Lübeck, Germany
11-12 July 2019

Proceedings of the

4th KuVS/GI Expert Talk on Localization

ITG VDE

GI

Technische Universität Braunschweig

TECHNISCHE HOCHSCHULE LÜBECK

TECHNISCHE UNIVERSITÄT DRESDEN

HCU HafenCity Universität Hamburg

CoSA Center of Excellence

Editors: Marco Cimdins, Horst Hellbrück

Cover Picture: The cover was prepared by Marco Cimdins based on a photograph by Jens Thiedke.

Printed in Germany

https://doi.org/10.24355/dbbs.084-201907151114-0

Message from the Editors

Localization is a key technology in the field of medical, industrial and logistics applications, thus many applications will benefit from localization. Although research interest and solutions are evolving, there are still technical solutions missing for many use cases of localization. The program of the 4th Expert Talk on Localization covers a variety of different aspects from technologies such as FMCW radar and LiDAR, to smartphone and ultra-wideband based systems, and localization hardware.

The 4th Expert Talk on Localization offers a platform to discuss recent results of research and development work and share ideas and opinions. The event brings together researchers from academia and industry and offers a broad range of contributions in this challenging area including complete localization systems and demonstrations.

We thank all authors for their contribution to the program. We express our appreciation to Technische Hochschule Lübeck for its support, Center of Excellence CoSA for the organization of the meeting. We thank HafenCity University, University of Braunschweig and Technische Universität Dresden for their continuing contribution to the program as well as GI and KuVS for facilitating this event.

July 2019 M Cimdins & H Hellbrück

https://doi.org/10.24355/dbbs.084-201907151114-0

General Chair

Horst Hellbrück, Technische Hochschule Lübeck

Program Committee

Marco Cimdins, Technische Hochschule Lübeck
Marco Gunia, Technische Universität Dresden
Yannic Schröder, University of Braunschweig
Janek Stoeck, HafenCity University

Organization

Marco Cimdins, Technische Hochschule Lübeck

https://doi.org/10.24355/dbbs.084-201907151114-0

Table of Contents

Contributions

Session 1: Radar and LiDAR

Session 2: Positioning Hardware

Session 3: Smartphone and Ultra-Wide Band Positioning

https://doi.org/10.24355/dbbs.084-201907151114-0

A Practical Toolbox for Getting Started with mmWave FMCW Radar Sensors

Manfred Constapel*, Marco Cimdins* and Horst Hellbrück*†
* Technische Hochschule Lübeck, Germany
Department of Electrical Engineering and Computer Science
Email: {manfred.constapel, marco.cimdins, horst.hellbrueck}@th-luebeck.de
† University of Luebeck, Germany,
Institute of Telematics

***Abstract*—In this paper, we sum up our experience gathered working with mmWave FMCW radar sensors for localization problems. We give a glimpse of the foundations of radar that is necessary to understand the benefit and advantages of this technology. Moreover, we introduce our open-source software toolbox *pymmw* based on Python for Texas Instruments IWR1443 ES2.0 EVM sensors to provide students and researchers easy access to those radar sensors. In doing so, one can jump right into sensing with mmWave FMCW radar from a practical point of view and start doing experiments and developing own applications. Finally, *pymmw* is used for data acquisition of a scene illuminated by three virtual radars in three different states of occupancy showing the potential of mmWave FMCW radar for indoor and distance-based localization applications.**

***Index Terms*—radar, sensing, fmcw, mmwave, localization**

I. Introduction & Radar Principles

In this paper, we introduce the open-source software toolbox *pymmw* in order to get the reader started with mmWave FMCW (frequency modulated continuous wave) radars for indoor localization. Recently, commercial off-the-shelf mmWave FMCW radar sensors are available for less than 500 Euro, therefore become interesting for a wide range of applications such as indoor localization and tracking.

mmWave is the band of spectrum between 30 GHz and 300 GHz. This technology is insensitive against environmental influences such as smoke, fog, rain, bad light, and extreme temperatures. FMCW mmWave radar can do range measurements with high accuracy (less than 1 mm) and detect very fine motions while it can penetrate through materials like plastic, fabric, and drywall.

Commonly FMCW radars use some form of linear frequency modulation, e.g. sawtooth. The continuous transmission of the signal often is organized into loops (or packets). Each loop consists of a series of linearly frequency modulated fragments, called chirps, which swipe the bandwidth B in the time interval T, which is often referred to as the Coherent Processing Interval (CPI), with slope m as shown in Figure 1. The transmitted chirp (TX) gets reflected off a target, and a time-delayed version of the chirp (RX) is received by the radar. The round-trip time of the chirp corresponds to the distance to the target, which can not be measured directly, instead, the received chirp is mixed, hence multiplied, with the signal being transmitted yielding the frequency difference f_B. The frequency difference f_B over time constitutes a frequency tone, which reveals after transformation to the frequency domain by Discrete Fourier Transform (DFT), e.g. utilizing the Fast Fourier Transform (FFT), a beat frequency, thus a peak in the frequency spectrum. The peak corresponds to the distance of the target at a given maximum range R_{max}.

For the sake of simplicity, and due to the focus on distance-based localization, this paper is confined to distance estimation only. Thus, velocity and angle estimations are not considered, which can be assumed being subsequent processing steps of range estimation. Nevertheless, mmWave FMCW radar is in general very capable of doing high-precision radial velocity measurements by exposing Doppler shifts [1], and, utilizing and virtually combining an array of RX and TX antennas in a MIMO configuration [2], able to estimate the angle of arrival in elevation and azimuth via beamforming techniques.

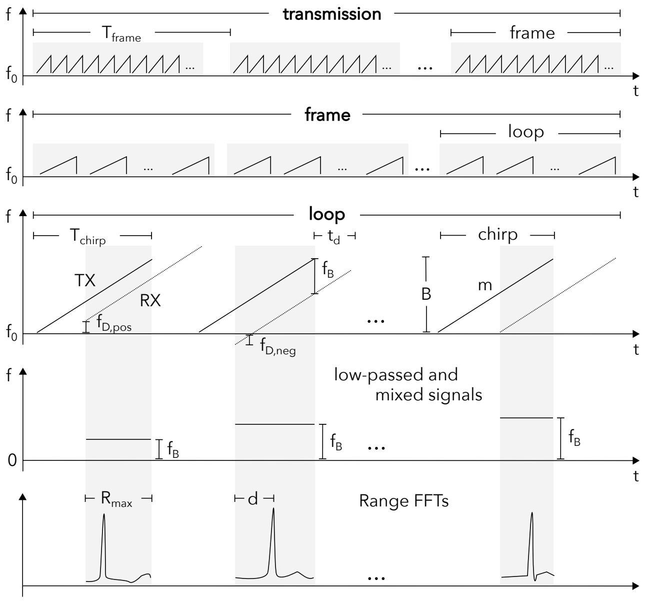


Fig. 1: Relationship between frames, loops, and chirps (upper half) and processing steps commonly used in FMCW radar (lower half)

https://doi.org/10.24355/dbbs.084-201907151114-0

II. Related work & Further Reading

This section provides a few references for further reading and more in-depth information for research.

[3] gives a classical overview of the radar data processing pipeline. It covers signal models, sampling, waveforms, Doppler and detection processing as well as beamforming. [4] reveals details about the techniques used for radar systems with distributed antennas, hence defining MIMO as a distributed system, from different angles of view: localization, adaptive signal design, and space-time coding. [5] provides a comprehensive theoretical and analytical background in reasonable depth to modern radars. It starts with different variants of the radar equation and ends up with several loss factors that should be kept into account while estimating ranges with radar. [6] focuses on signal processing, in particular on the difference frequency signal (DFS) for short-range FMCW applications in industrial environments. [7] describes methods for filtering, e.g. with Kalman filters, and tracking of multiple moving and maneuvering targets one would expect to see in air surveillance applications. Furthermore, it evaluates aspects of the radar data processing pipeline and introduces simulation concepts for radar data processing. [8] captures a large variety of indoor monitoring applications for daily living utilizing FMCW radar and CSI- or RSSI-based DFL systems. It focuses on monitoring and classification of motion activities of humans at home for elderly monitoring or in hospitals for vital signs monitoring. [9] describes the principle of a circuit for a constant false alarm rate (CFAR). CFAR schemes vary the detection threshold as a function of the sensed environment to detect targets.

III. Getting Started

In this section, the ingredients needed to immediately jump into experimenting with mmWave FMCW radar sensors are briefly introduced.

1) mmWave Sensor: In essence, the IWR1443BOOST evaluation board from Texas Instruments [10] provides everything needed to start developing radar applications on a low-power ARM Cortex-R4F processor. It provides plenty of interfaces as well as onboard emulation for programming and debugging. The device got a small form factor and operates at 76 GHz to 81 GHz while having low power requirements - consumption is regularly less than 0.5A of DC current. It can get up to, depending on the RF frontend configuration, 12 virtual antennas (3 TX × 4 RX) for aperture, which provides a reasonable resolution in azimuth and elevation for various indoor monitoring and tracking applications.

2) mmWave SDK: The mmWave software development kits (mmWave SDKs) contain firmware for the Radar Subsystem (RadarSS or BSS) and Master Subsystem (MSS) - called Labs or Demos respectively - of supported mmWave FMCW radar devices from Texas Instruments. mmWave SDKs come in two flavors supporting different devices: IWR for industrial applications and AWR for automotive applications. They are updated frequently, while the number of contained Demos and Labs increases, and more and more mmWave radar sensor EVMs become supported.

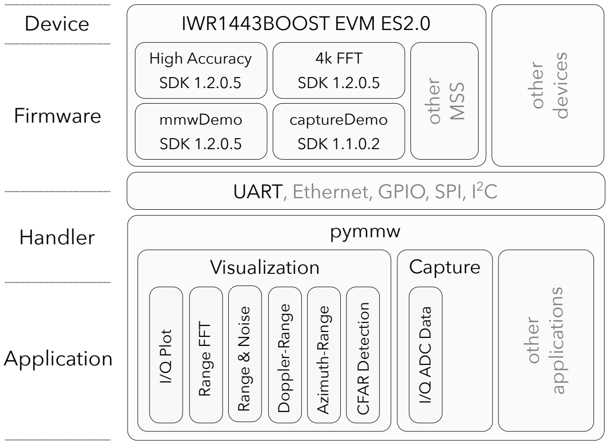


Fig. 2: Building blocks of *pymmw*

3) mmWave Toolbox: pymmw[1] acts as a host for various applications, e.g. I/Q Plot and Range FFT, which can be categorized into plots (visualization) and data aquisition (capture). For both, visualization and capturing applications, data is captured exclusively via two UART channels from a supported MSS except for the *captureDemo*, which performs a direct L3 memory read via the SPI, GPIO and RS233 capabilities of a FTDI chip of the onboard XDS110 debugger utilizing the Debug Server Scripting (DSS) library bundled in Code Composer Studio (CCS). In *pymmw* - at this point in time - four MSS for the Texas Instruments IWR1443 ES2.0 evaluation board from SDK 1.2.0.5 and SDK 1.1.0.2 are supported. Applications in *pymmw* are imported dynamically, and while running multiple applications in parallel, they are executed in different processes using pipes for interprocess communication (IPC). Hence, being in a separate process, an application does not affect other applications in execution to a great extent, i.e. complex post-processing can be done alongside a visualization with costly rendering threads in pseudo-realtime.

IV. Example Application

In this section, the potential of the mmWave radar in a simple indoor localization scenario is briefly shown by an example application, aiming to detect and localize targets within an area of interest or scene respectively. The example is inspired by an application that is driven by a device-free-localization system [11] developed in our institute.

Three mmWave radars [10] are placed in a triangular pattern at 1.37 m in height in an indoor corridor as shown in Figure 3. The task is to observe the area of interest in order to sense the presence of a target and estimate its position.

Approximately 500 measurements, i.e. Range FFTs, are taken with *pymmw* from each radar fixed at locations R_1, R_2 and R_3 illuminating the scene at three different states of occupancy: 1) idle, hence no target is located in the scene, 2) an octahedral corner reflector - composed of 12 isosceles triangles covered

[1] M. Constapel, ”Pythonic mmWave Toolbox for TI’s IWR Radar Sensors”, https://github.com/m6c7l/pymmw, 2019.

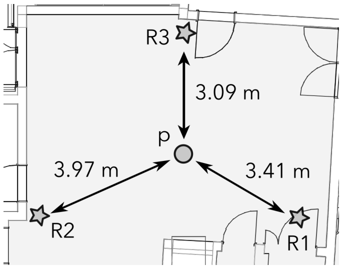


Fig. 3: Indoor measurement setup with radar locations R_1 to R_3 and corresponding distances d_1 to d_3 to target location p

with aluminum foil, 7 cm in length at side edges - is placed at the position p, and 3) an adult is standing at the position p. The radars are equipped with the mmwDemo MSS firmware of SDK 1.2.0.5 and are configured to transmit in temporally separated slots to avoid interference with other radars.

The most important RF configuration parameters applied to all radars are depict in Table I.

TABLE I: Radar RF configuration for scene illumination

Parameter	Value	Parameter	Value
f_0	77 GHz	P_{frame}	100 ms
m	50 MHz/μs	$\#chirps/frame$	32
B	3.6 GHz	$\#loops/frame$	16
$\#RX$	4	$\#samples/chirp$	144
$\#TX_\phi$	2	R_{max}	480 cm
$\#TX_\theta$	0	ΔR	4.2 cm

From the measurements of the idle state the mean value $\overline{r}_{\text{idle}}$ is estimated in order to remove clutter in the measurements of the other states having a target, either a reflector or an adult, placed the scene. For the detection of both, reflector and adult, a very simple peak detection algorithm is used that determines the absolute maximum in the Range FFT of a given measurement series. The principle idea is to calculate the difference Δr between the idle state and observation $\Delta r = r_{\text{target}} - \overline{r}_{\text{idle}}$ and find the peak that correlates to the obstacles $\hat{r} = \arg\max(\Delta r)$.

After the peak detection, the Euclidean distance between the ground truth distance r and the distance at the peak of the difference $\hat{r}$ is calculated in order to estimate μ_{error} and σ_{error}.

TABLE II: Results regarding reliability of the simple peak detection algorithm in the Range FFTs

	Reflector			Adult		
	R1	R2	R3	R1	R2	R3
μ_{error} [cm]	0.02	4.32	0.19	7.67	22.65	78.70
σ_{error} [cm]	0.00	0.00	0.00	3.75	66.70	110.21

The results are shown in Table II. The mean as well as the standard deviation for the reflector are less than 4 cm, which is a great result for device-free indoor localization systems. The result for the adult shows the potential of mmWave radar for localization by an offset of approximately 8 cm and a deviation of about 4 cm for R_1. The results for R_2 and R_3 are worse, however, this is quiete likely due to the simple peak detection algorithm we employed averaging noisy peaks which are not related to the adult. Furthermore, in comparison to the adult, the reflector provides a huge increase in the effective radar cross section (RCS) at a relatively small space. Therefore, considerable amounts of energy get reflected off from a virtually single point, thus the received signal is less noisy.

V. Conclusion & Future Work

This paper provide a quick start guide for mmWave FMCW radar sensors. It illustrates details about the foundations of radar that is necessary for creating distance-based localization applications. The open-source software toolbox *pymmw* based on Python provides students and researchers easy access to Texas Instruments IWR1443 ES2.0 EVM sensors and enables them getting started immediately. To demonstrate the ease of use and to show the potential of mmWave FMCW radar for indoor localization applications, *pymmw* is used in an example application involving three radars and three states of occupancy for capturing data for post-processing purpose.

In the future, support for more mmWave radar sensors and the attachable DCA1000EVM capture card will be added, to enable easy high-speed capturing of radar cube data (RDC) in pseudo-realtime via Ethernet with *pymmw* in order to employ machine learning techniques and methods of artificial intelligence for more advanced detection algorithms and filters.

Acknowledgments

This publication is a result of the research within the Center of Excellence CoSA.

References

[1] C. Iovescu and S. Rao, “The fundamentals of millimeter wave sensors,” *Texas Instruments, SPYY005*, 2017.

[2] S. Rao, “Mimo radar,” *Texas Instruments Application Report*, pp. 1–12, 2017.

[3] M. A. Richards, *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.

[4] J. Li and P. Stoica, *MIMO radar signal processing*. Wiley Online Library, 2009, vol. 7.

[5] D. K. Barton, *Radar equations for modern radar*. Artech House, 2013.

[6] B. A. Atayants, V. M. Davydochkin, V. V. Ezerskiy, V. S. Parshin, and S. M. Smolskiy, *Precision FMCW short-range radar for industrial applications*. Artech House, 2014.

[7] H. You, X. Jianjuan, and G. Xin, *Radar data processing with applications*. John Wiley & Sons, 2016.

[8] M. Amin, *Radar for indoor monitoring: Detection, classification, and assessment*. CRC Press, 2017.

[9] H. Finn, “Adaptive detection mode with threshold control as a function of spatially sampled clutter-level estimates,” *RCA Rev.*, vol. 29, pp. 414–465, 1968.

[10] *IWR1443BOOST Evaluation Module mmWave Sensing Solution: User’s Guide*, Texas Instruments, 4 2017, revised April 2018.

[11] M. Cimdins, M. Pelka, and H. Hellbrück, “Sundew: Design and evaluation of a model-based device-free localization system,” in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sep. 2018, pp. 1–8.

Architecture to Detect, Track, and Classify Objects using LiDAR Measurements in Highway Scenarios

Mathias Pelka*
*Ibeo Automotive Systems GmbH, Germany
Reference and Tools
Email: mathias.pelka@ibeo-as.com

***Abstract*—Self-driving cars require a holistic perception of their environment. To achieve this requirement, a plethora of sensor technologies exists e.g. RGB-camera, ultra-sonic and radar. Those sensor technologies have different range, as well as resolution and behave differently with varying weather conditions. Another technology is *Light Detection and Ranging (LiDAR)*, which enables precise distance measurements. In combination with RGB-cameras, ultra-sonic, and radar, LiDAR closes the gap to enable the holistic perception of the environment.**

Due to limited experience with LiDAR sensors, there is a lack of understanding how to detect, track, and classify objects (e.g. cars, guardrails) using LiDAR data. In this paper, we propose an architecture to detect, track, and classify objects based on LiDAR measurements in highway scenarios. We evaluate our architecture using preliminary sensor data obtained from a setup including six Ibeo Lux sensors and additional a roof mounted Velodyne HDL-64E.

***Index Terms*—LiDAR, object tracking, self-driving cars, autonomous cars.**

I. Introduction and Related Work

Light Detection and Ranging (LiDAR) originated shortly after the invention of the LASER. Early applications includes meteorology or surveying [1]. LiDAR allows accurate distance measurements (e.g. accuracy is below 0.1 m for the Ibeo Lux) between a LiDAR emitter and a reflector [2]. Multiple LiDAR measurements shape a point cloud, as shown in Figure 1. The figure shows a *scan* from six Ibeo Lux LiDAR sensors as well as a scan of a Velodyne HDL-64E mounted on a car (VW Passat) in a highway scenario [2], [3]. For simplicity, the figure shows the region of interest in front of the car. Some structures are recognizable. We aim to detect, track, and classify those structures.

The scan is usually centered around the *ego vehicle* which contains the recording devices. The six Lux sensors are mounted around the bumper of the car and provide 360 degrees field of vision. In contrast to radar, LiDAR provides a higher-density point cloud, however it is more susceptible to weather. Compared to a RGB-camera, LiDAR provides also the distance towards an object.

Recent advances allow LiDAR to be used in cars, indicating a mature sensor technology. Liu et al. [4] described the measurement principle and provided a survey of different LiDAR sensors in. Object tracking using LiDAR was discussed by Fürstenberg in [5].

Object tracking for automotive applications is usually divided into two categories: *online* and *offline*. Online processing is employed for example in high level driving functions including emergency brake assist or adaptive cruise control. Additionally, online processing serves as an input for self-driving cars. Such applications require real-time processing, consequently limiting the complexity of the algorithms. An Ibeo Lux LiDAR sensor has an update rate of 25 Hz, resulting in a required processing time smaller than 40 ms [2]. Furthermore, often more than one LiDAR sensor is employed, leading to large amounts of data. In such cases, search algorithms are expensive and real-time capable tracking algorithms are required.

Offline processing is usually used to verify the performance of online algorithms. This is called *reference*. Offline processing requires large amount of data and does not need real-time capabilities. In offline processing, advanced processing is possible. This, for instance, includes tracking objects backward in time, particularly useful when only sparse LiDAR point information is available.

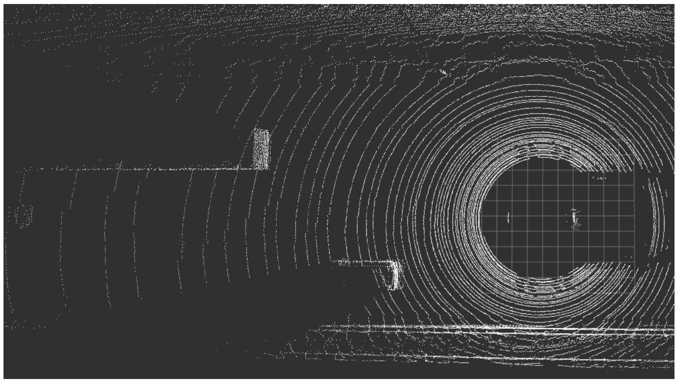


Fig. 1: Point cloud with six Ibeo Lux sensors and a roof mounted Velodyne HDL-64E. The region of interest is in front of the car.

In this paper, we discuss the relevant steps to detect, track, and classify objects based on LiDAR point cloud data in an online application for a highway scenario. The steps include ground detection, clustering, detecting best seeds (starting points for the tracking), association and Kalman-Filtering, ego motion compensation, and classification. This paper does not explain further concepts like lane detection or path planning.

The paper is structured as follows: We propose an architecture in Section II and discuss the parts of the architecture.

https://doi.org/10.24355/dbbs.084-201907151114-0

We show preliminary evaluation results in Section III using real-world data. Finally, we summarize our work in Section IV.

II. Architecture

In this section we present an architecture to detect, track, and classify objects in a highway scenario. The input for this architecture is the LiDAR point cloud, and the output are tracked objects. The architecture is shown in Figure 2 and is divided roughly in preprocessing and the tracking part.

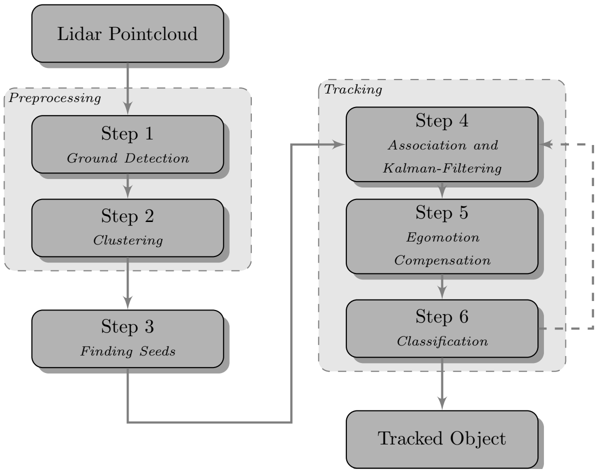


Fig. 2: Architecture to detect, track, and classify objects.

In the following subsections, we explain each block in detail.

A. Ground Detection

The ground detection is the first step in LiDAR point cloud processing and part of the preprocessing. Different techniques for ground detection are available. Ground detection labels all ground points in the point cloud as ground and consequently removes them from further processing. Ground points are included in almost every LiDAR point cloud, since they constitute the dominant plane in highway scenarios.

A simple and computationally expensive method is fitting a plane with a principle component analysis of all LiDAR points. However, this approach is not very robust, because objects, including cars or trees, will disturb the calculation. In order to remedy this, another approach is applied, where local samples of LiDAR points are taken, a plane fitted, and the plane normal vector is stored. By repeating this process and choosing different points, we determine the dominant normal vector, which is the usually the ground plane. This requires a lot of points in to be ground points. While in highway scenarios this is usually the case, in city scenarios or parking garages this may pose a problem. Here, the assumption may not be valid any longer.

In a typical scan, roughly 50 % of all points are part of the ground plane, greatly reducing the amount of data.

B. Clustering

The next step is to determine isolated objects in the point cloud. This is done using *clustering* or *segmentation* algorithms. Computation of clusters involves a distance function, which provides a measure of how far LiDAR points are apart of each other.

Standard clustering algorithms are employed for this step, including Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The algorithm finds isolated clusters and requires no prior information, e.g. how many clusters are in the point cloud. Other properties of DBSCAN includes almost determinism, meaning it is independent on the processing order of LiDAR points. Furthermore, DBSCAN handles a variety of distance functions (e.g. Euclidean Distance, Manhattan Distance) to compute the distance between LiDAR points and DBSCAN has linear complexity and is therefore suited for online applications in self-driving cars [6].

The combination of ground detection and clustering is usually called preprocessing. Both are required in order to detect, track, and classify object. The result of ground detection and clustering is shown in Figure 3. Each cluster has a different color. The largest cluster is the ground which contains roughly 50 % of all LiDAR points.

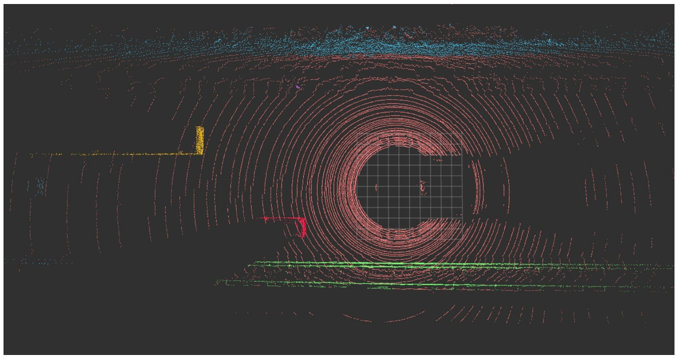


Fig. 3: Results of the preprocessing including ground detection and clustering. Each cluster has a different color. The largest cluster is the ground plane (shown in light red).

In Figure 3 three structures are visible, similar to Figure 1. A large object in front of the ego vehicle and two smaller objects, the structures have been correctly clustered.

C. Finding Seeds

The next step is to detect suitable seeds from the clusters. Those seeds are the starting point for the tracking. A cluster is not per se a good starting point for tracking and heuristics show that remote objects only consists of very few LiDAR data points. Those remote objects are hard to detect again, indicating bad seeds respectively clusters. Objects close to the ego vehicle are preferred; however, we like to detect objects not only close to the vehicle but within a certain distance. Further, clusters that are separated (i.e. clusters that do not have other clusters in their vicinity) are also preferred as starting points. The reason is, that clusters in a crowded vicinity (i.e. close to the ego vehicle) tend to occlude other objects. Consequently, we compute for each cluster a score, called *cluster quality*.

The higher the score, the more we are sure, that the cluster is a good seed.

If a seed is found to be suitable, we start tracking and create a *tracked object*. For each tracked object, we store the position, velocity, acceleration, and the heading.

D. Association and Kalman-Filtering

In the first tracking step of the tracked object, we aim to find clusters in the LiDAR point cloud, that match the tracked object. For that, we first predict the tracked object using a motion model (e.g. free mass with constant acceleration or a unicycle model) to the timestamp of the next LiDAR point cloud. We then associate a cluster which is in the vicinity of that tracked object and compute a measurement from it. Based on the prediction, we aim to predict the expected measurement and compare it with the actual measurement during the update step. For this, we employ an *unscented Kalman-Filter (UKF)* which is able to track non-linear movement of objects [7], [8].

In our implementation, the UKF also tracks the size of the object represented as a bounding box. Additionally, we store the covariance of the state.

E. Egomotion compensation

Since the tracked object moves relative to the ego vehicle, the motion of the ego vehicle (i.e., egomotion) has to be compensated [5]. For this, the complete ego motion between two points in time is required. The ego motion stores the change of translation and rotation. The change in position is a vector $\Delta\mathbf{r} = [\Delta x, \Delta y, \Delta z]^T$. The rotation matrix is denoted as $\mathbf{R}$. If the tracked object position is $\mathbf{r}$, the ego motion compensation is written as

$$\mathbf{r}_{\text{new}} = \mathbf{R}\left(\mathbf{r}_{\text{old}} + \Delta\mathbf{r}\right). \tag{1}$$

F. Classification

In order to optimally track an object, some additional information has proven beneficial. For instance, a car moves differently than a pedestrian. A pedestrian moves more in the sense of a free mass, rather than a car, which has certain restrictions in terms of motion. For instance, a car cannot turn on the spot, while a pedestrian can.

Different approaches for classification exist. Neural networks are popular, however, they require intensive training. Another approach based on Dempster-Shafer belief propagation was proposed by Magnier et al. [9]. A very simple approach is a decision tree, which is able to classify object, using velocity and the size of the bounding box.

After classification, the tracking process continues until no further associations have been found. In such cases, the UKF predicts the object while a confidence metric decreases. If the confidence becomes too low, the object tracking stops and the tracked object is stored.

III. Preliminary Results

We show preliminary results in this section. The data was recorded using a VW Passat with six Ibeo Lux LiDAR sensors mounted around the bumper and a Velodyne HDL-64E mounted on the roof. A visualization of the data is shown in Figure 4. It shows a similar scenario as shown in Figure 1 and Figure 3. This time, objects have been tracked (a truck and two cars) in front of the ego vehicle.

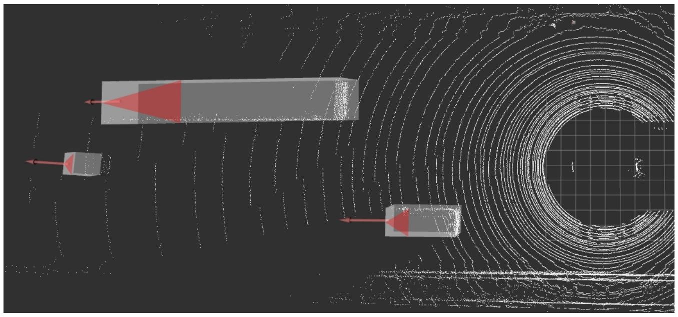


Fig. 4: Visualiazation of preliminary results

As this is still a work in progress, no *key performance indicators (KPI)* have been calculated. Examples for KPIs include accuracy, precision and the elements of a confusion matrix.

IV. Conclusion

In this paper, we proposed an architecture to detect, track, and classify objects using LiDAR measurement in a highway scenario. We have discussed the required steps to detect an object, using ground detection and clustering and discussed an approach for object tracking using Kalman filtering in combination with a decision tree to classify objects.

In future work we continue to develop this approach to enable reference object tracking. Furthermore, we aim to include key performance indicators (KPI) calculation and compare this approach against other state-of-the-art algorithms. This enables other researchers to evaluate their algorithms and to calculate performance metrics or key performance indicators.

References

[1] G. G. Goyer and R. Watson, "The laser and its application to meteorology," *Bulletin of the American Meteorological Society*, 1963.

[2] "Reference Sensor System Ibeo Automotive Systems GmbH, Hamburg," https://www.ibeo-as.com/ibeoreference/reference-sensor-system/, (Accessed on 05/21/2019).

[3] "HDL-64E," https://velodynelidar.com/hdl-64e.html, (Accessed on 05/21/2019).

[4] J. Liu, Q. Sun, Z. Fan, and Y. Jia, "TOF Lidar Development in Autonomous Vehicle," in *3rd Optoelectronics Global Conference*. IEEE, 2018.

[5] K. C. Fürstenberg, *Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner*. Universität Ulm Fakultät f. Ingenieurwissenschaften u. Informatik, 2009.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Conference on Knowledge Discovery and Data Mining*, 1996.

[7] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*. IEEE, 2000.

[8] M. Pelka and H. Hellbrück, "Introduction, discussion and evaluation of recursive Bayesian filters for linear and nonlinear filtering problems in indoor localization," in *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2016.

[9] V. Magnier, D. Gruyer, and J. Godelle, "Automotive LIDAR objects detection and classification algorithm using the belief theory," in *Intelligent Vehicles Symposium*. IEEE, 2017, pp. 746–751.

https://doi.org/10.24355/dbbs.084-201907151114-0

https://doi.org/10.24355/dbbs.084-201907151114-0

A Low-Cost GNSS Repeater for Indoor Operation

Yannic Schröder and Lars Wolf
Institute of Operating Systems and Computer Networks
Technische Universität Braunschweig
Braunschweig, Germany
Email: [schroeder|wolf]@ibr.cs.tu-bs.de

***Abstract*—GNSS hardware can only be used outdoors as reception is strongly attenuated by buildings. We present a cost-effective way to build a GNSS repeater that can be used to test GNSS hardware indoors. Its design comprises two GNSS antennas and four extra passive components. We evaluate the performance of the circuit and show that it can be used to relay GPS, GLONASS, and Galileo signals to a GNSS receiver with its antenna being indoors, close to the GNSS repeater.**

I. Introduction

Global Navigation Satellite Systems (GNSSs) can hardly be used indoors, as the required satellite signals are attenuated by the building structure. Especially when implementing applications that need a GNSS signal to function properly, this becomes an obstacle for developers. There are two viable options: to test and debug the application outdoors or attach a GNSS antenna with a long cable, so that the antenna can be placed outdoors while the device remains indoors as depicted in Figure 1 (top). However, this is not possible for all devices. Highly integrated electronics like smartphones or wireless sensor nodes might not allow to attach an external antenna.

Vector signal generators can be used to generate arbitrary GNSS signals to simulate arbitrary locations. Such devices however are very expensive. This cost can be reduced by using a Software Defined Radio (SDR). For development it is often sufficient to use the actual signals available at the developer's location. These signals only need to be available indoors by relaying them. Commercial GNSS repeaters are available that allow relaying of GNSS signals. They can be used as shown in Figure 1 (bottom) and can be used with devices, that cannot be attached to the outdoor antenna directly.

In this paper, we show how we implemented a simple and low-cost GNSS repeater that picks up the signals outdoors via one antenna and reradiates the signal indoors via a second antenna. It can be build from readily available low-cost components and requires no particular knowledge about high frequency circuit design.

II. Related Work

GNSS repeaters are used to enable the use of GNSS devices for Indoor Positioning Systems (IPSs). To facilitate this, multiple repeaters are placed at different locations, reradiating satellite signals. The signals of these so called pseudolites are used by a mobile device for localization. However, special algorithms or modifications to the GNSS receiver are needed for successful localization.

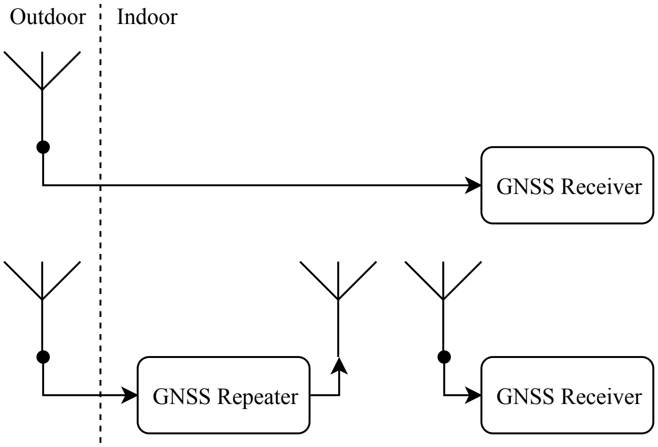


Figure 1. System overview. Top: *Direct* connection of the GNSS receiver to an outdoor antenna. Bottom: The GNSS *repeater* is attached to an outdoor antenna and relays the signal to the GNSS receiver.

Jardak and Samama present an IPS based on Global Positioning System (GPS) repeaters [1]. They use a single external antenna to relay the GPS signal to multiple indoor antennas that reradiate the signal. To avoid interference between the different antennas, they transmit in a round-robin schedule. To allow localization, the GPS receiver needs to be modified. However, as their system is only tested in simulations, the paper does not disclose how the GPS repeater would be implemented.

Riwa et al. use pseudolites that reradiate a GPS signal to locate a robot indoors [2]. To facilitate localization of the robot, a special GPS algorithm is implemented as SDR. A commercial GPS repeater is used in their experiments.

Özsoy et al. present a 2D IPS that uses multiple GPS repeaters [3]. The system uses a standard GPS receiver. However, its received raw data is processed on a computer to allow localization. As the repeaters will introduce additional delays to the signals, the positioning engine inside the GPS receiver returns erroneous results. They use specially designed directional antennas to radiate the GPS signal indoors. The antennas are attached to specially designed Low Noise Amplifiers (LNAs).

https://doi.org/10.24355/dbbs.084-201907151114-0

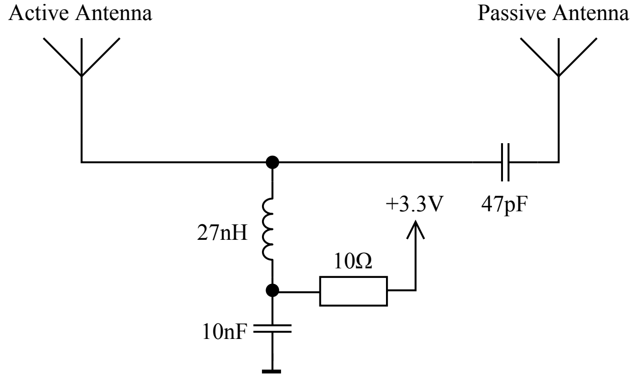


Figure 2. Circuit diagram of the GNSS repeater.

III. Hardware Design

The hardware of our GNSS repeater design mainly consists of two antennas. One receives the signals outdoors, while the other reradiates the signals indoors. The outdoor antenna is active. It contains an LNA, which amplifies the received signal. This ensures that the signal is strong enough for another (active) GNSS antenna to pick it up after being reradiated by the passive antenna, see Figure 1. It is crucial to use a passive antenna at the output of the repeater, as the LNA in an active antenna would not allow transmission of a signal.

To amplify the incoming signal, the LNA needs to be powered by a DC power source. Power is supplied directly via the coaxial cable. Most consumer GNSS antennas are actively powered by the attached receiver. Here we use an ANN-MS-0-005-0 [4] antenna by u-blox. According to its data sheet it requires 2.7-5.5 V operating voltage. The main purpose of the circuit shown in Figure 2 is to supply this operating voltage. Further, it blocks the DC voltage from entering the passive antenna that reradiates the signal via a decoupling capacitor. This is needed, as some passive antennas are designed to be a short circuit for DC signals.

The circuit is adopted from the u-blox hardware integration manual [5]. Instead of attaching a GNSS receiver directly, a second antenna reradiates the received signal.

IV. Evaluation

The performance of the GNSS repeater is evaluated regarding different performance metrics and compared with a reference setup. We evaluate the Time to First Fix (TTFF), Carrier-to-Noise-Density Ratio (C/N_0) and number of satellites used. As GNSS receiver we use a u-blox NEO-M8Q [6] connected via USB to a computer. It is configured to use GPS, GLONASS and Galileo satellites and reports its status at a frequency of 1 Hz.

A. Experimental Setup

We used two experimental setups to evaluate the repeater's performance, as shown in Figure 1. In both setups, the active GNSS antenna was positioned outside a window of our office building. Its integrated 5 m coaxial cable were run through the frame of the closed window into the building. The antenna had only limited Line-of-Sight (LOS) to the sky, as multiple buildings blocked the lower elevation angles and our own building blocked half the sky. All measurements were obtained in overcast weather conditions without rain.

Table I
Time to First Fix

	min. [s]	max. [s]	median [s]
with repeater	22	68	33
direct connection	28	89	36.5

In the first setup, we connected the GNSS receiver directly to the outdoor antenna. This setup is referred to as the *direct* setup. It serves as reference for the performance of the GNSS receiver in the evaluation environment.

In the second setup, we placed the (active) antenna of the GNSS receiver on a desk with 1 m distance to the next window. The receiver was again connected via 5 m of coaxial cable. It was unable get a fix, as the GNSS signals were attenuated by the building. The sending antenna of the repeater was placed in close proximity (5 cm) to the antenna of the receiver. The outdoor antenna was connected to the repeater. This setup is referred to as the *repeater* setup as shown in Figure 1 (bottom).

B. Time to First Fix

We compared the TTFF of both setups. We measured the time between the issuing of a cold start command to the GNSS receiver and the first 3D fix for GPS. This experiment was repeated 10 times in both setups. Table I shows the results. The median TTFF is comparable for both setups. Occasionally, the GNSS receiver requires more time to get a fix. However, this is not caused by the repeater device.

C. Receiver Performance over Time

We sampled GNSS data from the receiver for one hour with both setups as satellite positions and thus performance vary over time. This results in 3600 samples for each setup.

Figure 3 shows that the number of satellites used for positioning by the GNSS receiver is similar in both setups. The number of satellites increases in the first 15 minutes, as the receiver learns more satellite positions and starts receiving on the respective channels. We were able to receive signals from all tested GNSS systems (GPS, GLONASS, and Galileo) in both setups.

We also investigated the Carrier-to-Noise-Density Ratio (C/N_0) of both setups. As the repeater introduces a more complex signal path and additional amplification, it might also introduce additional noise to the system which might reduce performance. Figure 4 shows the C/N_0 over time for both systems. Again, the performance for both setups is comparable.

https://doi.org/10.24355/dbbs.084-201907151114-0

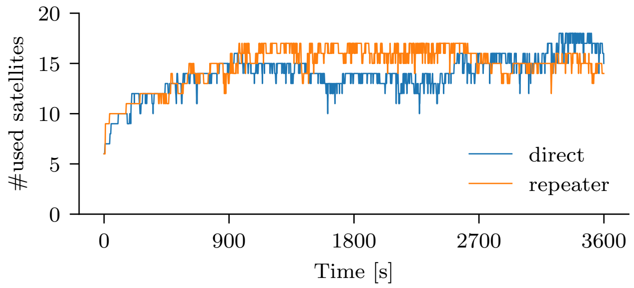


Figure 3. Number of used satellites over time.

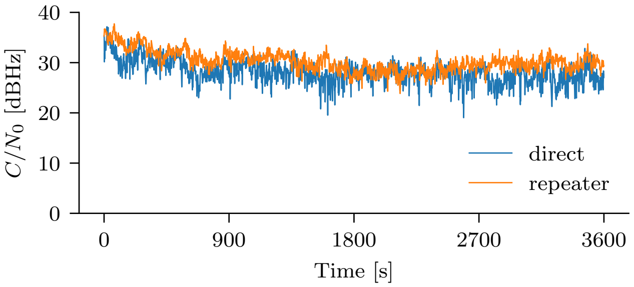


Figure 4. Carrier-to-Noise-Density Ratio (C/N_0) over time.

V. Limitations

As the GNSS signal is repeated from an outdoor antenna, the location inside the building cannot be determined with this setup. Instead, the location of the outdoor antenna is reported, independently of the position of the GNSS receiver. Further, as the signal path is longer, the time information will be late. However, as the signal is not digitally processed, but only forwarded from one antenna to the other in the analog domain, this time offset is negligible.

The maximum distance between the GNSS repeater and the receiver's antenna is limited. In our setup, the receiver was able to get a fix at up to 50 cm distance between the passive antenna of the repeater and the antenna of the GNSS receiver. This distance is limited by the amplification of the active outdoor antenna and losses in the repeater hardware and coaxial cables.

VI. Conclusion

We presented a low-cost GNSS repeater that can be built with readily available components. It does not require special parts other than one active and one passive GNSS antenna. Our evaluation shows that performance is comparable to using the outdoor antenna directly with a GNSS receiver. The repeater can be used to supply GNSS signals to devices inside buildings, where satellite reception is otherwise not possible. This allows to test and evaluate applications indoors which need a GNSS signal for proper operation.

References

[1] N. Jardak and N. Samama, "Indoor Positioning Based on GPS-Repeaters: Performance Enhancement using an Open Code Loop Architecture," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 347–359, Jan. 2009.

[2] Haruhiko Niwa, Kenri Kodaka, Yoshihiro Sakamoto, Masaumi Otake, Seiji Kawaguchi, Kenjirou Fujii, Yuki Kanemori, and S. Sugano, "GPS-based indoor positioning system with multi-channel pseudolite," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 905–910.

[3] K. Özsoy, A. Bozkurt, and I. Tekin, "2D Indoor positioning system using GPS signals," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, Sep. 2010, pp. 1–6.

[4] *ANN-MS Data Sheet*, u-blox AG, Switzerland, Nov. 2017.

[5] *NEO-8Q / NEO-M8 Hardware Integration Manual*, u-blox AG, Switzerland, Nov. 2017.

[6] *NEO-M8 Data Sheet*, u-blox AG, Switzerland, Aug. 2016.

https://doi.org/10.24355/dbbs.084-201907151114-0

https://doi.org/10.24355/dbbs.084-201907151114-0

Hardware Design for an Angle of Arrival Positioning System

Marco Gunia*, Adrian Zinke*, Niko Joram* and Frank Ellinger*
*Chair of Circuit Design and Network Theory (CCN)
Technische Universität Dresden, 01062 Dresden
Email: marco.gunia@tu-dresden.de

***Abstract*—Nowadays, there are a variety of different indoor positioning systems employing diverse techniques. Besides inertial navigation, most systems are based on ranging. For instance, techniques utilizing the Received Signal Strength (RSS), phase-difference, time of flight or time of arrival usually convert this information to distances, where trilateration is used to calculate the position. In contrast, approaches utilizing the Angle of Arrival (AoA) are commonly overlooked in literature. Within this paper, we present the hardware design for building such a system and give an outlook towards the algorithms to determine of the AoA. Preliminary measurements show the operability of the system.**

Keywords—Localization, Positioning, Angle of Arrival

I. Introduction

Modern hardware platforms are typically advertised by means of new localization features. For outdoor applications, customers expect an integration of the Global Positioning System (GPS), which is a low-cost readily available solution. For indoor, there exist a multitude of isolated techniques ranging from cheap and inaccurate RSS-based variants with errors up to 30 m [1] to proprietary approaches like Frequency Modulated Continuous Wave- (FMCW) radar, exhibiting errors well below 1 m. Furthermore, there are already commercial off-the-shelf Integrated Circuit (IC). Offering decimetre accuracy, the *Decawave DW1000* enables the setup of an Ultra-WideBand- (UWB) radar [2]. An alternative with comparable accuracy and costs is phase-based ranging by means of the *Atmel AT86RF233* IC [3]. These indoor approaches have in common that the measured quantity is converted to ranges by means of a channel model. Afterwards, the position is obtained with the help of some form of trilateration, e.g. least squares. In contrast to these ranging-based variants, there are techniques employing the angle of incidence, where the final position is determined by triangulation. In hybrid systems, this provides another quantity which can be used to improve the positioning estimate from classical ranging-based techniques.

In this paper, the hardware design for an AoA positioning system is introduced by employing off-the-shelf components, with the overall goal to reduce time to market. Another requirement on the system is to be prepared for hybrid localization in future releases. The rest of this paper is organized as follows. In section II, related research is shortly introduced. This is followed by the general principle in section III. We present the hardware design in section IV. Preliminary results are illustrated in section V. The last section concludes the paper.

II. Related Work

AoA systems can broadly be classified into beam former and subspace-based techniques. Beam former steer and shape the beam, e.g. by utilizing antenna characteristics or by employing digital orientation within multiple antenna designs. In contrast, subspace-based algorithms subdivide the covariance matrix of the received signal into a signal and noise space by singular value decomposition. An orthogonal categorization is the classification into parametric and spectral algorithms. Parametric approaches directly determine the angle of incidence, e.g. by solving equation systems. In contrast, spectral techniques calculate a spectrum, showing power, amplitude or signal strengths as a function of the angle of incidence, where peaks determine the angle estimate [4]. Below, we introduce one variant from each of these 2×2 categories.

The reference signal method, presented in detail in section III, is a parametric beam former. It determines the angle by measuring the phase of the incident signal on every antenna in relation to a local reference. Beneficial is the low complexity, however the signal shape must be known, only one sender can be identified and the method is prone to time-dependent frequency drifts. Bartlett is a classical spectral beam former, where the main lobe of the antenna is digitally moved back and forth to record a spectrum. It has low complexity and no a-priori knowledge about the signal characteristics is necessary. However, the resolution depends on the antenna configuration [5]. ESPRIT [6] is a parametric subspace-based approach consisting of multiple antennas. A prerequisite is a constant distance between adjacent antenna elements, thus ideally the incident signal exhibits the same timely displacement between adjacent antenna elements. ESPRIT firstly determines the covariance matrix and calculates the signal and noise space in a second step. Afterwards, the distance can be calculated directly. MUSIC, which is a spectral subspace-based approach, is similar to ESPRIT in its first steps, but determines a spectrum in its final phase instead [4], [6]. Beneficial is its high resolution, its independence on the shape of the incident signal and the recognition of multiple senders. However, the amount of senders must be known a-priori, the computing complexity is high and there are strong requirements concerning the equivalence in the multiple receiving paths.

III. Determining the Angle of Arrival

Below, we present the general principle for determining the angle of arrival. Let $\mathtt{A}$ be a sending node, transmitting a

https://doi.org/10.24355/dbbs.084-201907151114-0

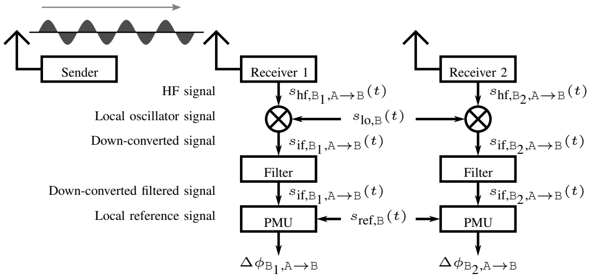

Sender
Receiver 1
Receiver 2
HF signal
Local oscillator signal
Down-converted signal
Down-converted filtered signal
Local reference signal
$s_{\mathrm{hf},\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}(t)$
$s_{\mathrm{hf},\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}(t)$
$s_{\mathrm{lo},\mathtt{B}}(t)$
$s_{\mathrm{if},\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}(t)$
$s_{\mathrm{if},\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}(t)$
Filter
Filter
$s_{\mathrm{if},\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}(t)$
$s_{\mathrm{if},\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}(t)$
PMU
$s_{\mathrm{ref},\mathtt{B}}(t)$
PMU
$\Delta\phi_{\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}$
$\Delta\phi_{\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}$

Fig. 1. Receiving path

sinusoidal signal (index: hf) with frequency $\mathrm{f}_{\mathrm{hf},\mathtt{A}}$

$$s_{\mathrm{hf},\mathtt{A},\mathtt{A}\to\mathtt{B}}(t) = \underbrace{A_{\mathrm{hf},\mathtt{A}}(t)}_{:=1}\ \sin\left(2\pi f_{\mathrm{hf},\mathtt{A}}t + \varphi_{\mathrm{hf},\mathtt{A}}\right) \qquad (1)$$

where without loss of generality it is assumed $A_{\mathrm{hf},\mathtt{A}}(t) = 1$. Let $\mathtt{B}$ be a two-antenna receiving node, where the antennas are separated by a distance a, as shown in Fig. 2. The above signal received at the first antenna (index: $\mathtt{B}_1$) reads

$$s_{\mathrm{hf},\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}(t) = \sin\left(2\pi f_{\mathrm{hf},\mathtt{A}}\left[t - \frac{d_1}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}}\right) \qquad (2)$$

Considering Fig. 2, the signal travels an additional distance Δd to the second antenna

$$\Delta d = d\sin(\alpha) \qquad (3)$$

which is equivalent to an additional propagation time ${}^{\Delta d}/_{c}$. Thus, we have for the received signal at antenna 2 (index: $\mathtt{B}_2$)

$$s_{\mathrm{hf},\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}(t) = \sin\left(2\pi f_{\mathrm{hf},\mathtt{A}}\left[t - \frac{d_1}{c} - \frac{\Delta d}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}}\right) \qquad (4)$$

For simplicity we assume that both signals are down-converted by means of the same local oscillator (index: lo)

$$s_{\mathrm{lo},\mathtt{B}}(t) = \sin\left(2\pi f_{\mathrm{lo},\mathtt{B}}t + \varphi_{\mathrm{lo},\mathtt{B}}\right) \qquad (5)$$

as illustrated in Fig. 1. Analogous to [3], these down-converted signals (index: if) read

$$\begin{aligned} &s_{\mathrm{if},\mathtt{B}_1,\mathtt{A}\to\mathtt{B}}(t)\\ &= \sin\left(2\pi f_{\mathrm{if},\mathtt{B}}t - 2\pi f_{\mathrm{hf},\mathtt{A}}\left[\frac{d_1}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}} - \varphi_{\mathrm{lo},\mathtt{B}} + \frac{\pi}{2}\right)\\ &s_{\mathrm{if},\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}(t)\\ &= \sin\left(2\pi f_{\mathrm{if},\mathtt{B}}t - 2\pi f_{\mathrm{hf},\mathtt{A}}\left[\frac{d_1}{c} + \frac{\Delta d}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}} - \varphi_{\mathrm{lo},\mathtt{B}} + \frac{\pi}{2}\right) \end{aligned} \qquad (6)$$

where $f_{\mathrm{if},\mathtt{B}} := f_{\mathrm{hf},\mathtt{A}} - f_{\mathrm{lo},\mathtt{B}}$. Afterwards, these signals are compared to a local reference $s_{\mathrm{ref},\mathtt{B}}(t)$ with equal frequency

$$s_{\mathrm{ref},\mathtt{B}}(t) = \sin\Big(2\pi f_{\mathrm{ref},\mathtt{B}}t + \varphi_{\mathrm{ref},\mathtt{B}}\Big) \quad ; \quad f_{\mathrm{ref},\mathtt{B}} := f_{\mathrm{if},\mathtt{B}} \qquad (7)$$

to obtain phase differences $\Delta\phi_{\mathtt{B}_\mathtt{i},\mathtt{A}\to\mathtt{B}}$ to antenna i $(i = 1, 2)$ [7]. Doing so, we have

$$\begin{aligned} \Delta\phi_{\mathtt{B}_1,\mathtt{A}\to\mathtt{B}} &= -2\pi f_{\mathrm{hf},\mathtt{A}}\left[\frac{d_1}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}} - \varphi_{\mathrm{lo},\mathtt{B}} + \frac{\pi}{2} - \varphi_{\mathrm{ref},\mathtt{B}}\\ \Delta\phi_{\mathtt{B}_2,\mathtt{A}\to\mathtt{B}} &= -2\pi f_{\mathrm{hf},\mathtt{A}}\left[\frac{d_1}{c} + \frac{\Delta d}{c}\right] + \varphi_{\mathrm{hf},\mathtt{A}} - \varphi_{\mathrm{lo},\mathtt{B}} + \frac{\pi}{2} - \varphi_{\mathrm{ref},\mathtt{B}} \end{aligned} \qquad (8)$$

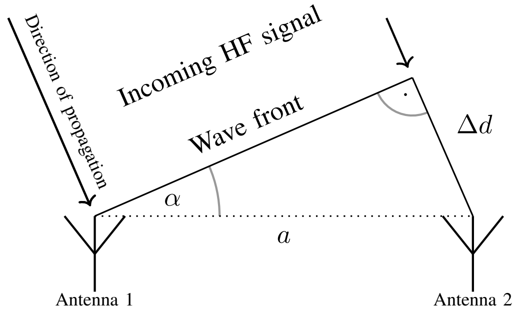


Fig. 2. Wave front of HF signal

Subtracting these phase differences, we obtain a formula for the incident angle α

$$\Delta\phi_{\mathtt{B}_1,\mathtt{A}\to\mathtt{B}} - \Delta\phi_{\mathtt{B}_2,\mathtt{A}\to\mathtt{B}} = 2\pi f_{\mathrm{hf},\mathtt{A}}\frac{\Delta d}{c} = 2\pi f_{\mathrm{hf},\mathtt{A}}\frac{d\sin(\alpha)}{c} \qquad (9)$$

where eq. (3) was used. Reordering, we have

$$\sin(\alpha) = \frac{c\left(\Delta\phi_{\mathtt{B}_1,\mathtt{A}\to\mathtt{B}} - \Delta\phi_{\mathtt{B}_2,\mathtt{A}\to\mathtt{B}}\right)}{2\pi\ f_{\mathrm{hf},\mathtt{A}}\ d} \qquad (10)$$

Above, we assumed that both signals are down-converted by means of the same local oscillator and evaluated with respect to the same reference. It is only mentioned in passing, that both assumptions are not essential and could be discarded in a productive system.

IV. Hardware Design

Below, we present the hardware, where we limit ourselves to the design of the Printed Circuit Board (PCB), which is shown in Fig. 3. On the long back side, there are SMA connectors for four 2.4 GHz antennas, where each is controlled by a single RF transceiver. Here, we employ the *Atmel AT86RF215*. These are placed close to these connectors, where the length of the wires between antenna and transceiver are kept equal for all IC. Moreover, at each transverse side, there is an additional SMA connector for 0.9 GHz communication, respectively. The system is controlled by a *Lattice iCE40HX8K* Field Programmable Gate Array (FPGA), which is responsible for parallel configurating the RF transceiver via the Serial Peripheral Interface (SPI). In turn, the transceiver generate Low Voltage Differential Signaling (LVDS) data at 128 MBit/s, being converted by LVDS converters to single-ended signals. The FPGA either stores the data stream within an 16 MBit SRAM or directly transfers it to a host via Universal Serial Bus (USB) 3.0. The distance between the USB entities and the 2.4 GHz nets are selected as far away as possible to avoid coupling into these high frequency nets. For configuration and debugging, the PCB contains five switches, eight LED and an eight pin General Purpose Input Output (GPIO) header. The FPGA can either be configured via SPI, a separate flash memory or a programming interface. The PCB described so far is a stand-alone solution. However, it can easily be included within our hybrid positioning system [2], [3], by assembling the 100 pin connectors on the front side, which are left empty in Fig. 3. The PCB is supplied by 5 V Direct Current (DC), either from USB or from the 100 pin connectors. This is converted to 3.3 V, 2.5 V, 1.2 V level for the FPGA. These

https://doi.org/10.24355/dbbs.084-201907151114-0

DC-DC converters are also placed far away from the high frequency nets. In the following paragraphs, we present the layout regarding two critical design parameters.

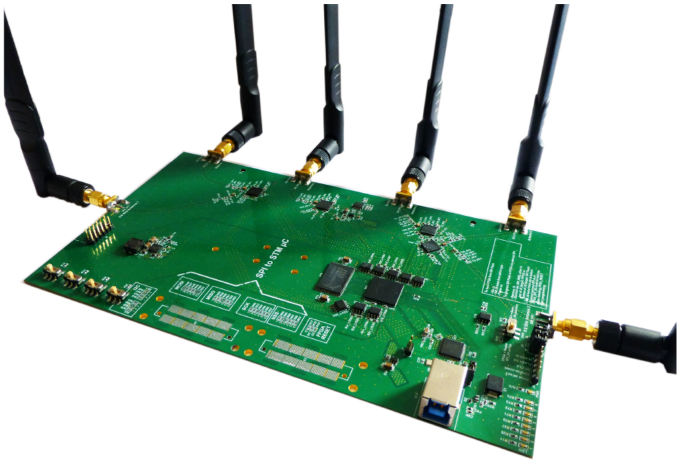


Fig. 3. Printed Circuit Board

A. Antenna distance

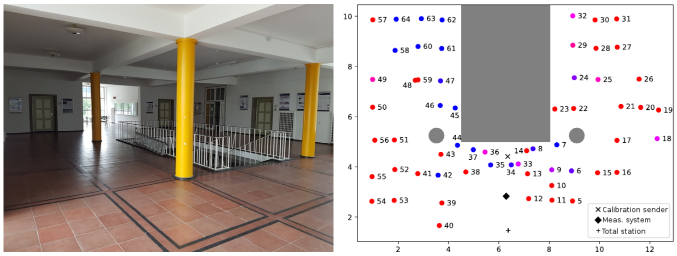
For determining the AoA, we employ the MUSIC algorithm [8]. The influence of antenna spacing a on the accuracy is analysed in [9]. It is shown that the error decreases with increasing a. [10] argues that the optimal distance is close to $0.5\,\lambda$ and that ambiguities arise for $a > 0.6\,\lambda$. Hence, a should fulfil $0.5\,\lambda \leq a \leq 0.6\,\lambda$. In this work, we select $a := 0.5\,\lambda$. [11] recommends dimensioning the antenna distance for the highest possible frequency, which is 2483.5 MHz for the AT86RF215. In doing so, we have $a = 0.5\,\lambda = 0.5\,{}^{c}\!/\!_{f} = 6.04\,\mathrm{cm}$.

B. Clock generation

The transceivers operate with 26 MHz clock. For determining the AoA, the phase relations of these IC are required. To alleviate frequency drifts between the transceiver, these are connected to the same clock. We used a voltage controlled temperature compensated crystal oscillator, which is placed in the centre of the transceiver. We carefully designed the distances between crystal and all transceiver to be almost equally long. In doing so, the overall clock net length must be minimized to reduce capacitive load to obtain a pure clock.

V. Experiments

In Fig. 4b we present the first preliminary results for a $17\,\mathrm{m} \times 12\,\mathrm{m}$ university hall, shown in Fig. 4. The hall contains pillars, stairs and metallic handrails in the line of sight, hence multi-path is supposed to happen. The receiver is held at a constant position, indicated by the diamond. In contrast, the sender is placed at 64 different positions, characterized by the circles. A very coarse metric is used, where the blue colour indicates differences below 10° between the true and measured incident angle. All other measurement points are coloured red. As can be seen, there are many blue points close to the receiver. In contrast, for positions where the signals travel through the metallic handrails, the measurements are worse. Nevertheless, it can be seen that the system is operable.



(a) Outline

(b) Measurements

Fig. 4. Experiment in university hall

VI. Summary and Outlook

This paper presented the hardware design for an AoA-based positioning system. Besides introducing the general principle together with related work, special emphasis is put on the selection and collaboration of the hardware components for the embedded system. The core is a FPGA, on the one hand being responsible for controlling four RF IC. On the other hand, it sends the measurement data via an USB 3.0 to a host PC, where the angle is calculated by means of a Python script. Finally, the first measurements are presented showing good results in sophisticated multi-path environments.

Acknowledgment

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-FP7-611526 (MAGELLAN).

References

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007, ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905750.

[2] M. Gunia, F. Protze, N. Joram, and F. Ellinger, "Setting up an ultra-wideband positioning system using off-the-shelf components," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, Oct. 2016, pp. 1–6. DOI: 10.1109/WPNC.2016.7822860.

[3] M. Gunia, A. Zinke, N. Joram, and F. Ellinger, "Setting up a phase-based positioning system using off-the-shelf components," in *Proc. 14th Work. Positioning, Navig. Commun. (WPNC'17)*, Bremen, Germany: IEEE, 2017, pp. 1–6. DOI: 10.1109/WPNC.2017.8250065.

[4] S. Ejaz and A. Shafiq, "Comparison of spectral and subspace algorithms for fm source estimation," *Progress in Electromagnetics Research C*, vol. 14, pp. 11–21, Jan. 2010. DOI: 10.2528/PIERC10042705.

[5] C. M. Reustle, "Konzeptionierung, erforschung und erprobung eines 24ghz sekundärradar tachymeters zur präzisen messung der ausladung eines mobilkranes," pp. 1–193, 2018. DOI: urn:nbn:de:bvb:29-opus4-102718.

[6] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, Jul. 1989, ISSN: 0096-3518. DOI: 10.1109/29.32276.

[7] M. Pelka, C. Bollmeyer, and H. Hellbrück, "Accurate radio distance estimation by phase measurements with multiple frequencies," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2014, pp. 142–151. DOI: 10.1109/IPIN.2014.7275478.

[8] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986, ISSN: 0018-926X. DOI: 10.1109/TAP.1986.1143830.

[9] L. Huang, H. Chen, Y. Chen, and H. Xin, "Research of doa estimation based on music algorithm," Jan. 2016. DOI: 10.2991/mmebc-16.2016.432.

[10] M. Mohanna, M. L. Rabeh, E. M. Zieur, and S. Hekala, "Optimization of music algorithm for angle of arrival estimation in wireless communications," *NRIAG Journal of Astronomy and Geophysics*, vol. 2, no. 1, pp. 116–124, 2013. DOI: 10.1016/j.nrjag.2013.06.014.

[11] H. Tang, "Doa estimation based on music algorithm," *Linné-Universität*, 2019.

https://doi.org/10.24355/dbbs.084-201907151114-0

https://doi.org/10.24355/dbbs.084-201907151114-0

Navigation in a library using smartphones

Janek Stoeck, Harald Sternberg
Dept. Geodesy and Geoinformatics
HafenCity University Hamburg
Hamburg, Germany
firstname.name@hcu-hamburg.de

***Abstract*—In these times, when everyone wants to do as much work as possible in the least amount of time, it is important to make the workflow as efficient as possible. Spending an unnecessary amount of time looking for things is therefore contra productive. Using the library HafenCity University Hamburg (HCU) as an example, this paper shows how the time required to find a book can be reduced by introducing an indoor navigation system for that purpose.**

Acceleration sensors are used to realize a step counter. In combination with gyroscope data a dead reckoning can be carried out. The height information is derived of barometric measurements. For additional support, routing graphs are used to constrain the walking direction. With this setup and the additional topological we achieve deviations between last estimated to wanted position less than two meters.

***Index Terms*—indoor navigation, indoor localization, dead reckoning, routing, topological support**

I. Introduction

There are many technologies to track people in indoor environments. Infrastructure-based technologies, such as fingerprinting with WLAN APs or Bluetooth beacons, but also those that do not require additional external hardware, such as inertial measurement technology.

Infrastructure-based technologies have the disadvantage that, on the one hand, costs are incurred for hardware and and on the other hand an own development must be adapted in such a way that the normal user can handle the product untrained. The advantage here is that the achievable accuracies remain the same under optimal conditions.

The advantage of inertial measurement technology is that it can be used out-of-the-box. However, due to the quality of the sensors used and the possible low robustness of the algorithms, time-dependent errors can occur. The biggest advantage for the broad use of such a system is that there is a multi-sensor system which almost every user already knows: the smartphone.

It combines measurement technology for tracking the position as well as all other, even if possibly limited, advantages of modern computers in a portable, handy device. This feature makes it possible to realize even complex applications that combine different disciplines.

Such a complex application is also represented by the presented project. In the library of the HCU an indoor navigation system is to be made available, which is to lead the user to the desired book. This is necessary because the library's stock have been brought together from three different old stocks, thus creating an inhomogeneous signature pattern. In addition, the locations of the shelves of the books are only indicated for each floor, so that on the one hand the entire floor often has to be searched for the correct shelf and on the other hand the indication of the signature area on the shelf makes recognition even more difficult.

In the following Section II related works are shown. Afterwards in Section III the application with regard to localization is described. In Section IV first results are shown, whereupon in Section V a summary and an outlook into the following work is given.

II. Related Work

There are different approaches to enable an indoor navigation system via smartphone. Without further support by other sensors dead reckoning (DR) is usually used by means of step counters from accelerations and integrated rotation rate as in [1] and [2] and partly in [3].

To minimize the inaccuracies of the sensor drift, [1] and [2] use particle filters. In [1] topological information, like the distance of a particle to a routing graph, is used to weight the particles. The particles in [2] are scattered over the whole area and disappear after passing a wall.

The accuracy of particle filters strongly depends on the number of particles used. Due to the restricted performance of smartphones, however, this is limited, so that a sufficiently accurate position estimate can no longer be achieved at runtime.

Another possibility is, as in [3], to integrate acceleration twice to distances. Here the DR by smartphone is supported by using a Smartwatch attached to the ankle of a foot. It must be taken into account that the drift caused by the sensor noise increases significantly with time advances due to integration. This is particularly the case with double integration. Here, corrections are made using zero velocity update (ZUPT) when the foot is at a standstill.

An additional support by further devices, which have to be located at other places of the body, is not reasonable, because this would have to be provided by the library and furthermore this does not allow an intuitive use of the application. Accordingly, no ZUPTs can be applied, because in the use-case the user won't hold the smartphone that still that they could be calculated, and double integration would lead to significant errors after a very short time.

The position can also be supported by external sensors. This

https://doi.org/10.24355/dbbs.084-201907151114-0

happens in [2] where magnetic fields are used to further weight the particles of the filter based on the received magnetic field strength. This greatly shortens the time it takes to estimate the position.
Since no new sensors are to be installed in the library, this is out of the question.

III. System setup

For a navigation system used to find rooms, an accuracy of 5 m is sufficient. In this case, this is not accurate enough, as the individual shelves are approximately 0.5 m apart. The goal is to stand with an accuracy of less than 2 m in front of the shelf. Since the shelves themselves have printed information on the signature area, this is sufficient. Any remaining errors can be detected by the user himself. Reaching the correct side of the shelf is an extended goal. Here the user should be guided into the corridors and stand in front of the correct side of the shelf (see Fig. 1).
The inertial navigation using smartphones is realized by the inertial measuring unit (IMU) (accelerometer, gyroscope) and a barometer. As already mentioned in Section II, a DR with a step counter and integration of the rotation angle is usually realized. This is also the case in this project. The step counter is an implementation according to [1].
In order to obtain a correct trajectory, the translation must be determined. On the one hand, a fixed value could be assumed (mean step length) on the other hand a step length estimation could be performed. For this purpose the step length estimation according to [4] by Weinberg with the formula (1) can be used. K is a factor that remains constant and results in a different step length based on the variation of the accelerations in the maximum and minimum amplitude (A_{max} and A_{min}) during the step. K results from the ratio between the real distance and the estimated distance of the step length estimator.

$$l_i = K * \sqrt[4]{A_{max} - A_{min}} \tag{1}$$

A variation of this is given in [5] by Ho, Truong and Jeong, where the same formula is used, but the K factor for each step is estimated. They estimate the K factor by applying a polynomial function seen in equation (2) with the average velocity v_{step} during a step. However, they use a smoothed signal from the accelerometer so that the constant of 0.68 they found for the polynomial does not apply to unsmoothed signals.

$$K_i = 0.68 - 0.37 * v_{step} + 0.15 * v_{step}^2 \tag{2}$$

Another possibility is shown in [6] by Kim et al.. In formula (3) the absolute sum of the accelerations ΣA in respect to total samples N during a step is used instead of the minima and maxima.

$$l_i = K * \sqrt[3]{\frac{\Sigma A}{N}} \tag{3}$$

The height information for the assignment of the correct floor is obtained using the general barometric height formula (4) by [7], where $p(h)$ is the air pressure measured by the barometer, so that a 2D+1D trajectory can now be formed.

$$h_b = \frac{288.15K - 288.15K * \sqrt[5.255]{\frac{p(h)}{1013.25hPa}}}{0.0065\frac{K}{m}} \tag{4}$$

The orientation as well as the positions are supported with the help of a routing graph. The algorithm in 1 changes the current orientation of the smartphone Ψ to the orientation of the graph $\sphericalangle_P$ if the distance between the routing graph is less than 0.3 m and the absolute difference between both orientations is less than 7.5°. In addition, the position is set to the base point on the graph. Thus the user moves on the graph. If the difference between $\sphericalangle_P$ and Ψ is greater than 7.5° the graph is left.

Another topological support is done if the position of the smartphone is within a predefined polygon. This marks the area of a staircase and the step length is set to the step width of a staircase step. This way one can maintain an approximated correct position, since the step length algorithms do not calculate this correctly for stairs. If no barometer is present in a smartphone, this information can also be used to artificially change the height by one stair step height.

Algorithm 1 Snap Position to Graph

```
if $\overline{P(x,y), BP(x,y)} < 0.3m$ then
    if $|\Psi - \sphericalangle_{graph}| < 7.5°$ then
        $\Psi \leftarrow \sphericalangle_{graph}$
        $P \leftarrow BP$
    end if
end if
```

IV. Experimental Results

In order to determine the achievable results of the step length estimator presented in the previous Section, a test route was carried out in the HCU building. The distance covered was about 49 m. The tests was done with six different persons, but with the same smartphone. First of all the K factor had to be determined. This was done by comparing the estimated distance of each algorithm to the ground truth. Afterwards an average value for the factor was calculated and included to the algorithms, except for the mean value (0.75 m) and the estimator of Ho, Truang and Jeong. As the formula (2) is used for smoothed accelerations, we decreased the constants iteratively to get the best for unsmoothed signals. Based on this we decided to decrease it to 0.61. After recalculating the different distances an average of the difference between ground truth and the different distances of the persons could be made. The results can be seen in TABLE I.

There is basically no big difference between the algorithm by Weinberg and the one by Kim et al. As the algorithm of Kim et al. has a greater maximum deviation, we chose the Weinberg method. The possibility to estimate the K factor with (2) seems to be to inaccurate, as its deviation to the truth is more than 20 % but as mentioned in Section III this estimator expects a filtered signal according to [5], which

TABLE I: Results of the tests for the different step length estimator with six different Persons (P1-P6). Values are the absolute difference to the ground truth in percent

P	Weinberg [%]	Kim [%]	Ho [%]	mean [%]
P1	7.0	5.7	12.9	11.7
P2	5.8	2.7	28.0	13.1
P3	8.0	5.5	32.9	17.9
P4	5.8	8.6	15.7	0.5
P5	0.3	8.1	26.5	20.9
P6	2.7	3.5	21.1	1.5
Mean	4.9	5.7	22.8	10.9

wasn't done in this case. The method by applying a mean step length turns out to be better than that, but compared to the estimators of Weinberg and Kim et al. it performs worse than them.

Eventually the K factor (for the Weinberg method) was taken into the account for the following tests. The whole system was tested in the library of the HCU. The starting point of the test was on the ground floor of the library in front of the stairs and the destination was a shelf on the 1st floor. This route was run 13 times and the difference between target and actual position was calculated. The results are shown in Fig. 1. There the blue crosses represent the last estimated position of the tests, the red cross the end node to be reached, the red rectangle the corresponding shelf and the green line the approximate course of the different tests. In the Fig. 1 it can be seen that the results are all scattered in the range of maximum 2 m around the true position and are also rather southwest. This indicates that the step counter counts too few steps. The mean deviation is 1.35 ± 0.5 m. This is a accuracy of 4 % of traveled distance.

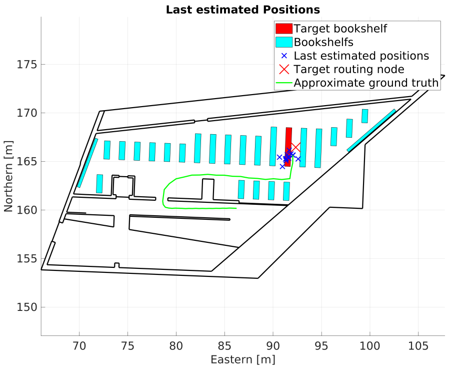


Fig. 1: Results of the position estimated in the library

V. Conclusion and Outlook

Since the stock of the library was merged of three different old ones, the book signatures are inhomogeneous. This and the inaccurate description of the location (only to the floor), lead to a unnecessary big time to find a book searched. Because of that this paper presents an application and the algorithms used to navigate trough the library in order to lead the way to the book looked for.

In general other works show, that results with an accuracy of 2-5 m is achievable, but in this particular case a higher accuracy was required. In order to achieve this, we added topological support by taking the routing graph into account. To overcome the problem if there is no barometer present in a smartphone, a topological support can be used to define areas where height changes are possible like on stairs. However, this hasn't been compared to the barometric height estimation, which is planned for the next investigations. Further we investigated different step estimators to improve the accuracy of the translation between two positions. In the past we used a mean step length, which worked fine, if one user would use the application, but since the application is meant to be usable to everyone this would turn out to be a big error source. Because of the results in Section IV we decided to use the the method of Weinberg, but we will do further investigate the algorithm of Ho, Truong and Jeong, because according to their work, the results should perform better.

The requirement of less than two meters is satisfied as the experimental results are within this range. Nevertheless the last estimated position don't match the requirement to stand before the correct side of the shelf. This seems to be due to to less recognized steps, so the step counters parameters have to be tweaked.

Another possibility to overcome these last deviation is to implement more sensors. For example, the camera could be used to identify the shelf in front of which the users are located, and and give feedback if the user has to go further. Also it may be worth to implement the camera as a source of position estimation doing structure from motion.

References

[1] T. Willemsen, "Fusionsalgorithmus zur autonomen Positionsschätzung im Gebäude, basierend auf MEMS-Inertialsensoren im Smartphone," 2016.

[2] C. Real Ehrlich, J. Blankenbach, and A. Sieprath, "Towards a robust smartphone-based 2, 5d pedestrian localization," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.

[3] M. Gunia, Y. Wu, N. Joram, and F. Ellinger, "Building up an inertial navigation system using standard mobile devices," *Journal of Electrical Engineering*, vol. 5, pp. 299–320, 2017.

[4] H. Weinberg, "Using the adxl202 in pedometer and personal navigation applications," *Analog Devices AN-602 application note*, vol. 2, no. 2, pp. 1–6, 2002.

[5] N.-H. Ho, P. Truong, and G.-M. Jeong, "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone," *Sensors*, vol. 16, no. 9, p. 1423, 2016.

[6] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Positioning*, vol. 1, no. 08, p. 0, 2004.

[7] H. Kahmen, "Angewandte geodäsie: Vermessungskunde [applied geodesy: Measurement science]," 2006.

https://doi.org/10.24355/dbbs.084-201907151114-0

Understanding and Prediction of Ultra-Wide Band Channel Impulse Response Measurements

Benjamin Matthews*, Sven Ole Schmidt† and Horst Hellbrück†
Lübeck University of Applied Sciences, Germany
Department of Electrical Engineering and Computer Science
* Email: benjamin.matthews@stud.th-luebeck.de
† Email: sven.ole.schmidt, horst.hellbrueck@th-luebeck.de

***Abstract*—Recently, ultra-wide band transceiver systems have provided data transfer, timestamps and channel impulse response measurements to the user. The interpretation of the timestamps and the channel impulse response, however, is difficult and not intuitive. In simple scenarios, line of sight and non-line of sight pulses can be distinguished easily, which simplifies the reconstruction. For more complex scenarios, the interpretation remains difficult and is still an unsolved problem. In this paper, we investigate the channel impulse response measurements of the DecaWave DW1000 ultra-wide band transceiver and model the expected results for simple scenarios based on information provided from the transceiver data sheet. We will show that we are able to predict the measurement results of the transceiver with acceptable accuracy by applying the model above in experiments.**

***Index Terms*—channel impulse response, ultra-wide band, channel estimation, internet of things**

I. Introduction and Related Work

For the Internet of Things (IoT), ultra-wide band (UWB) communication is a promising alternative compared to existing narrow-band or spread spectrum solutions. In contrast to other IEEE 802.15.x solutions, UWB is very resilient against multipath propagation [1]. The reason is that due to large bandwidth, symbols are much shorter compared to a smaller bandwidth. Consequently, individual path components due to reflections do not overlap and this reduces the effects of inter symbol interference. Combining the short pulses with high speed clocks and time measurement units, the transceiver provides precise time-of-flight measurements, which are applied in various localization systems, like [2] and [3].

For spread spectrum technologies, pilot symbols are sent to estimate the current multipath channel in an OFDM System in [4], which is also an option for UWB systems. In [5], Zhou et. al. apply a stochastic approach to predict the current channel impulse responses (CIR) in a high-speed railway. In [6] and [7], the authors show how to generate keys for security and private transmissions from CIR measurements. Additionally, the channel information is an important optimization criterion during the installation of wireless communication systems [8]. In summary, the CIR serves various purposes in different application fields.

Consequently, we investigate CIRs as a feature of an UWB transceiver which is called *CIR data*. The CIR data from the transceiver is a result of the transmitted signal, which is affected by the wireless channel and its corresponding multipath spreading, as well as some signal processing performed by the hardware.

The contributions of the paper are:

- We develop a generalized model for wireless UWB propagation.
- We propose a new estimation of the CIR data called the *forward method*.
- We evaluate the approach with measurements.

The rest of the paper is organized as follows. Section II introduces our generalized model for the wireless UWB propagation. Our main contribution, the estimation of the CIR data, called the *forward method*, is proposed in Section III. Section IV provides a comparison of the estimated received signal with measurements. Section V concludes the paper and gives an outlook on future work.

II. Model for UWB signal propagation

This section introduces the transmission characteristics of wireless UWB transceivers and shows how the received signal is affected by multipath propagation.

Assume the signal $x(t)$ is transmitted wirelessly by a tag in a given room geometry. The signal $y(t)$, which is received by the anchor, is a superposition of multiple noisy and power-distinct copies of the transmitted signal, depending on the corresponding multipath propagation of the transmission channel. If the transmission channel is linear and time-invariant (LTI) and includes noise, the received signal $y(t)$ can be described by:

$$y(t) = x(t) * h(t) + w(t), \tag{1}$$

where $w(t) \sim \mathcal{N}(0, \sigma_n^2)$ is additive white gaussian noise.

The CIR $h(t)$ is assumed to be a superposition of N time-shifted dirac-pulses δ with individual receive power values P_{Rx}. Each of the N pulses is a single path of the multipath propagation:

$$h(t) = \sum_{i=0}^{N-1} P_{Rx,i} \cdot \delta(t - \tau_i), \tag{2}$$

where τ_i is the time shift of the signal of the i-th path.

In our investigation, the transmitted signal $x(t)$ is from an IEEE 802.15.4 compliant DecaWave DW1000 coherent UWB transceiver [9]. The CIR measurements of the transceiver are

https://doi.org/10.24355/dbbs.084-201907151114-0

presented in Section IV. Figure 1 shows the amplitude spectrum $X(f)$ of the transceiver from the datasheet, as well as the reconstructed corresponding transmitted signal $x(t)$ in the time domain.

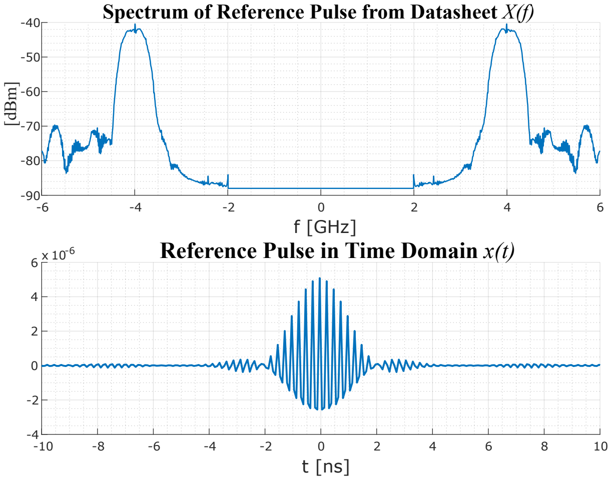


Fig. 1: Transmitted signal in frequency (top) and time domain (bottom)

The signal $x(t)$ is similar to the expected root-raised cosine reference pulse specified in the IEEE standard. Therefore, we model the transmitted signal $x(t)$ from the DecaWave transceiver as a pulse shape given in Figure 1.

The received signal $y(t)$ is processed by the hardware; mainly down-mixing and subsequently filtering. This results in a baseband signal $y_{CIR}(t)$, which is the signal provided by the transceiver hardware. Since this is the signal used to evaluate our predicted CIR, we will call it the CIR data.

III. Forward Method for Estimation of the CIR Data

After introducing the shape of the transmitted signal $x(t)$ and the incurrence of the corresponding CIR data $y_{CIR}(t)$ in the last section, this section will show the forward method to predict this signal. Figure 2 shows a block diagram outlining the signal processing.

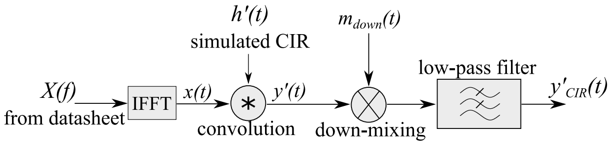


Fig. 2: Block diagram of the forward method

As shown in (2), the CIR $h(t)$ is modeled as a series of dirac pulses of varying magnitude. These pulses may be either positive or negative, depending on the individual paths of the signals. The phase of the complex transmission waves, and therefore the sign of the pulse, results from three factors: the phase offset at transmitter side, the phase based on the path length and a phase shift by π for each reflection at an obstacle.

An estimation of a CIR $h'(t)$ must be created. Ideally, this could be derived from the geometry of the room. In this case, $h'(t)$ was created with knowledge of the optimal result. The estimated received signal $y'(t)$ is calculated by convolving the reference pulse $x(t)$ with $h'(t)$. In this case, we assume a noise-free transmission with $w(t) = 0$. Figure 3 shows an exemplary CIR $h'(t)$, as well as its corresponding received signal $y'(t)$ in the time domain.

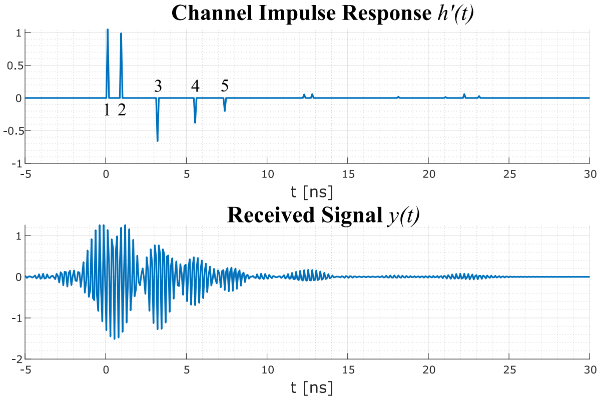


Fig. 3: Convolution of an exemplary CIR $h'(t)$ with reference pulse $x(t)$ to estimate $y'(t)$

To calculate the estimation of the CIR data, $y'(t)$ is down-converted to the baseband. In the next step, a low-pass filter eliminates irrelevant signal components from the baseband signal. This results in an estimation for the CIR data $y'_{CIR}(t)$, which is compared to $y_{CIR}(t)$ measured by the transceiver hardware in the next section.

IV. Comparison between CIR Data and its Prediction

The last sections described the CIR data $y_{CIR}(t)$ and its estimation $y'_{CIR}(t)$. After introducing the measurement setup in this section, we will compare these two signals.

We performed measurements in an obstacle-free space outside of a building to minimize the number of signal paths. The DecaWave transceivers were at a height of 1.35 m each and 3 m apart. When expecting a line-of sight (LOS) peak at 0 ns, the ground reflection is predicted to arrive approx. 3 ns later because the path is approx. 1 m longer. The modeled CIR is the one illustrated in Figure 3 with Peak 1 being the LOS and Peak 3 the expected ground reflection. Figure 3 shows additional peaks that we needed to add to model the measured signal correctly. Peak 2 is a path component that arrives 1 ns after the LOS which we always measure and do not have an explanation for. The path length is just 30 cm longer than the LOS path. Peaks 4 and 5 might arise from additional paths created by the measurement equipment, like the screen of the laptop, and need further investigations.

The transmission is configured to a center frequency of $f_c = 3993.6$ MHz and a corresponding bandwidth of 499.2 MHz and the measurement was captured 176 times.

https://doi.org/10.24355/dbbs.084-201907151114-0

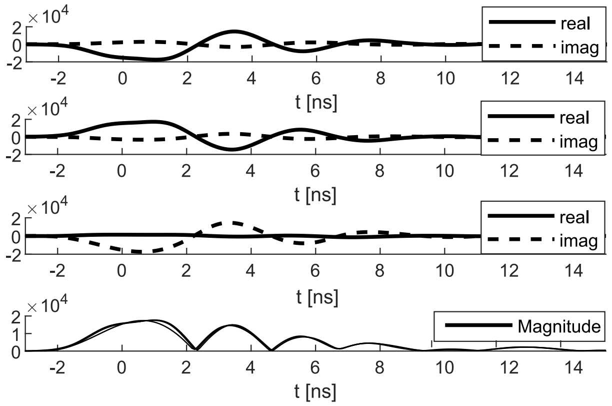


Fig. 4: Three measurements with complex components and magnitudes

Before examining the result of the convolution, we investigate the CIR data $y_{CIR}(t)$. The sampling rate of the signal is 998 MHz. Figure 4 shows a selection of three measurements of the same set up, each shifted to align at $t = 0$ ns. Since $y_{CIR}(t)$ includes phase and magnitude, each measurement provides real and imaginary components. To smooth the signal, a spline fit has been applied to each component. The plot at the bottom of the figure shows the absolute value of all three measurements.

Although the real and imaginary components of the signal change significantly from measurement to measurement, the magnitudes are nearly indistinguishable. The changes for the real and imaginary components result from varying phase offsets between the transmitter oscillator that creates $x(t)$ and the receiver oscillator. Since the offsets are unknown, the magnitude of the measurements is the simplest way to compare the measured signal to the estimated signal.

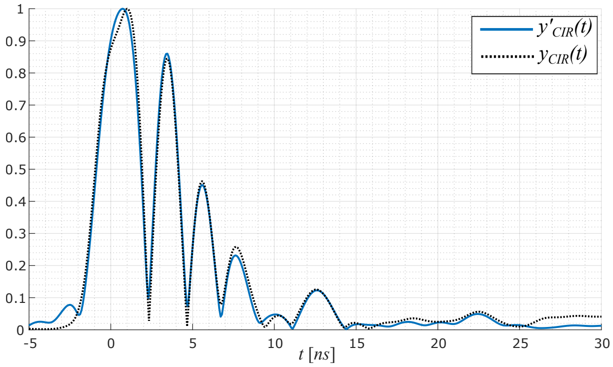


Fig. 5: Comparison of the CIR data $y_{CIR}(t)$ with its estimation $y'_{CIR}(t)$

Figure 5 shows a comparison between a single measurement of the CIR data $y_{CIR}(t)$ and the corresponding estimation $y'_{CIR}(t)$ resulting from the estimated CIR $h'(t)$.

The measurement data and the result of our estimation align very well. The shape and values of the measurements and estimation are very close as shown in Figure 5. The first peak is the line of sight peak. The second peak shows the first reflection. It supports the accuracy of this prediction.

V. Conclusion and Future Work

In this paper, we have proposed a forward method for an estimation of the CIR data $y_{CIR}(t)$. For this, we modeled the wireless UWB transmission and applied it to a baseband signal derived from the datasheet of a transceiver, based on a simulated CIR. Comparing the superimposed measurement data with our estimation depicts the accuracy of this prediction. The shape and values of the measurements and estimation are very close.

For the future, we will investigate the phase – real and imaginary part of the measurements – in more detail. We expect to retrieve additional information which means more details from the path components by these complex measurements compared to the analysis of the magnitudes in the investigation in this paper. Also, the estimation of the CIR $h'(t)$ itself is important. An improved model for the multipath propagation results in a more realistic behavior, which supports the prediction. As an implementation, we will apply the forward model and the measured CIR data in a single anchor localization system.

Acknowledgments

This publication is a result of the research of the Center of Excellence CoSA and funded by the Federal Ministry of Education & Research of the Federal Republic of Germany (Id FKZ ZF4186108BZ8, MOIN). Horst Hellbrück is adjunct professor at the Institute of Telematics of University of Lübeck.

References

[1] M. Z. Win and R. A. Scholtz, "On the robustness of ultra-wide bandwidth signals in dense multipath environments," *IEEE Communications Letters*, vol. 2, no. 2, pp. 51–53, Feb 1998.

[2] S. Krishnan, P. Sharma, Z. Guoping, and O. H. Woon, "A uwb based localization system for indoor robot navigation," in *2007 IEEE International Conference on Ultra-Wideband*, Sep. 2007, pp. 77–82.

[3] B. Grosswindhager, M. Rath, J. Kulmer, M. S. Bakr, C. A. Boano, K. Witrisal, and K. Römer, "Salma: Uwb-based single-anchor localization system using multipath assistance," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '18, 2018, pp. 132–144. [Online]. Available: http://doi.acm.org/10.1145/3274783.3274844

[4] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for ofdm systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3065–3073, Dec 2001.

[5] L. Zhou, F. Luan, S. Zhou, A. F. Molisch, and F. Tufvesson, "Geometry-based stochastic channel model for high-speed railway communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4353–4366, May 2019.

[6] R. Wilson, D. Tse, and R. A. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 364–375, Sep. 2007.

[7] P. Walther, C. Janda, E. Franz, M. Pelka, H. Hellbrück, T. Strufe, and E. Jorswieck, "Improving quantization for channel reciprocity based key generation," 10 2018, pp. 545–552.

[8] T. S. Rappaport, S. Y. Seidel, and K. Takamizawa, "Statistical channel impulse response models for factory and open plan building radio communicate system design," *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 794–807, May 1991.

[9] *DW1000 IEEE 802.15.4-2011 UWB Transceiver Data Sheet*, decaWave, 2015, rev. 2.09.

https://doi.org/10.24355/dbbs.084-201907151114-0