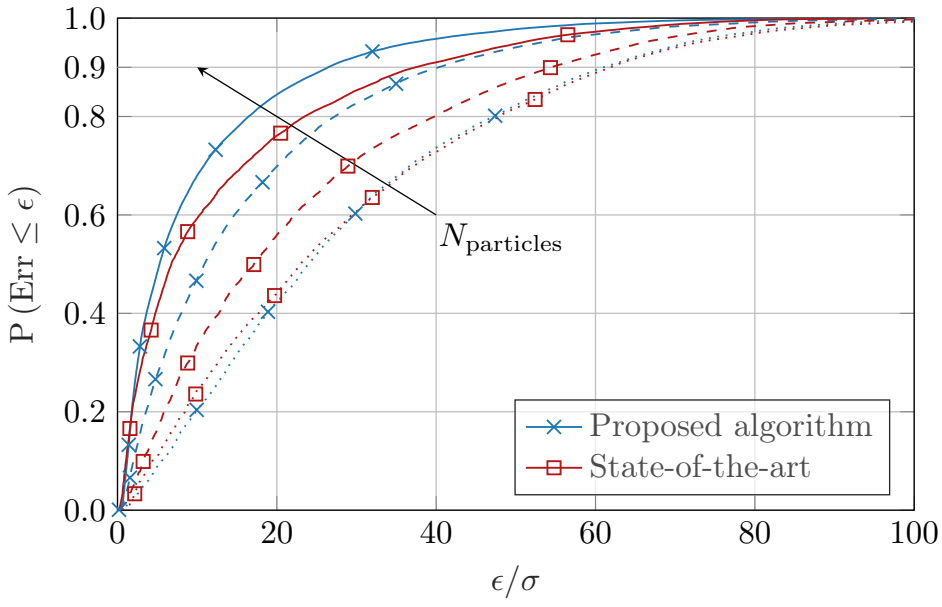
With the increasing number of particles, the RMSE decreases faster for the proposed algorithm than in the state-of-the-art approach due to reusing redeemed resources during the resampling step to explore the areas where the likelihood is assumed to be significant. The proposed algorithm allows obtaining the assumed maximal achievable precision for orderly fewer particles. The scenario, along with the VT placement and the system trajectory, limits the precision, making calculating the true maximal achievable precision difficult. The maximal achievable precision is when the RMSE is no longer improving with an increasing number of particles.

Interestingly, the proposed method can achieve better precision even for a large volume of particles, where both algorithms would be expected to converge to the same precision. This might be caused by the fact that the VT in the state-of-the-art approach performs a weak Brownian movement during the kernel resampling step. In contrast, the GMM-based method assumes static VTs as required by the Channel-SLAM and performs exploration to approximate the posterior PDF using a low volume of particles.

The better the VT state estimate accuracy, the better the system state accuracy. Hence, the VT precision affects the RMSE of the system state estimation depicted in Fig. 28a similarly.

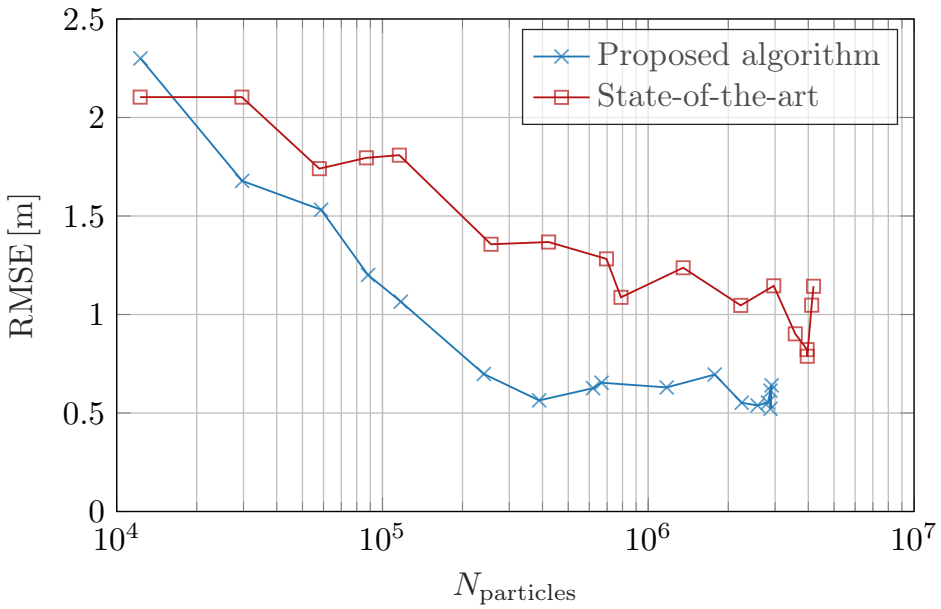


(a)

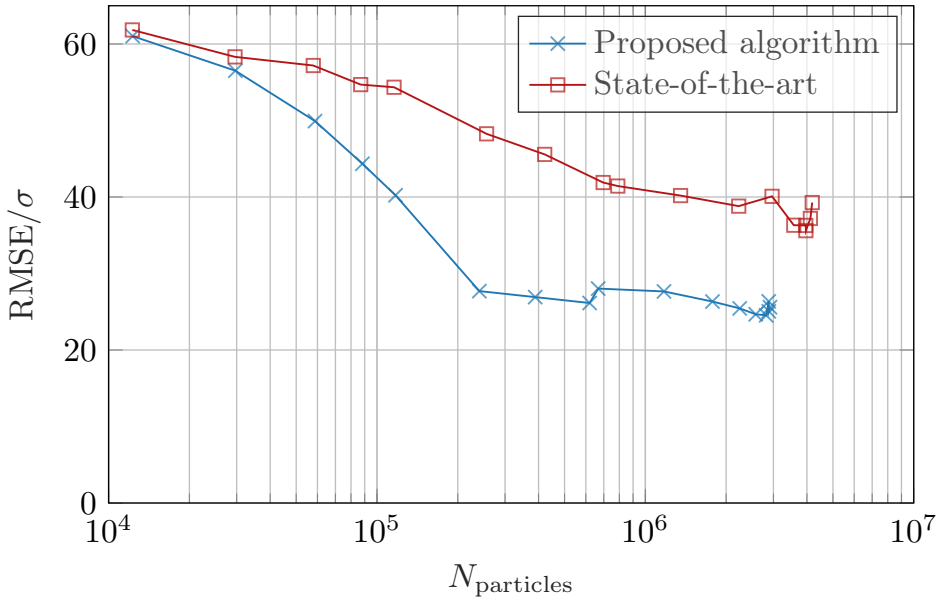


(b)

Figure 27: The CDF of the system $\mathbf{x}_{\mathrm{Rx},t}$ (a) and VT position (b) RMSE was estimated using the proposed PF resampling method marked by a blue line and the state-of-the-art algorithm shown by a red line. The corresponding STD σ normalizes each VT error to be able to plot one generalizing RMSE curve for the VTs with different observation covariance. The figures show the RMSE curves for three different particle volumes.



(a)



(b)

Figure 28: The comparison of the proposed and the state-of-the-art PF resampling method RMSE achieved for a given number of particles. The 90% confidence intervals of RMSE for a system (a) and VT (b) position are plotted.

Chapter 6

Channel Parameter Estimation Using Artificial Neural Network

This chapter studies the possibility of using an ANN to estimate the parameters of the multipath radio channel. The experiments using ANN for radio channel estimation are a recent addition to the growing ecosystem of ANN applications.

One of the first experiments shows the CNN application for AoA estimation in acoustics systems [CH17]. This method is a dictionary-based approach, where one AoA is extracted as an index of the output layer maximum, limiting the achievable precision and number of possible signal sources. The latter approach in [APV18] solves the single AoA limitation by using ANN to preprocess the received signal. The resulting pseudo-spectrum is processed by the MUSIC algorithm to obtain multiple AoAs. Since the number of ANN outputs does not limit the MUSIC algorithm resolution, the presented method is no longer a dictionary-based approach.

Similar ideas are used in the following research aiming for the line spectra estimation in [IBF19; IMF19; JLR19]. An ANN preprocesses the received signal, and the output is then processed by classical frequency estimation algorithms like MUSIC or root-MUSIC.

For the ANN, it is a challenging problem to directly estimate desired radio channel parameters like AoA, ToA, TDoA, or Doppler frequency. This approach was briefly studied in [IBF19], where some open problems were highlighted, e.g., how to output a variable number of radio channel parameters with a fixed ANN architecture and how to compare this set of outputted parameters with the ground truth to obtain a loss function for the ANN training. The ANN-based direct estimation of the LoS component delay in the multipath radio channel is proposed in [HYK21].

The previous work lack comparison of achievable performance with the theoretical limit given by CRLB. Only [JLR19] compares the performance with the CRLB, but it uses ANN to preprocess the signal, and the frequency is estimated using the MUSIC algorithm.

This chapter provides a novel idea of MPC parameter estimation using convolutional ANN architecture trained with a novel multi-loss function. The achieved results are compared with the theoretical limits and state-of-the-art approaches to channel parameter estimation.

The main contributions presented in this chapter are:

1. I compare the ANN-based estimator precision with the theoretical limit given by the CRLB and with the benchmark algorithms approaching this theoretical limit.

2. I propose and evaluate a method to solve the problem of unknown model order. The ANN has a fixed number of outputs, but the model order, or the number of delays present in the signal, can be different for each radio channel measurement. The unknown model order problem is overcome by appending the outputted delay estimations with a set of weights. The weights are trained to reflect the probability that the corresponding delays are active or inactive.

3. I introduce a novel loss function for the ANN training. It was found that the ANN using the delay Mean-Square Error (MSE) loss function for training struggles to approach the theoretical limit given by the CRLB. The proposed combination of the delay MSE, binary cross-entropy and a loss function based on the ML radio channel estimation technique significantly improves the ANN model performance. It even approaches the CRLB for a radio channel with a single MPC.

4. I propose a noise regularization technique, which adds noise to the input dataset with the same variance as the noise present in the radio channel. This technique is comparable to using an infinite size training dataset, and it shows that it successfully suppresses the ANN overfitting problem.

5. Finally, I propose three different methods for the data association of the ANN outputs with the true values, which is vital for ANN training.

   The first data association method uses an ordered set of true values. Hence, the ANN needs to learn to output the delays ordered as well.

   The second approach implements a suboptimal but fast data association algorithm directly into the ANN loss function. Therefore, the ANN does not need to learn to output the delays in a specific order.

   Finally, the third approach is the optimal Munkres data association algorithm implemented directly in the ANN loss function. The evaluation shows that removing the task of correct ordering from the ANN and implementing it directly to the ANN loss function improves the precision of the trained models. Moreover, it has a significant impact when the model order is unknown.

6.1 Dataset Generator

This section describes the dataset generator that generates examples of the multipath radio channel observations and the corresponding MPC parameters describing the mul-

Table 6.1: List of all configurable parameters of the dataset generator

nMPC	number of MPCs
nSa	number of CIR samples
nData	size of the whole dataset
SNRmin	minimal SNR of MPC in dB
SNRmax	maximal SNR of MPC in dB
rndVis	random number of visible MPCs
minSep	minimal separation between MPC delays

tipath radio channel. The generated dataset serves to train an ANN architecture to estimate the multipath radio channel parameters from a noisy radio channel observation.

The number of MPCs is selected before generating the training dataset by the *nMPC* parameter. However, it is possible to have a random number of MPCs in each generated measurement by switching on the *rndVis* parameter. Then, the *nMPC* serves as the maximal possible number of MPCs. The actual number of MPCs is selected randomly and uniformly between 1 and *nMPC*.

The SNR of each MPC is randomly and uniformly selected between *SNRmin* and *SNRmax*. Note that the SNR is sampled uniformly in the dB scale; hence, the SNR distribution in the linear scale follows the log-uniform distribution.

The *minSep* parameter gives the minimal separation between two MPCs. This parameter is important to control or avoid the problem of two closely spaced sources [VBC14]. The delay value sorts the true MPC parameters in the dataset in ascending order. The set of all configurable parameters in the dataset generator is given in Table 6.1.

The measurements are generated directly from the multipath propagation model (1.9) in the frequency domain as

$$\boldsymbol{S} = \boldsymbol{B}\left(\boldsymbol{\tau}\right)\boldsymbol{\alpha} + \boldsymbol{n} = e^{-j2\pi\boldsymbol{F}\boldsymbol{\tau}^{\mathrm{T}}}\left(\|\boldsymbol{\alpha}\| \odot e^{-j\angle\boldsymbol{\alpha}}\right) + \boldsymbol{n}, \tag{6.1}$$

where $\boldsymbol{F} = [0, \ldots, m, \ldots, M-1]^{\mathrm{T}}$, delays $\boldsymbol{\tau}$ (1.7) are randomly sampled from a uniform distribution on the interval $[0, 1)$ but assured that no pair of delays is closer than *minSep*, MPC magnitudes $\|\boldsymbol{\alpha}\|$ (1.8) are obtained from randomly sampled SNR, $\odot$ is Hadamard product, MPC phases $\angle\boldsymbol{\alpha}$ are sampled from a uniform distribution on the interval $[0, 2\pi)$, and $\boldsymbol{n}$ is multivariate normally distributed noise

$$\boldsymbol{n} \sim \mathcal{N}\left(0, 2\pi^2\boldsymbol{I}\right) \tag{6.2}$$

with identity matrix $\boldsymbol{I}$ and $\mathrm{rank}(\boldsymbol{I}) = M$. Finally, the generated measurements representing the CIR are obtained from (6.1) using the IFFT. The real and imaginary parts of the CIR are appended before being inputted to the ANN.

6.2 Artificial Neural Network for Delay Estimation

The architecture and training of the ANN consist of many hyperparameters having a massive impact on the achievable precision of the estimation. In this section, I describe all the aspects taken into account when designing the ANN architecture and configuring the training schedule of the proposed ANN. I use the TensorFlow library [Aba+15] to design the ANN architecture in an organized manner and leverage the automatic differentiation that simplifies the use of custom ANN loss functions. The aim is to find an ANN architecture with minimal design choices and hyperparameters to tune while assuring robust convergence properties.

The proposed ANN architecture consists of a Gaussian noise layer, normalization layer, four convolutional layers, and two fully connected layers. All convolutional layers and the second last fully connected layer use the ReLU activation function. The last fully connected layer has twice as many outputs as the maximum expected number of MPCs. The first *nMPC* outputs represent estimated delays $\hat{\boldsymbol{\tau}}$ and use the ReLU activation function. The remaining *nMPC* outputs represent estimated weights $\hat{\boldsymbol{w}}$ and use a logistic sigmoid activation function

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{6.3}$$

The logistic sigmoid activation function assures that each weight in $\hat{\boldsymbol{w}}$ is scaled to the interval $(0, 1)$, where 0 indicates that the MPC is inactive and 1 indicates that the MPC is active. Thus, it can be used as a probability that the corresponding delay is active or inactive. The weights $\hat{\boldsymbol{w}}$ provide a simple method for coping with a changing number of MPCs while the ANN architecture is fixed.

When the ANN architecture is large enough, consisting of enough tunable parameters, it assures that it can learn an arbitrary function from the training data. On the other hand, the ANN architecture is flexible enough to memorize the training set perfectly, which leads to severe overfitting. The standard ANN architecture usually contains some regularization method [Bis06] to resist overfitting. However, the most common regularization methods like weight or activity regularization, early stopping, and dropout are hard to tune to reduce the overfitting but not to cause underfitting.

Experiments show that overfitting and underfitting is a severe problem that causes the ANN not to get closer to the CRLB. With a significant amount of activity regularization tuning, it was possible to achieve delay precision about ten times worse than CRLB for one MPC in noise. However, assuming a different number of MPCs, or any model change caused a significant drop in precision. Interestingly, also the batch size influences the achievable precision. The achievable precision decreased when the batch size was larger than 32, probably due to ANN converging to a local minimum. This behavior is undesirable since it is not possible to fully utilize the parallelization of the ANN using a Graphics Processing Unit (GPU).

I found that the noise regularization [Rot+19] is significantly more robust to small changes in the ANN architecture or changes of MPCs number. Therefore, only noise regularization is assumed in this work. Generally, the noise regularization method adds a

small amount of Gaussian noise to the input or output of an ANN layer. The TensorFlow layer *GaussianNoise* adds Gaussian noise with zero-mean and STD *stdIn* to the input data.

I use a trick for the ANN training where the inputted CIR is noise-free, but then the Gaussian noise layer adds noise with the variance set to achieve the desired SNR. Intuitively, it can be seen as generating an infinite size training dataset since the noise added to the noise-free CIR is different in each training epoch. Thanks to this, the ANN is not overfitting the training data. Moreover, this approach solves the batch size problem. Now, the batch size no longer influences the achievable precision with the noise regularization. Thanks to the noise regularization, the training process can fully utilize the GPU (*NVIDIA GeForce GTX 980 Ti*) using batch size $nBatch = 258$ without getting stuck in local optima.

The Gaussian noise layer is followed by the normalization layer, which normalizes all noisy CIRs to unit energy. This step is required. Otherwise, the model order might be estimated based on the CIR energy, which is undesirable. Normalization of the input of the ANN is always preferable, and the proposed normalization to the unit energy is a logical choice for the signal processing task.

The training uses Adam [KB17] optimizer with exponential decay defining the learning rate lr at each training *step* as

$$lr = lrInit \cdot lrRate^{step/lrStep}. \tag{6.4}$$

The ANN loss function for the delay estimation aims to minimize MSE error between the predicted and true delays

$$L\left(\boldsymbol{\tau}, \hat{\boldsymbol{\tau}}\right) = \left(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}\right)^{\mathrm{T}} \left(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}\right). \tag{6.5}$$

However, this straightforward error evaluation is not possible since any delay ordering in vector $\boldsymbol{\tau}$ yields the same CIR. This is a problem because, while $\boldsymbol{\tau}$ is an unordered set, the $\hat{\boldsymbol{\tau}}$ outputted from the ANN is always an ordered set. Moreover, the number of delays is unknown and possibly differs for each CIR, but the cardinality of the ANN output $|\hat{\boldsymbol{\tau}}|$ is fixed.

Setting the maximum number of output delays large enough can fix the cardinality problem. However, the possibility of inactivating some delays is needed to allow fewer delays than the number of ANN outputs. This is achieved by doubling the number of ANN outputs. Half of the outputs represent delays $\hat{\boldsymbol{\tau}}$, and the second half represents the activation probability or weight $\hat{\boldsymbol{w}}$ of each delay.

The training of an ANN architecture requires a loss function that needs to be minimized. The minimization of the loss function is usually implemented using an effective backpropagation algorithm [RHW86]. I propose and evaluate three possible solutions for comparing the predicted delays with the training dataset to obtain a loss function allowing reliable training of the proposed ANN.

The first and most straightforward solution is to sort the dataset delays and let the ANN learn to output the delays sorted. Then the delay MSE loss function is

$$L_2 = \left(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}\right)^{\mathrm{T}} \left(\left(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}}\right) \odot \boldsymbol{w}\right) + L_{\mathrm{bce}}\left(\boldsymbol{w}, \hat{\boldsymbol{w}}\right), \tag{6.6}$$

where the weights are trained using the binary cross-entropy function defined by

$$L_{\mathrm{bce}}\left(\boldsymbol{w},\hat{\boldsymbol{w}}\right)=-\frac{1}{|\boldsymbol{w}|}\left(\boldsymbol{w}^{\mathrm{T}}\log\left(\hat{\boldsymbol{w}}\right)+\left(1-\boldsymbol{w}\right)^{\mathrm{T}}\log\left(1-\hat{\boldsymbol{w}}\right)\right). \tag{6.7}$$

Notice that the true weight vector $\boldsymbol{w}$ is used to omit the influence of nonactive delays in the first member on the r.h.s. of (6.6). Thanks to this, the delay estimation is not dependent on the precision of weight estimation. Weight estimation is based on the second part of the loss function, the second member on the r.h.s. of (6.6). This loss function is called the ordered loss (*Ord*) throughout this work.

The following two proposed loss functions aim to remove the task of ordering the delays from the ANN and encode the unordered nature of delays directly to the loss function. The first proposal implements a suboptimal association between the members of $\boldsymbol{\tau}$ and $\hat{\boldsymbol{\tau}}$. It takes the first delay τ_1 from $\boldsymbol{\tau}$ and finds the best corresponding delay in $\hat{\boldsymbol{\tau}}$ as

$$j=\arg\min_{\ell}\left(\tau_1-\hat{\tau}_\ell\right)^2, \tag{6.8}$$

then $\hat{\tau}_j$ is removed from $\hat{\boldsymbol{\tau}}$, and the process is repeated for the following members of $\boldsymbol{\tau}$. This algorithm does not provide optimal associations between $\boldsymbol{\tau}$ and $\hat{\boldsymbol{\tau}}$, but it scales linearly with the model order $|\boldsymbol{\tau}|=nMPC$. Note that the same indexing is used to evaluate weight loss. The pseudocode is in Algorithm 1. However, the actual function can be implemented to apply over the whole batch, and $\left(\tau_\ell-\hat{\tau}_i\right)^2$ is evaluated only once. The provided pseudocode aims for readability and not optimality.

The last proposal is to perform an optimal assignment of estimated delays $\hat{\boldsymbol{\tau}}$ with the ground truth delays $\boldsymbol{\tau}$. Finding the optimal ordering of $\boldsymbol{\tau}$, so that (6.6) is minimized is a well-known assignment problem with exponential complexity if a naive approach is selected. This work uses the Hungarian algorithm or the Munkres assignment algorithm, which complexity is polynomial [Kuh55]. Specifically, I use a Python implementation that runs in $O\left(n^3\right)$ time. [1] The text refers to the Munkres assignment algorithm-based loss function, ordering $\boldsymbol{\tau}$ as the Munkres loss (*Munk*).

Comparing the proposed ANN architecture performance with the CRLB, I found that the ANN struggles to approach this theoretical limit even in a simple single MPC scenario with a constant SNR in the whole dataset. This leads us to propose a novel ANN loss function improving the learning convergence in the vicinity of the CRLB. The proposed loss function is inspired by the ML solution to channel parameter estimation in (1.15). This correlation loss is defined as

$$L_C=-\boldsymbol{s}^{\mathrm{H}}\boldsymbol{B}\left(\hat{\boldsymbol{\tau}}\right)\hat{\boldsymbol{\alpha}}, \tag{6.9}$$

where the amplitudes are estimated using BLUE

$$\hat{\boldsymbol{\alpha}}=\left(\boldsymbol{B}^{\mathrm{H}}\left(\hat{\boldsymbol{\tau}}\right)\boldsymbol{B}\left(\hat{\boldsymbol{\tau}}\right)\right)^{-1}\boldsymbol{B}^{\mathrm{H}}\left(\hat{\boldsymbol{\tau}}\right)\boldsymbol{s}. \tag{6.10}$$

There are two application notes regarding the proposed correlation loss function. In some cases, the Tensorflow cannot handle complex numbers while using GPU. Hence, all

[1] available: http://software.clapper.org/munkres/

Algorithm 1: Adjacent loss (Adj)

Input: $\boldsymbol{\tau}, \boldsymbol{w}, \hat{\boldsymbol{\tau}}, \hat{\boldsymbol{w}}$
Output: L
$L_\tau = 0, L_w = 0$
for $i = 1 : nMPC$
  $j = \arg\min_\ell (\tau_i - \hat{\tau}_\ell)^2$
  $L_\tau = L_\tau + w_i (\tau_i - \hat{\tau}_j)^2$
  $L_w = L_w + L_{\text{bce}}(w_i, \hat{w}_j)$
  $\hat{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}.\text{remove}(\ell)$ `// remove` ℓ`-th element`
  $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}.\text{remove}(\ell)$
$L = L_\tau + L_w / nMPC$

complex-valued variables have to be implemented using real numbers. E.g., the Moore-Penrose pseudo-inverse in (6.10) is implemented using two real-valued Moore-Penrose pseudo-inverses as [Fal07]. Also, to improve the precision, I use the Numpy implementation of the Singular Value Decomposition (SVD) in a double-precision floating-point rather than the Tensorflow implementation. Finally, the standard Moore-Penrose pseudo-inverse uses a cutoff parameter to omit weak eigenvalues obtained by the SVD. However, for correct gradient backpropagation, the weak eigenvalues are substituted by Inf rather than zero.

Interestingly, the correlation loss function can be viewed as unsupervised learning since it does not need the ground truth parameters. On the other hand, it cannot be used alone for the training. If the error between the true and estimated delay is larger than the 0.5 sampling period, the measured signal $\boldsymbol{s}$ is weakly correlated with the estimated replica. Moreover, the first derivative of the correlation function is not guaranteed to point towards the true delay for delay error larger than the 0.5 sampling period of the measured signal. Hence, the ANN cannot learn to estimate delay using only (6.9).

The use of correlation loss for ANN-based channel parameter estimation is a promising novel idea. I train the proposed ANN architecture using a total loss

$$L = L_2 + lossW \cdot L_C. \tag{6.11}$$

Combining one of the proposed loss functions with the ordering of the $\hat{\boldsymbol{\tau}}$ and correlation loss shows superior performance that was impossible to achieve using only (6.6).

6.3 Achieved Results

The main results are comparing the ANN-based channel parameter estimation method with the theoretical limits given by CRLB and with the ML estimation method SAGE [Fle+99] and the SLSE [HFR18]. The comparison is made in three stages with an increasing level of generalization.

Table 6.2: List of all configurable parameters of the ANN training schedule

nBatch	batch size
stdIn	STD of input regularization noise
lrInit	initial learning rate
lrStep	learning rate decay step
lrRate	learning rate decay rate
nEpochs	number of training epochs
lossFnc	training loss functions
lossW	weight of correlation loss function
valSplit	fraction of dataset used as validation dataset

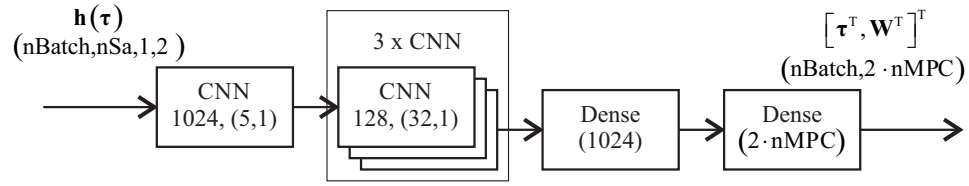


Figure 29: The core of the ANN architecture, including dimensions, number of filters, kernel sizes, and number of units of individual layers. The normalized noisy CIR is separated into the real and imaginary parts along the last axis. Then, it is inputted to the first convolutional layer as $\boldsymbol{h}(\boldsymbol{\tau})$.

In the first stage, the performance is studied while estimating the delay of a single MPC in noise. In the second stage, the complexity is increased by adding an MPC to the CIR. The final stage is adding the last generalization step, the unknown number of MPCs in noise. According to Section 6.1, all MPCs are generated with a random delay, magnitude, and phase.

All presented results are obtained using the same feed-forward ANN architecture. The details of ANN architecture configuration are summarized in Fig. 29.

The custom normalization layer normalizes the input impulse response, including the regularization noise, so the energy equals one. All convolutional layers and the following dense layer use the ReLU activation function. The last dense layer uses ReLU activation for delay outputs and sigmoid activation for weight outputs. The weight outputs are active only for training with an unknown number of MPCs.

Also, many hyperparameters are fixed for studied cases. In this study, I focus on the estimation of well-separated MPCs. For this reason, the minimum separation between two delays is two samples of CIR. The study of performance for closely separated and hence highly correlated MPCs is a target for future work. The STD of regularization noise added during training is adding the required amount of noise to obtain the desired MPC SNR. The decay step is set as 50 times the ratio between the training data size and the batch size. As a result, the decay rate defines how much the learning rate decreases every 50 epochs. The list of fixed parameters is summarized in Table 6.3.

After the training, the ANN model is saved and used to evaluate a newly generated dataset. The same dataset is then processed using the SAGE and SLSE algorithms. Note

Table 6.3: List of fixed parameters

nSa	64	$nBatch$	2^7
$nData$	2^{17}, 2^{18}	$lrInit$	10^{-4}
$SNRmin$	15 dB	$lrStep$	$50\frac{nData}{nBatch}$
$SNRmax$	25 dB	$lrRate$	0.9
$minSep$	2	$nEpochs$	1000
$stdIn$	$2\pi/\sqrt{2 \cdot nSa}$	$lossW$	0.1

that the SAGE algorithm cannot estimate the true number of MPCs and is therefore provided with the number of MPCs. Hence, the SAGE algorithm [Fle+99] does not need the model order estimation methods like AIC or MDL. The current state-of-the-art approach to channel parameter estimation, the SLSE, is a deterministic approximation approach, which provides MAP estimates of all channel parameters, including the model order [HFR18]. When multiple MPCs are assumed, the Munkres algorithm [Kuh55] is used to associate the predicted delays with their true values.

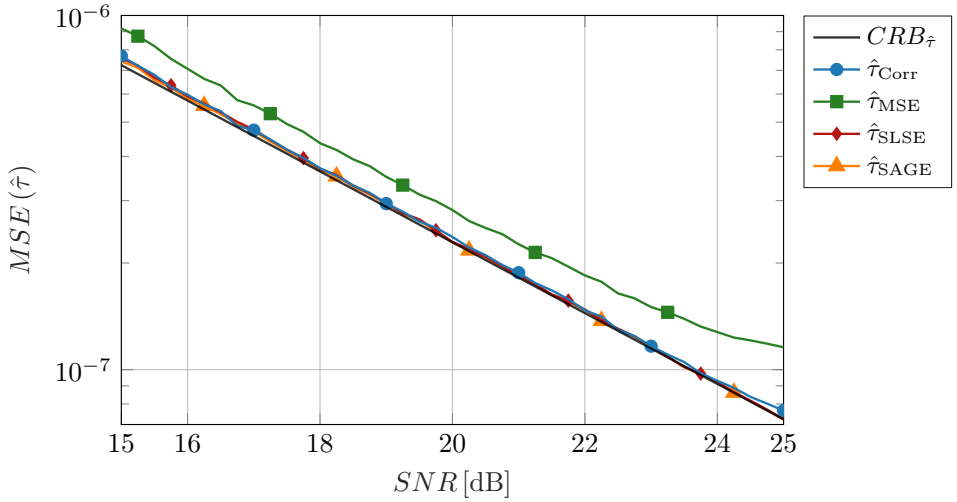
The precision evaluation is done using the PDF of error between the estimated delays $\hat{\boldsymbol{\tau}}$ and the true delays $\boldsymbol{\tau}$. The fair comparison for different SNR values is achieved by normalizing the error by the corresponding $\sqrt{CRB_{\hat{\tau}}}$ obtained from (1.11).

Performance for single multipath component

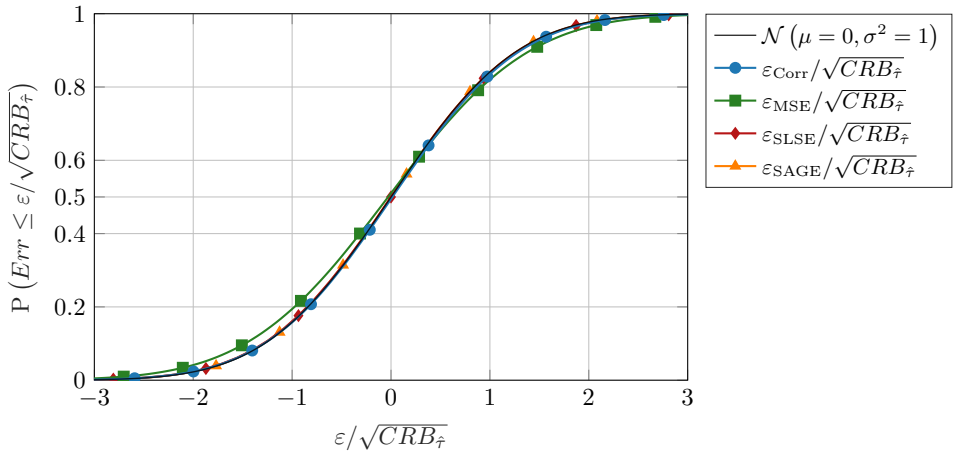
First, the training of the ANN using a dataset with random MPC SNR is performed. Then, 41 datasets with SNR between 15 dB and 25 dB with 0.5 dB step are generated to evaluate the performance. Each dataset has fixed SNR and consists of 2^{17} impulse responses with one random MPC delay.

The comparison of the MSE for different values of SNR is shown in Fig. 30a. The figure shows the MSE as a function of the MPC SNR. The black line shows the theoretical limit given by the CRLB. The blue $\hat{\tau}_{\text{Corr}}$ line shows the MSE achieved by the ANN architecture trained with the combination of the correlation loss function (6.9) and the delay MSE loss function (6.5) combined according to (6.11). The green $\hat{\tau}_{\text{MSE}}$ line shows the MSE achieved by the ANN architecture trained with the delay MSE loss (6.5). Finally, the red $\hat{\tau}_{\text{SLSE}}$ and yellow $\hat{\tau}_{\text{SAGE}}$ lines show the MSE achieved by the SLSE and SAGE algorithms, respectively.

The SAGE and SLSE closely follow the theoretical limit given by CRLB in the whole SNR span. The ANN architecture trained using only the delay MSE loss function does not approach the CRLB. Moreover, with increasing SNR, the performance gap between the CRLB and ANN is even increasing since the ANN slowly learns to estimate the delay with the high precision achievable for the higher MPC SNR. This drawback of the ANN-based approach is successfully mitigated using the newly proposed correlation loss function (6.9) together with the delay MSE loss function (6.5). When using the weighted combination of MSE and correlation losses, according to (6.11), the delay estimation of one MPC in noise approaches the CRLB theoretical limit and is comparable with both SAGE and SLSE algorithms.



(a) Comparison of the MSE for different values of MPC SNR.



(b) Comparison of the delay estimation error CDFs. The $\sqrt{CRB_{\hat{\tau}}}$ corresponding to the actual SNR normalizes the delay error obtained from all 41 datasets with different SNR values.

Figure 30: Performance comparison of the proposed ANN-based method with the CRLB theoretical limit, the SAGE, and SLSE algorithms for a single MPC in noise. The proposed ANN-based method using a combination of the delay MSE and the correlation loss functions $\hat{\tau}_{\mathrm{Corr}}$ is further compared with the ANN using only the delay MSE loss function $\hat{\tau}_{\mathrm{MSE}}$ for training.

Table 6.4: Comparison of mean and variance of the proposed ANN-based estimator with the theoretical CRLB and the benchmark approaches SAGE and SLSE for a single MPC in noise. The overall MSE estimation is obtained as $\mu^2 + \sigma^2$.

Method	$\mu/\sqrt{CRB_{\hat{\tau}}}$	$\sigma^2/CRB_{\hat{\tau}}$	MSE$/CRB_{\hat{\tau}}$
$\hat{\tau}_{\text{Ideal}}$	0	1	1
$\hat{\tau}_{\text{Corr}}$	+1.5e-2	1.030	1.031
$\hat{\tau}_{\text{MSE}}$	-3.1e-2	1.281	1.281
$\hat{\tau}_{\text{SLSE}}$	+4.8e-4	1.022	1.022
$\hat{\tau}_{\text{SAGE}}$	+4.0e-4	1.014	1.014

If the $\sqrt{CRB_{\hat{\tau}}}$ normalizes the delay estimation error, it is possible to compare the error CDFs of all approaches independently of the SNR value. After normalizing the delay estimation error by the corresponding $\sqrt{CRB_{\hat{\tau}}}$, the theoretically optimal error CDF becomes the CDF of zero-mean , unit variance normal distribution $\mathcal{N}(\mu = 0, \sigma^2 = 1)$. Fig. 30b shows the error CDF of the delay estimation approaches and compares them with the theoretically optimal error CDF. Note that Fig. 30b uses the error rather than the absolute value of the error because the deviation from the zero-mean value would be better visible in the CDF when the error is not in absolute value.

The SAGE and SLSE error closely follows the theoretical error CDF. The ANN-based estimator error also follows a normal distribution. However, it can be noticed that the normalized error variance of the ANN-based approach using the delay MSE loss function is larger than one. When the newly proposed correlation loss is utilized during the ANN training, the variance decreases and is similar to the optimal benchmark methods SAGE and SLSE.

Since the error CDF of all delay estimation methods closely follow the normal distribution, the first two moments of the error PDF normalized by $\sqrt{CRB_{\hat{\tau}}}$ suffice as the performance comparison. The mean and variance of all methods are included in Table 6.4. The experiments show that the ANN-based approach tends to have a higher mean delay estimation error than the classical approaches.

Usually, fine training with a reduced learning rate can decrease the mean of delay estimation error. During the fine training, the sign of the mean changes. Hence, the rather high mean error might be linked with the Adam optimizer overshoot issue. Using the AMSGrad [RKK19] optimizer to improve the convergence of the Adam Optimizer did not improve the performance.

Performance for Two Multipath Components

First, the training of the ANN using a dataset with random MPC SNR is performed. Then, 41 datasets with SNR between 15 dB and 25 dB are generated to evaluate the performance. Each dataset has a fixed SNR of both MPCs and consists of 2^{15} CIRs with two random MPC delays. The minimal delay separation is set to $\tau_{\Delta_{\text{MIN}}} = 2/nSa$. The delay separation assures that the MPCs can always be distinguished as separate MPCs.

The datasets with constant SNR show the performance for the given SNR, but it does not capture the proper performance. In reality, each MPC in a CIR can have different SNR. Therefore, I create an additional dataset where each MPC has a random SNR. The SNR of each MPC is sampled uniformly in a range between 15 dB and 25 dB. The random SNR dataset serves to evaluate the overall performance of the trained model. The random SNR dataset consists of 2^{18} CIRs.

The comparison of the MSE for different values of SNR is shown in Fig. 31a. The figure shows that the SAGE and SLSE closely follow the theoretical limit given by the CRLB in the whole SNR span. The ANN architecture trained using only the delay MSE loss function (6.5) has again worse performance than the architecture trained using the combination of the delay MSE and the correlation loss functions in (6.6).

Three different methods of data association during the training are compared. When using the ordered loss marked as *Ord*, the ANN must learn to order the outputted delays in ascending order.

The suboptimal data association method marked as *Adj* is associating each true delay value with the best corresponding prediction one at a time. The *Adj* data association method performs better than the *Ord* method but needs a longer training time.

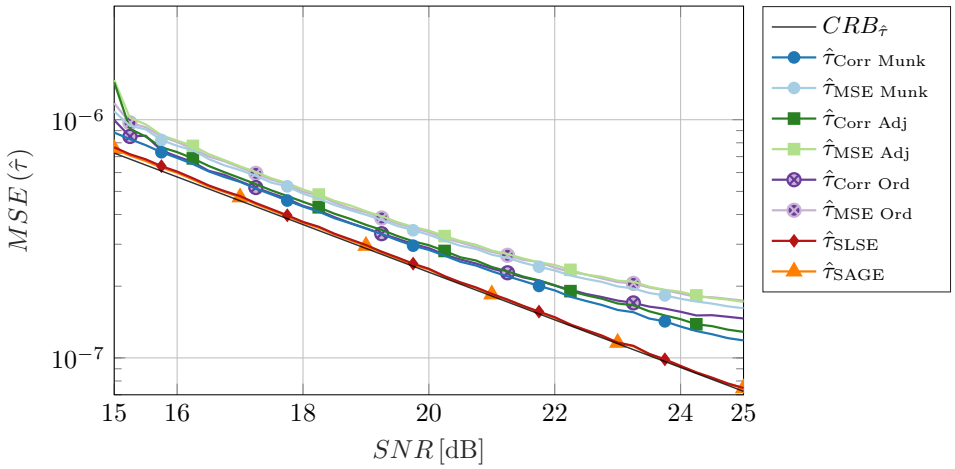
The last data association method (*Munk*) uses an optimal Munkres algorithm to associate the predicted delays with the true values during the training. The ANN trained using the optimal data association provides the best performance for the cost of the slowest training.

Even after doubling the training dataset size, none of the trained models follow the CRLB closely. At this point, it is not apparent if further increasing the dataset size can solve this issue or fine-tuning the model hyperparameters is required.

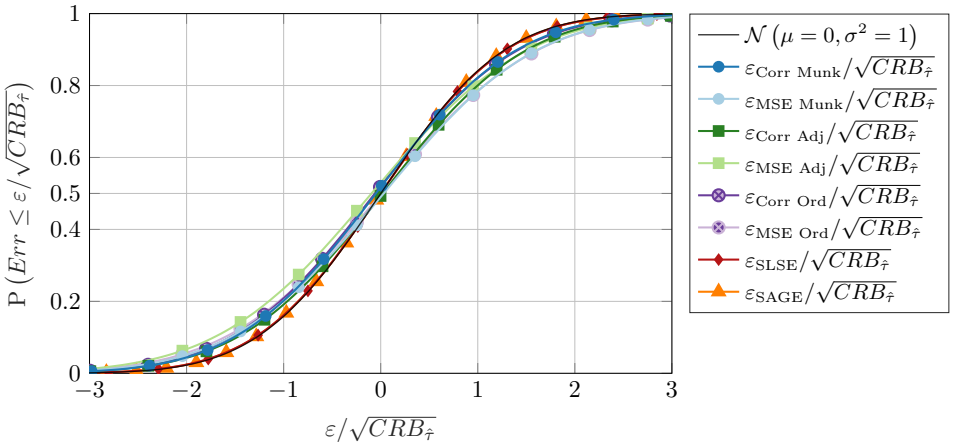
The random SNR dataset evaluates the performance of the trained ANN models in a realistic scenario where each MPC has a random SNR value. The delay error of each MPC is normalized using the corresponding $\sqrt{CRB_{\hat{\tau}}}$. Fig. 31b shows the normalized error CDF of the delay estimation approaches and compares them with the theoretically optimal error CDF.

The SAGE and SLSE error closely follows the theoretical error CDF given by the $\mathcal{N}(\mu = 0, \sigma^2 = 1)$. The ANN-based estimator error also follows a normal distribution. However, it can be noticed in Fig. 31b that the normalized error variance of the ANN-based approaches is larger than one.

The quantitative performance comparison is in Table 6.5. Similar to the single MPC experiment, the ANN-based approach tends to have a higher mean delay estimation error than the classical approaches. Interestingly, the performance is similar for all tested ANN-based methods. It seems that the data association provides only a small improvement for the two MPCs scenario. However, the combination of the delay MSE loss and the proposed correlation loss always improves the performance. Again, using the AMSGrad [RKK19] option did not improve the convergence of the Adam Optimizer.



(a) Comparison of the MSE for different values of MPC SNR.



(b) Comparison of the delay estimation error CDFs. The $\sqrt{CRB_{\hat{\tau}}}$ corresponding to the actual SNR normalizes the delay error obtained from all 41 datasets with different SNR values.

Figure 31: Performance comparison of the proposed ANN-based method with the CRLB theoretical limit, the SAGE, and SLSE algorithms for two MPCs in noise. The proposed ANN-based method using a combination of the MSE and the correlation loss functions $\hat{\tau}_{\text{Corr}}$ is further compared with the ANN using only the delay MSE loss function $\hat{\tau}_{\text{MSE}}$ during the training. Three different methods of data association during the training are also compared.

Table 6.5: Comparison of mean and variance of the proposed ANN-based estimator with the theoretical CRLB and the benchmark approaches SAGE and SLSE for two MPCs in noise. The overall MSE estimation is obtained as $\mu^2 + \sigma^2$.

Method	$\mu/\sqrt{CRB_{\hat{\tau}}}$	$\sigma^2/CRB_{\hat{\tau}}$	MSE$/CRB_{\hat{\tau}}$
$\hat{\tau}_{\text{Ideal}}$	0	1	1
$\hat{\tau}_{\text{Corr Munk}}$	-5.0e-2	1.319	1.321
$\hat{\tau}_{\text{MSE Munk}}$	-3.3e-2	1.598	1.599
$\hat{\tau}_{\text{Corr Adj}}$	+1.7e-2	1.408	1.408
$\hat{\tau}_{\text{MSE Adj}}$	-9.5e-2	1.651	1.660
$\hat{\tau}_{\text{Corr Ord}}$	-6.2e-2	1.377	1.381
$\hat{\tau}_{\text{MSE Ord}}$	+9.2e-3	1.654	1.654
$\hat{\tau}_{\text{SLSE}}$	-1.0e-4	1.030	1.030
$\hat{\tau}_{\text{SAGE}}$	+3.0e-5	1.023	1.023

Performance for an unknown number of multipath components

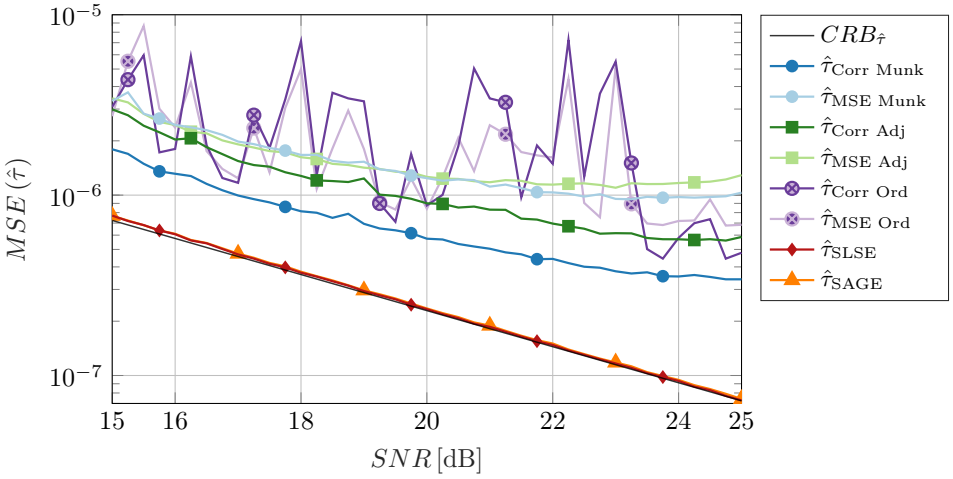
Similar to the previous approach, the ANN training is done using a dataset with random MPC SNR. However, the number of MPCs is unknown and random. In general, the number of MPCs can be arbitrary, but the number of MPCs is randomly chosen between one and two for this first study.

The evaluation uses 41 datasets with SNR between 15 dB and 25 dB. Each dataset has a fixed SNR of all MPCs present in the CIR and consists of 2^{15} CIRs. The number of MPCs is also randomly set to one or two MPCs. The minimal delay separation is set to $\tau_{\Delta_{\text{MIN}}} = 2/nSa$.

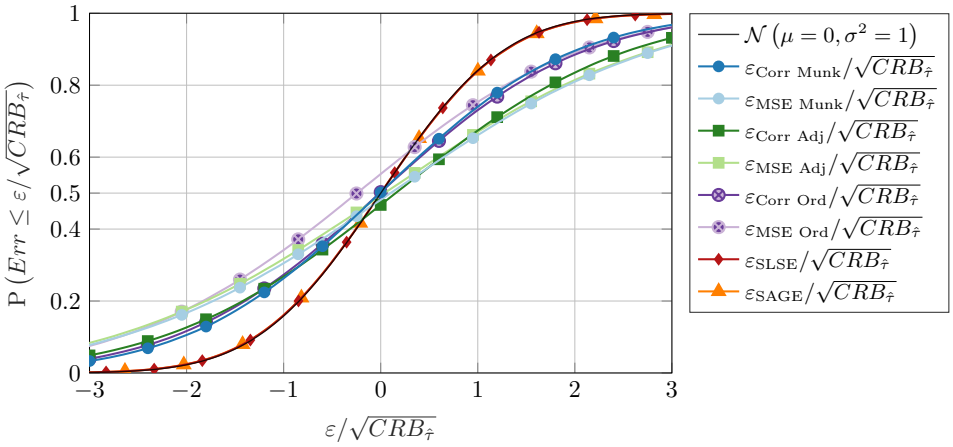
As in the scenario with a fixed number of MPCs, I create an additional dataset where each MPC has a random SNR. The SNR of each MPC is sampled uniformly in a range between 15 dB and 25 dB. The random SNR dataset consists of 2^{18} CIRs.

The comparison of the MSE for different values of SNR is shown in Fig. 32a. The figure shows that the SAGE and SLSE closely follow the theoretical limit given by the CRLB in the whole SNR span. The combination of the MSE and correlation loss functions outperforms the ANN trained using only the delay MSE loss function. The *Adj* data association method performs better than the *Ord* method but needs a longer training time. Moreover, the *Ord* method has varying performance for different SNR values. This inconsistency seems to be mitigated when the ANN does not need to learn to order the MPCs on its own and uses one of the presented association methods. Since this problem was not observed for the fixed number of MPC scenarios, it seems that the ANN cannot associate the MPCs and estimate the model order jointly. The best performance is again obtained using the optimal *Munk* association technique.

The presented performance was obtained using the training dataset with 2^{18} CIRs. The obtained performance is worse than the known model order setting, but it still provides interesting precision. Even if the MSE is approximately 4.7 times larger than the CRLB for the 25 dB SNR, the delay error STD is approximately 26.8 times smaller than the spacing between CIR samples. This clearly shows that the ANN-based multipath



(a) Comparison of the MSE for different values of MPC SNR.



(b) Comparison of the delay estimation error CDFs. The $\sqrt{CRB_{\hat{\tau}}}$ corresponding to the actual SNR normalizes the delay error obtained from all 41 datasets with different SNR values.

Figure 32: Performance comparison of the proposed ANN-based method with the CRLB theoretical limit, the SAGE, and SLSE algorithms for an unknown number of MPCs in noise. The proposed ANN-based method using a combination of the MSE and the correlation loss functions $\hat{\tau}_{\text{Corr}}$ is further compared with the ANN using only the delay MSE loss function $\hat{\tau}_{\text{MSE}}$ during the training. Three different methods of data association during the training are also compared.

radio channel estimator provides a super-resolution performance.

Fig. 32b shows the normalized error CDF of the delay estimation methods and compares them with the theoretically optimal error CDF. Again, the random SNR dataset is used to show a realistic performance in a case where each MPC in a CIR has different SNR. The delay error of each MPC is normalized using the corresponding $\sqrt{CRB_{\hat{\tau}}}$.

The SAGE and SLSE error closely follows the theoretical error CDF given by the $\mathcal{N}\,(\mu = 0, \sigma^2 = 1)$. The ANN-based estimator error also follows a normal distribution. Again, it has a larger variance than one.

Table 6.6: Comparison of mean and variance of the proposed ANN-based estimator with the theoretical CRLB and the benchmark approaches SAGE and SLSE for an unknown number of MPCs in noise. The overall MSE estimation is obtained as $\mu^2 + \sigma^2$.

Method	$\mu/\sqrt{CRB_{\hat{\tau}}}$	$\sigma^2/CRB_{\hat{\tau}}$	MSE$/CRB_{\hat{\tau}}$
$\hat{\tau}_{\text{Ideal}}$	0	1	1
$\hat{\tau}_{\text{Corr Munk}}$	-9.1e-3	2.924	2.924
$\hat{\tau}_{\text{MSE Munk}}$	+1.1e-1	7.005	7.018
$\hat{\tau}_{\text{Corr Adj}}$	+1.9e-1	4.656	4.691
$\hat{\tau}_{\text{MSE Adj}}$	+4.6e-2	7.463	7.466
$\hat{\tau}_{\text{Corr Ord}}$	+6.0e-3	6.677	6.677
$\hat{\tau}_{\text{MSE Ord}}$	-3.2e-1	5.667	5.770
$\hat{\tau}_{\text{SLSE}}$	+9.9e-4	1.024	1.024
$\hat{\tau}_{\text{SAGE}}$	-5.1e-4	1.034	1.035

The quantitative performance comparison is in Table 6.5. Interestingly, the individual approaches' performance differences are more severe than for the fixed number of MPCs scenarios. The data association improves the performance significantly when the number of MPCs has to be estimated by the ANN. Combining the delay MSE loss with the proposed correlation loss function improves the performance significantly when the optimal Munkres data association method is used.

Comparison of the calculation complexity

During the ANN training stage, the algorithm has to evaluate the loss and gradients to update all the trained model weights. For this reason, the training step is slower than the evaluation using the pre-trained model. Hence, I compare the speed of the ANN approach with the benchmark methods only for the model evaluation. Additionally, I compare the relative speed of the training for the presented training approaches with different loss functions. Since all the trained ANN models consist of the same number of trained parameters, the comparison shows the complexity of the different loss functions.

During the training stage, the Munkres algorithm-based data association performs one training epoch with two MPCs approximately 31% slower than the *Ord* approach, which does not perform any data association during the training. The *Adj* method is only 10% slower than the *Ord* training method.

However, the difference changes when using the additional correlation loss function. With the combined loss functions, the Munkres data association is approximately 17% slower than the *Ord* approach, and the *Adj* method is 13% slower than the *Ord* training method.

The training with the additional correlation loss is slower, but it provides better performance than without it. Hence, the training with the proposed correlation loss is preferred.

The duration of a training epoch with the maximal model order of two MPCs using delay MSE loss and the combination of the delay MSE and correlation loss with batch

Table 6.7: The comparison of the training epoch duration of different ANN training methods in seconds.

	Ord	Adj	Munk
MSE	104	115	150
MSE and Corr	144	166	174

Table 6.8: Comparison of the evaluation time of the ANN-based approach with the benchmark methods

	SLSE	SAGE	MSE	MSE and Corr
CPU $nBatch = 1$	15.3 ms	2.02 ms	4.86 ms	5.02 ms
GPU $nBatch = 1$			4.52 ms	6.52 ms
GPU $nBatch = 2^7$			85.2 μs	106 μs
GPU $nBatch = 2^8$			69.2 μs	85.2 μs

size equal to 2^7 and dataset size of 2^{18} is compared in Table 6.7.

The increase in the training complexity for a low number of MPCs is almost negligible. However, it might be preferred for a large number of MPCs to use the *Adj* association method, which scales linearly with the number of MPCs.

Comparison of the evaluation time is difficult since the benchmark algorithms are sequential optimization methods utilizing CPU, while the ANN approach leverages parallel computing using GPU. The strength of the ANN comes from the possibility to evaluate multiple CIRs in parallel while evaluating a single measurement at the time can be slow.

For this reason, I show the evaluation speed for the benchmark algorithms and the trained ANNs. I use different batch sizes for GPU-based evaluation and a CPU to evaluate the ANN. The average duration of processing one CIR is in Table 6.8. Using the combination of the MSE and correlation loss is slightly slower, which should not be since the loss function calculation is performed only during the training. The actual implementation of the method can cause this discrepancy. However, the ANN output ordering type does not affect the evaluation speed.

Chapter 7

Conclusion

This thesis proposes a new version of the Channel-SLAM, including a novel IMU-based pedestrian transition model, a joint data association algorithm, and a GMM-based representation of the VTs. It evaluates the precision boost achieved by the proposed improvements compared to the previous version of the Channel-SLAM. Also, it shows the possibility of using the idea of multipath assisted positioning for positioning using a SFN. Furthermore, the thesis studies the possibility of using ANNs to speed up the radio channel parameter estimation vital for all multipath assisted positioning methods.

The main contributions of this thesis are:

- The demonstration of the usability of the multipath assisted positioning idea for SFN in Chapter 2. Multiple transmitters transmits the same signal synchronously in a SFN. Then, the received signal consists of multiple delayed replicas of the transmitted signal, similarly to multipath propagation, and hence can be interpreted as MPCs. Thus, the channel multipath model can explain this received signal, and the estimated MPCs are used for positioning. The presented approach was tested using a real-case scenario using a DVB-T service. This contribution fulfills the thesis goal: Research the possibility of positioning in Single Frequency Network and show the equivalence with multipath assisted positioning [KP15; NK16; NKV16; KV18].

- The novel pedestrian transition model proposed in Chapter 3. The novel pedestrian transition model utilizes more information from the IMU than the previous version, which only used gyroscope measurements for the heading estimation. In addition to the gyroscope measurements, the presented novel model uses the accelerometer and a step detection-based approach to estimate the pedestrian walking speed. The slower divergence of the novel pedestrian transition model provides improved convergence of the VT position estimation, which again improves the precision of the mobile receiver position estimation. This contribution fulfills the thesis goal:

Propose and test a novel pedestrian transition model for Channel-SLAM using inertial sensor fusion [KG20].

- Chapter 4 contains the derivation of the joint data association for the Channel-SLAM. The derived joint data association algorithm associating MPCs with the previously observed VTs decreases the computational load significantly. Also, the reusing of previously observed VTs allows for estimating its position more accurately, improving the positioning precision. Moreover, the proposed approach can detect the association error caused by the radio channel estimator. This contribution fulfills the thesis goal: Propose and verify a stochastic data association method for tracking and associating multipath components with the Virtual Transmitters in Channel-SLAM [KG20].

- A novel representation of the VTs by the GMM was proposed in Chapter 5. The motivation for using the GMM machine learning technique was the fact that the newly initiated VTs carry low information for the Channel-SLAM. Hence, the low order GMM is sufficient early after the VT initialization. In contrast, the PF-based VT model has to be initialized with orderly larger model order reduced over time. The parameters of the GMM are learned sequentially from the VT observations. It was demonstrated that the GMM achieves better precision than the PF-based VT representation while using a significantly smaller initial model order. This novel approach utilizes the computational resources better because the VT provides marginal information to the Channel-SLAM shortly after initialization. This contribution fulfills the thesis goal: Research the possibility of using a machine learning-based methods for Virtual Transmitter estimation in Channel-SLAM [KG21].

- With the new and faster version of the Channel-SLAM, new ways of reducing the calculational load had to be investigated to allow a real-time application of the Channel-SLAM. The first results presented in Chapter 6 show that utilizing ANNs for radio channel parameter estimation can significantly decrease the calculational load. The main strength of the ANN-based approach is the parallelization of the MPC estimation that allows the processing of multiple CIRs simultaneously. However, the ANN can be faster than the classical radio channel estimators, even for a single CIR processing. The thesis proposes a novel method to handle an unknown number of MPCs present in the CIR. Also, a novel signal correlation-based loss function is proposed to train the ANN models. The results show that the signal correlation-based loss function allows approaching the theoretical CRLB when estimating the delay of a single MPC in a noisy CIR. The super-resolution performance is achieved for an unknown number of MPCs in a noisy CIR scenario. This contribution fulfills the thesis goal: Investigate the possibility of using Artificial Neural Networks for radio channel parameter estimation and compare the proposed method with the theoretical limits and benchmark algorithms.

All goals of this thesis were addressed and fulfilled.

Further steps in multipath assisted positioning with the assistance of machine learning I recommend pursuing by the successors are:

- The joint estimation of the radio channel parameters and Channel-SLAM: The Channel-SLAM and the radio channel parameter estimation are currently independent. The cooperation between the Channel-SLAM and the radio channel parameter estimator can improve the performance of both.
- The pedestrian heading and walking speed used in the proposed transition model can be insufficient for more dynamic pedestrian movement. Rather than study all possible pedestrian movement styles and search for mathematical models, it might be better to learn ANN serving as a pedestrian movement model from actual IMU data. Since Bayesian filter-based multipath assisted positioning algorithms, such as Channel-SLAM, require a stochastic transition model, Mixture Density Network (MDN) might be used as a future stochastic pedestrian transition model.
- The initial study on using the ANNs presented in this thesis could serve as a starting point for further research of model-free approaches to radio channel parameter estimators. The attractive direction could be extending the proposed ANN from a snapshot-based estimator to track the MPCs over time using a Recurrent Neural Network (RNN). Another direction could be estimating the multipath radio channel with DMCs. The existence of DMCs is often ignored by the modern channel estimation algorithms, which can decrease the precision or even cause a failure of the channel estimation process.

List of Acronyms

GNSS	Global Navigation Satellite System . . . 1
GPS	Global Positioning System . . . 25
GPU	Graphics Processing Unit . . . 76
HDoP	Horizontal Dilution of Precision . . . 24
HMM	Hidden Markov Model . . . 45
IFFT	Inverse Fast Fourier Transform . . . 5
IMU	Inertial Measurement Unit . . . 33
ISI	Inter-Symbol Interference . . . 19
JPDA	Joint Probabilistic Data Association . . . 11
JPDA-MINT	Joint Probabilistic Data Association for Multipath-assisted Indoor Navigation and Tracking . . . 11
JPDF	Joint Probability Density Function . . . 34
KEST	Kalman Enhanced Super Resolution Tracking . . . 7
KF	Kalman Filter . . . 7
LoS	Line-of-Sight . . . 2
LS	Least Squares . . . 9
MAP	Maximum A Posteriori . . . 7
MCMC	Markov Chain Monte Carlo . . . 11
MDL	Minimum Description Length . . . 9
MDN	Mixture Density Network . . . 93
MEMS	Micro-Electro-Mechanical System . . . 36
MHT	Multiple Hypothesis Tracking . . . 32
MIP	Mega-Frame Initialization Packet . . . 17
ML	Maximum Likelihood . . . 6
MPC	Multipath Component . . . 2
MSE	Mean-Square Error . . . 74
MUSIC	MUltiple SIgnal Classification . . . 7
NDoP	North-South Dilution of Precision . . . 24
NLoS	Non-Line-of-Sight . . . 2
OFDM	Orthogonal Frequency Division Multiplexing . . . 17
PDF	Probability Density Function . . . 11
PDP	Power Delay Profile . . . 9
PF	Particle Filter . . . 10
PMF	Probability Mass Function . . . 37
PRBS	Pseudo-Random Binary Sequence . . . 18
QAM	Quadrature Amplitude Modulation . . . 17
r.h.s.	right hand side . . . 45
RBPF	Rao-Blackwellized Particle Filter . . . 11
ReLU	Rectified Linear Unit . . . 13
RMSE	Root-Mean-Square Error . . . 40
RNN	Recurrent Neural Network . . . 93
RPF	Regularized Particle Filter . . . 65
RSSI	Received Signal Strength Indication . . . 5
SAGE	Space-Alternating Generalized Expectation-maximization . 7

SDR Software-Define Radio 27
SFN Single Frequency Network 2
SGD Stochastic Gradient Descent 14
SLAM Simultaneous Localization And Mapping 2
SLSE Superfast Line Spectral Estimation 9
SNR Signal-to-Noise Ratio 5
SP Scattered Pilot 18
STD Standard Deviation 34
SVD Singular Value Decomposition 78
TDoA Time Difference of Arrival 1
ToA Time of Arrival 1
TPS Transmission Parameter Signalling 18
UWB Ultra-WideBand 5
VDoP Vertical Dilution of Precision 24
VT Virtual Transmitter 2
WNA White Noise Acceleration 35
WSSUS Wide-Sense Stationary Uncorrelated Scattering 4

List of Latin Symbols

b_{u}	receiver clock bias after multiplying by the speed of light
$b_{\mathrm{v},\ell}^{(k\,p)}$	p-th subordinate PF additional propagation distance particle of ℓ-th VT of k-th superordinate PF
$b_{\mathrm{v},\ell\,t}$	additional propagation distance of ℓ-th VT
$b_{\dot{\Psi}}$	yaw rate bias in rad/s
$h\left(\tau\right)$	CIR as a function of τ
K	number of system state PF particles
k	index of system state particle
L	number of MPCs or VTs
ℓ	MPC or VT index
M	number of $\boldsymbol{s}$ samples
$n\left(t\right)$	zero-mean AWGN noise signal at time t
$\mathbf{n}$	data association vector
P	number of VT PF particles
p	index of VT particle
P_{D}	detection probability
P_{FA}	false alarm probability
$r\left(t\right)$	transmitted signal at time t
$s\left(t\right)$	received signal at time t
$\boldsymbol{S}$	frequency spectrum of $\boldsymbol{s}$
$\boldsymbol{s}$	sampled received signal
$\tilde{\boldsymbol{S}}$	frequency spectrum of $\tilde{\boldsymbol{s}}$
$\tilde{\boldsymbol{s}}$	noise-free sampled received signal
$\tilde{s}\left(t\right)$	noise-free received signal at time t
T_{Δ}	guard interval duration
τ_i	track indicator
$w^{(k)}$	weight of $k$superordinate PF particle
$w_{\ell}^{(k\,p)}$	p-th subordinate PF particle weight of ℓ-th VT of k-th superordinate PF
$\mathbf{x}_{\mathrm{Rx}}$	system Cartesian position coordinates
$\mathbf{x}_{\mathrm{Tx},\ell}$	Cartesian position coordinates of ℓ-th VT
$\mathbf{x}_{\mathrm{u}}^{(k)}$	k-th system state superordinate PF particle
$\mathbf{x}_{\mathrm{u}}$	system state

$\mathbf{x}_{\mathrm{v},\ell}^{(k\,p)}$	p-th subordinate PF particle of ℓ-th VT of k-th superordinate PF
$\mathbf{x}_{\mathrm{v},\ell}$	ℓ-th VT state
$\mathbf{z}$	measurement vector of delay estimates obtained from $\boldsymbol{s}$

List of Greek Symbols

α	MPC amplitude
$\boldsymbol{\alpha}$	concatenation of MPC amplitudes
β	STD estimation of IMU acceleration
$\dot{\beta}$	time derivation of β
γ	pedestrian step length
γ_{th}	CA-CFAR threshold
$\delta(\cdot)$	Dirac delta distribution
δ_t	time step duration
δ_ℓ	detection indicator for $\mathbf{x}_{\mathrm{v},\ell\,t}$
ε	flag indicating the validity of IMU-based control signals
ζ	IMU-based step detection flag
$\boldsymbol{\theta}$	concatenation of all relevant system parameters
$\Lambda(\tau)$	log-likelihoo function
ν_i	new target indicator for $z_{i\,t}$
ξ_t	IMU-based *ismoving* flag
$\sigma_{\cdot}$	STD of subscript variable
$\sigma_{\cdot}^2$	variance of subscript variable
τ	MPC delay
$\boldsymbol{\tau}$	concatenation of MPC delays
ϕ	number of false alarms
Ψ	system state yaw in rad
$\dot{\Psi}$	system state yaw rate in rad/s
ω	pedestrian step frequency

List of Notation and Other Symbols

$\cdot^*$	element-wise complex conjugation of superscripted object
$\cdot^{\mathrm{T}}$	transpose operation applied to superscripted object
$\cdot^{\mathrm{H}}$	conjugate transpose operation applied to superscripted object
$*$	convolution operator
∇	gradient operator
$\angle$	phase of a complex number
$\lvert\cdot\rvert$	cardinality of a set or number of dimensions of a vector
$\lVert\cdot\rVert$	norm function
$\mid\mid$	logical *or* operator
$!$	logical *not* operation
$\cdot_t$	value of subscripted object at time step t
$\cdot^{(i)}$	i-th PF particle or set element
$\hat{\cdot}$	estimator or estimate of the marked variable
$\mathfrak{F}$	FFT operator
$\mathfrak{F}^{-1}$	IFFT operator
$\arg\max$	returns subscripted parameter for which is the function on r.h.s. maximized
$\arg\min$	returns subscripted parameter for which is the function on r.h.s. minimized
$\mathrm{p}(A \mid B)$	conditional PDF of random variable A conditioned on B
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multi-variate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$

Bibliography

[Aba+15] M. Abadi et al. *TensorFlow: large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org. 2015.

[AC10] A. Agostini and E. Celaya. "Reinforcement learning with a Gaussian mixture model". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2010.

[AG87] G. S. Ammar and W. B. Gragg. "The generalized Schur algorithm for the superfast solution of Toeplitz systems". In: *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1987, pp. 315–330.

[Ama98] S. Amari. "Natural gradient works efficiently in learning". In: *Neural Computation* 10.2 (Feb. 1998), pp. 251–276.

[APV18] S. Adavanne, A. Politis, and T. Virtanen. "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, Sept. 2018.

[Aru+02] M. Arulampalam et al. "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking". In: *IEEE Transactions on Signal Processing* 50.2 (2002), pp. 174–188.

[AS72] D. Alspach and H. Sorenson. "Nonlinear Bayesian estimation using Gaussian sum approximations". In: *IEEE Transactions on Automatic Control* 17.4 (Aug. 1972), pp. 439–448.

[BB03] D. Bizup and D. Brown. "The over-extended Kalman filter - don't use it!" In: *Sixth International Conference of Information Fusion, 2003. Proceedings of the*. IEEE, 2003.

[Bel63] P. Bello. "Characterization of randomly time-variant linear channels". In: *IEEE Trans. Commun. Syst.* 11.4 (Dec. 1963), pp. 360–393.

[Bis06] C. M. Bishop. *Pattern recognition and machine learning (information science and statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[BL95] Y. Bar-Shalom and X.-R. Li. *Multitarget–multisensor tracking: principles and techniques*. Yaakov Bar-Shalom, 1995. ISBN: 978-0964831209.

[BLK01] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: algorithms and software for information extraction*. Wiley, 2001. ISBN: 978-0-471-41655-5.

[Boa15] B. Boashash. *Time-frequency signal analysis and processing*. 2nd ed. London, United Kingdom: Elsevier, 2015.

[Boz87] H. Bozdogan. "Model selection and akaike's information criterion (AIC): the general theory and its analytical extensions". In: *Psychometrika* 52.3 (Sept. 1987), pp. 345–370.

[BP00] P. Bahl and V. N. Padmanabhan. "Radar: an in-building rf-based user location and tracking system". In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*. Vol. 2. Mar. 2000, 775–784 vol.2.

[BSB97] J. van de Beek, M. Sandell, and P. Borjesson. "ML estimation of time and frequency offset in OFDM systems". In: *IEEE Transactions on Signal Processing* 45.7 (July 1997), pp. 1800–1805.

[Cap69] J. Capon. "High-resolution frequency-wavenumber spectrum analysis". In: *Proceedings of the IEEE* 57.8 (Aug. 1969), pp. 1408–1418. ISSN: 0018-9219.

[CH17] S. Chakrabarty and E. A. P. Habets. "Broadband doa estimation using convolutional neural networks trained with noise signals". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Oct. 2017.

[Che+15] L. Chen et al. "TOA estimation for positioning with DVB-T signals in outdoor static tests". In: *IEEE Transactions on Broadcasting* 61.4 (Dec. 2015), pp. 625–638.

[Che99] P.-C. Chen. "A non-line-of-sight error mitigation algorithm in location estimation". In: *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No.99TH8466)*. Vol. 1. 1999, 316–320 vol.1.

[Cra46] H. Cramér. *Mathematical methods of statistics*. Princeton, NJ, USA: Princeton University Press, 1946.

[CS95] J. Caffery and G. Stuber. "Radio location in urban cdma microcells". In: *Proceedings of 6th International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol. 2. 1995, 858–862 vol.2.

[CWM02] D. Cassioli, M. Win, and A. Molisch. "The ultra-wide bandwidth indoor channel: from statistical model to simulations". In: *IEEE Journal on Selected Areas in Communications* 20.6 (2002), pp. 1247–1257.

[CYY06] C. Ching-Yung, W. Yi-Ting, and H. Yung-Hua. *Time-frequency correlation-based synchronization for coherent OFDM receiver*. 2006.

[Dav12] S. L. Davis. *Google Earth Toolbox*. 2012.

[DHS11] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the em algorithm". In: *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 39.1 (1977), pp. 1–38.

[Dou+00] A. Doucet et al. "Rao-Blackwellised particle filtering for dynamic Bayesian networks". In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. UAI '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 176–183. ISBN: 1-55860-709-9.

[DT09] T. Deissler and J. Thielecke. "Feature based indoor mapping using a bat-type UWB radar". In: *2009 IEEE International Conference on Ultra-Wideband*. IEEE, Sept. 2009.

[DT10] T. Deissler and J. Thielecke. "UWB SLAM with Rao-Blackwellized Monte Carlo data association". In: *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, Sept. 2010.

[ETS06] ETSI. *Digital video broadcasting (DVB); DVB mega-frame for single frequency network (SFN) synchronization*. 2004–2006.

[ETS15] ETSI. *Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television DVB-T*. 2015.

[Fal07] A. Falkenberg. *Method to Calculate the Inverse of a Complex Matrix using Real Matrix Inversion*. Tech. rep. 2007.

[FF95] J. Francos and B. Friedlander. "Bounds for estimation of complex exponentials in unknown colored noise". In: *IEEE Transactions on Signal Processing* 43.9 (1995), pp. 2176–2185.

[FH94] J. Fessler and A. Hero. "Space-alternating generalized expectation-maximization algorithm". In: *IEEE Transactions on Signal Processing* 42.10 (1994), pp. 2664–2677.

[Fle+99] B. Fleury et al. "Channel parameter estimation in mobile radio environments using the SAGE algorithm". In: *IEEE Journal on Selected Areas in Communications* 17.3 (Mar. 1999), pp. 434–450.

[Fro+13] M. Froehle et al. "Cooperative multipath-assisted indoor navigation and tracking (co-MINT) using UWB signals". In: *2013 IEEE International Conference on Communications Workshops (ICC)*. IEEE, June 2013.

[Gen+16a] C. Gentner et al. "Multipath assisted positioning with simultaneous localization and mapping". In: *IEEE Transactions on Wireless Communications* 15.9 (Sept. 2016), pp. 6104–6117.

[Gen+16b] C. Gentner et al. "Simultaneous localization and mapping in multipath environments". In: *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, Apr. 2016.

[Gen+17] C. Gentner et al. "Positioning using terrestrial multipath signals and inertial sensors". In: *Mobile Information Systems* 2017 (2017), pp. 1–18.

[GJ13] C. Gentner and T. Jost. "Indoor positioning using time difference of arrival between multipath components". In: *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, Oct. 2013.

[GSS93] N. Gordon, D. Salmond, and A. Smith. "Novel approach to nonlinear/non-Gsian Bayesian state estimation". In: *IEE Proceedings F Radar and Signal Processing* 140.2 (1993), p. 107.

[GU17] C. Gentner and M. Ulmschneider. "Simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope". In: *International Conference on Indoor Positioning and Indoor Navigation*. Saporo, Japan, Sept. 2017.

[HFR18] T. L. Hansen, B. H. Fleury, and B. D. Rao. "Superfast line spectral estimation". In: *IEEE Transactions on Signal Processing* 66.10 (May 2018), pp. 2511–2526.

[HSW89] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366.

[HYK21] Y.-S. Hsiao, M. Yang, and H.-S. Kim. "Super-resolution time-of-arrival estimation using neural networks". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, Jan. 2021.

[Ian82] J. Ianniello. "Time delay estimation via cross-correlation in the presence of large estimation errors". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 30.6 (1982), pp. 998–1003.

[IBF19] G. Izacard, B. Bernstein, and C. Fernandez-Granda. "A learning-based framework for line-spectra super-resolution". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019.

[IMF19] G. Izacard, S. Mohan, and C. Fernandez-Granda. "Data-driven estimation of sinusoid frequencies". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[IWW83] J. Ianniello, E. Weinstein, and A. Weiss. "Comparison of the Ziv-Zakai lower bound on time delay estimation with correlator performance". In: *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 8. 1983, pp. 875–878.

[JDW08] D. B. Jourdan, D. Dardari, and M. Z. Win. "Position error bound for UWB localization in dense cluttered environments". In: *IEEE Transactions on Aerospace and Electronic Systems* 44.2 (2008), pp. 613–628.

[Jia+14] Y. Jia et al. "Caffe: convolutional architecture for fast feature embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[JLR19] Y. Jiang, H. Li, and M. Rangaswamy. "Deep learning denoising based line spectral estimation". In: *IEEE Signal Processing Letters* 26.11 (Nov. 2019), pp. 1573–1577.

[Jos+12] T. Jost et al. "Detection and tracking of mobile propagation channel paths". In: *IEEE Trans. Antennas Propag.* 60.10 (Oct. 2012), pp. 4875–4883.

[Jum+21] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* (July 2021).

[Kal60] R. E. Kalman. "A new approach to linear filtering and prediction problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45.

[Kar+21] T. Karras et al. *Alias-free generative adversarial networks*. 2021. arXiv: 2106.12423 [cs.CV].

[Kay98] S. M. Kay. *Fundamentals of statistical signal processing, volume ii: detection theory*. 1st ed. Upper Saddle River, US: Prentice-Hall PTR, 1998. ISBN: 9780135041352.

[KB17] D. P. Kingma and J. Ba. *Adam: a method for stochastic optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[KE17] M. Khedr and N. El-Sheimy. "A smartphone step counter using IMU and magnetometer for navigation and health monitoring applications". In: *Sensors* 17.11 (Nov. 2017), p. 2573.

[Kes+17] N. S. Keskar et al. *On large-batch training for deep learning: generalization gap and sharp minima*. 2017. arXiv: 1609.04836 [cs.LG].

[KFL01] F. R. Kschischang, B. J. Frey, and H. .-. Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Transactions on Information Theory* 47.2 (Feb. 2001), pp. 498–519. ISSN: 0018-9448.

[KG20] R. Karásek and C. Gentner. "Stochastic data association for multipath assisted positioning using a single transmitter". In: *IEEE Access* 8 (2020). 10.1109/ACCESS.2020.2977558, pp. 46735–46752.

[KG21] R. Karásek and C. Gentner. "Gaussian mixture model learning for multipath assisted positioning". In: *2021 15th European Conference on Antennas and Propagation (EuCAP)*. IEEE, Mar. 2021.

[KLZ13] P. Kozierski, M. Lis, and J. Zietkiewicz. "Resampling in particle filtering - comparison". In: *Studia z Automatyki i Informatyki* 38 (Jan. 2013), pp. 35–64.

[KP15] R. Karásek and J. Popp. "Broadband signals using for position estimation". In: *Proceedings of the 19th International Scientific Student Conference POSTER 2015*. Czech Technical University in Prague, May 2015. ISBN: 978-80-01-05728-5.

[KSH17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90.

[Kuh55] H. W. Kuhn. "The hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (Mar. 1955), pp. 83–97.

[KV18] R. Karásek and F. Vejražka. "The DVB-T-based positioning system and single frequency network offset estimation". In: *Radioengineering* 27.4 (Sept. 2018). 10.13164/re.2018.1155, pp. 1155–1165.

[KWC13] E. Kupershtein, M. Wax, and I. Cohen. "Single-site emitter localization via multipath fingerprinting". In: *IEEE Transactions on Signal Processing* 61.1 (Jan. 2013), pp. 10–21.

[Lee+13] J. H. Lee et al. "Real time adaptive step length estimation for smartphone user". In: *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*. IEEE, Oct. 2013.

[Lei+15] E. Leitinger et al. "Simultaneous localization and mapping using multipath channel information". In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, June 2015.

[Lei+16] E. Leitinger et al. "Belief propagation based joint probabilistic data association for multipath-assisted indoor navigation and tracking". In: *2016 International Conference on Localization and GNSS (ICL-GNSS)*. June 2016, pp. 1–6.

[Lei+17] E. Leitinger et al. "Factor graph based simultaneous localization and mapping using multipath channel information". In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. May 2017, pp. 652–658.

[LGW21] E. Leitinger, S. Grebien, and K. Witrisal. "Multipath-based SLAM using belief propagation with interacting multiple dynamic models". In: *2021 15th European Conference on Antennas and Propagation (EuCAP)*. 2021, pp. 1–5.

[Li+16] L. Li et al. "Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring". In: *Transportation Research Part C: Emerging Technologies* 64 (Mar. 2016), pp. 45–57.

[Li+18] X. Li et al. "Enhanced GMM-based filtering with measurement update ordering and innovation-based pruning". In: *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, July 2018.

[Mar20] J. Martens. *New insights and perspectives on the natural gradient method*. 2020. arXiv: 1412.1193 [cs.LG].

[ME01] P. Misra and P. Enge. *Global positioning system: signals, measurements, and performance*. 1st ed. Lincoln, US: Ganga-Jamuna Press, 2001. ISBN: 9780970954404.

[MLW14] P. Meissner, E. Leitinger, and K. Witrisal. "UWB for robust indoor tracking: weighting of multipath components for efficient estimation". In: *IEEE Wireless Communications Letters* 3.5 (Oct. 2014), pp. 501–504.

[Mol09] A. F. Molisch. "Ultra-wide-band propagation channels". In: *Proceedings of the IEEE* 97.2 (Feb. 2009), pp. 353–371.

[MP43] W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5.4 (Dec. 1943), pp. 115–133.

[NK16] V. Navrátil and R. Karásek. "Position estimate using DVB-T signal". In: *Proceedings of the 20th International Scientific Student Conference POSTER 2016*. Czech Technical University in Prague, May 2016. ISBN: 978-80-01-05950-0.

[NKV16] V. Navrátil, R. Karásek, and F. Vejražka. "Position estimate using radio signals from terrestrial sources". In: *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 10.1109/plans.2016.7479775. IEEE, Apr. 2016.

[NW17] M. Nowicki and J. Wietrzykowski. "Low-effort place recognition with WiFi fingerprints using deep learning". In: *Automation 2017*. Springer International Publishing, 2017, pp. 575–584.

[Pao94] L. Y. Pao. "Multisensor multitarget mixture reduction algorithms for tracking". In: *Journal of Guidance, Control, and Dynamics* 17.6 (Nov. 1994), pp. 1205–1211.

[Pas+19] A. Paszke et al. "Pytorch: an imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[Pra13] K. M. M. Prabhu. *Window functions and their applications in signal processing*. 1st ed. Boca Raton, US: CRC Press, 2013. ISBN: 9781466515840.

[PSW19] M. L. Piscopo, M. Spannowsky, and P. Waite. "Solving differential equations with neural networks: applications to the calculation of cosmological phase transitions". In: *Physical Review D* 100.1 (July 2019).

[Rap96] T. S. Rappaport. *Wireless communications - principles and practice*. first. Prentice Hall, 1996.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536.

[Ric05] A. Richter. "Estimation of radio channel parameters: models and algorithms". PhD thesis. Ilmenau, Germany: Technischen Universität Ilmenau, Feb. 2005.

[Ric07] V. Ricny. "Single frequency networks (SFN) in digital terrestrial broadcasting". In: *Radioengineering* 16.4 (Dec. 2007). ISSN: 1805-9600.

[Ris78] J. Rissanen. "Modeling by shortest data description". In: *Automatica* 14.5 (Sept. 1978), pp. 465–471.

[RK89] R. Roy and T. Kailath. "ESPRIT-estimation of signal parameters via rotational invariance techniques". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.7 (July 1989), pp. 984–995.

[RKK19] S. J. Reddi, S. Kale, and S. Kumar. *On the convergence of adam and beyond*. 2019. arXiv: 1904.09237 [cs.LG].

[Rot+19] J. Rothfuss et al. *Conditional density estimation with neural networks: best practices and benchmarks*. 2019. arXiv: 1903.00954 [stat.ML].

[Sal90] D. J. Salmond. "Mixture reduction algorithms for target tracking in clutter". In: *Signal and Data Processing of Small Targets 1990*. SPIE, Oct. 1990.

[San+20] A. Sanchez-Gonzalez et al. *Learning to simulate complex physics with graph networks*. 2020. arXiv: 2002.09405 [cs.LG].

[Sch78] G. Schwarz. "Estimating the dimension of a model". In: *The Annals of Statistics* 6.2 (Mar. 1978).

[Sch86] R. Schmidt. "Multiple emitter location and signal parameter estimation". In: *IEEE Transactions on Antennas and Propagation* 34.3 (Mar. 1986), pp. 276–280. ISSN: 0018-926X.

[Sil+16] D. Silver et al. "Mastering the game of go with deep neural networks and tree search". In: *Nature* 529.7587 (Jan. 2016), pp. 484–489.

[SJL97] P. Stoica, A. Jakobsson, and J. Li. "Cisoid parameter estimation in the colored noise case: asymptotic Cramer-Rao bound, maximum likelihood, and nonlinear least-squares". In: *IEEE Transactions on Signal Processing* 45.8 (1997), pp. 2048–2059.

[Sko08] M. I. Skolnik. *Radar handbook*. 3rd ed. McGraw Hill, 2008. ISBN: 9780071485470.

[SR96] M. Silventoinen and T. Rantalainen. "Mobile station emergency locating in gsm". In: *1996 IEEE International Conference on Personal Wireless Communications Proceedings and Exhibition. Future Access*. 1996, pp. 232–238.

[Sut+13] I. Sutskever et al. "On the importance of initialization and momentum in deep learning". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1139–1147.

[SW08] Y. Shen and M. Z. Win. "Effect of path-overlap on localization accuracy in dense multipath environments". In: *2008 IEEE International Conference on Communications*. 2008, pp. 4197–4202.

[SW09] Y. Shen and M. Z. Win. "On the use of multipath geometry for wideband cooperative localization". In: *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. IEEE, Nov. 2009.

[Thr+04] S. Thrun et al. "FastSLAM: an efficient solution to the simultaneous localization and mapping problem with unknown data association". In: *Journal of Machine Learning Research* 2004 (2004).

[TRD02] P. A. Torres-Carrasquillo, D. A. Reynolds, and J. Deller. "Language identification using Gaussian mixture model tokenization". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, May 2002.

[Tri+06] M. Triki et al. "Mobile terminal positioning via power delay profile fingerprinting: reproducible validation simulations". In: *IEEE Vehicular Technology Conference*. IEEE, Sept. 2006.

[Tur56] G. Turin. "Communication through noisy, random-multipath channels". In: *1956 IRE Convention Record*. Vol. 4. 1956, pp. 154–166.

[Tur72] G. L. Turin. "A statistical model of urban multipath propagation". In: *IEEE Trans. Veh. Technol.* 21.1 (Feb. 1972), pp. 1–9.

[Ulm+17] M. Ulmschneider et al. "Multiple hypothesis data association for multipath-assisted positioning". In: *2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*. Oct. 2017, pp. 1–6.

[VBC14] F. Vincent, O. Besson, and E. Chaumette. "Approximate maximum likelihood estimation of two closely spaced sources". In: *Signal Processing* 97 (Apr. 2014), pp. 83–90.

[Vin+19] O. Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning". In: *Nature* 575.7782 (Oct. 2019), pp. 350–354.

[Wan+16] X. Wang et al. "CSI-based fingerprinting for indoor localization: a deep learning approach". In: *IEEE Transactions on Vehicular Technology* (2016), pp. 1–1.

[WH96] M. Wylie and J. Holtzman. "The non-line of sight problem in mobile location estimation". In: *Proceedings of ICUPC - 5th International Conference on Universal Personal Communications*. Vol. 2. 1996, 827–831 vol.2.

[WM12] K. Witrisal and P. Meissner. "Performance bounds for multipath-assisted indoor navigation and tracking (MINT)". In: *2012 IEEE International Conference on Communications (ICC)*. IEEE, June 2012.

[Woo07] O. J. Woodman. *An Introduction to Inertial Navigation*. Tech. rep. Computer Laboratory, Cambridge, UK, Technical Report, 2007.

[XBM17] L. Xiao, A. Behboodi, and R. Mathar. "A deep learning approach to fingerprinting indoor localization solutions". In: *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, Nov. 2017.

List of Author's Publications

[KG20] R. Karásek and C. Gentner. "Stochastic data association for multipath assisted positioning using a single transmitter". In: *IEEE Access* 8 (2020). 10.1109/ACCESS.2020.2977558, pp. 46735–46752.

[KV18] R. Karásek and F. Vejražka. "The DVB-T-based positioning system and single frequency network offset estimation". In: *Radioengineering* 27.4 (Sept. 2018). 10.13164/re.2018.1155, pp. 1155–1165.

[KP15] R. Karásek and J. Popp. "Broadband signals using for position estimation". In: *Proceedings of the 19th International Scientific Student Conference POSTER 2015*. Czech Technical University in Prague, May 2015. ISBN: 978-80-01-05728-5.

[NK16] V. Navrátil and R. Karásek. "Position estimate using DVB-T signal". In: *Proceedings of the 20th International Scientific Student Conference POSTER 2016*. Czech Technical University in Prague, May 2016. ISBN: 978-80-01-05950-0.

[NKV16] V. Navrátil, R. Karásek, and F. Vejražka. "Position estimate using radio signals from terrestrial sources". In: *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 10.1109/plans.2016.7479775. IEEE, Apr. 2016.

[Gen+18] C. Gentner et al. "Simultaneous localization of a receiver and mapping of multipath generating geometry in indoor environments". In: *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*. 10.33012/2018.16043. Institute of Navigation, Oct. 2018.

[GKS19] C. Gentner, R. Karásek, and M. Schmidhammer. "Crowd sourced pedestrian dead reckoning and mapping of indoor environments using smartphones". In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. Institute of Navigation, Oct. 2019.

[KG21] R. Karásek and C. Gentner. "Gaussian mixture model learning for multipath assisted positioning". In: *2021 15th European Conference on Antennas and Propagation (EuCAP)*. IEEE, Mar. 2021.

All authors contributed equally unless otherwise stated.