



Data preprocessing for a vehicle-based localization system used in road traffic applications

Timo Patelczyk, Andreas Löffler, and Erwin Biebl

Fachgebiet Höchstfrequenztechnik, Technische Universität München, Arcisstraße 21, 80333 München, Germany

Correspondence to: Timo Patelczyk (timo.patelczyk@tum.de)

Received: 31 December 2015 – Revised: 8 April 2016 – Accepted: 20 July 2016 – Published: 28 September 2016

Abstract. This paper presents a fixed-point implementation of the preprocessing using a field programmable gate array (FPGA), which is required for a multipath joint angle and delay estimation (JADE) used in road traffic applications. This paper lays the foundation for many model-based parameter estimation methods. Here, a simulation of a vehicle-based localization system application for protecting vulnerable road users, which were equipped with appropriate transponders, is considered. For such safety critical applications, the robustness and real-time capability of the localization is particularly important. Additionally, a motivation to use a fixed-point implementation for the data preprocessing is a limited computing power of the head unit of a vehicle. This study aims to process the raw data provided by the localization system used in this paper.

The data preprocessing applied includes a wideband calibration of the physical localization system, separation of relevant information from the received sampled signal, and preparation of the incoming data via further processing. Further, a channel matrix estimation was implemented to complete the data preprocessing, which contains information on channel parameters, e.g., the positions of the objects to be located.

In the presented case of a vehicle-based localization system application we assume an urban environment, in which multipath propagation occurs. Since most methods for localization are based on uncorrelated signals, this fact must be addressed. Hence, a decorrelation of incoming data stream in terms of a further localization is required. This decorrelation was accomplished by considering several snapshots in different time slots. As a final aspect of the use of fixed-point arithmetic, quantization errors are considered. In addition, the resources and runtime of the presented implementation

are discussed; these factors are strongly linked to a practical implementation.

1 Introduction

The objective of a localization system is to determine the relative position of an object to be located. Different applications have brought forth several localization methods (van der Veen et al., 1997), most of which are theoretical. These studies focus on evaluating the performance of these localization methods; however, they neglect the implementation aspect and the impact of hardware components within the localization system. These aspects are strongly linked to a practical system design, which is motivated by the application of a vehicle-based localization system for protecting vulnerable road users. These vulnerable road users are equipped with appropriate transponders. The detection of the vehicle environment is an important step toward autonomous driving. For such safety critical applications, the robustness and real-time capability of the localization is the focus. In addition, the computing power of the head unit of a vehicle is limited. In a real system, such as that introduced above, some non-idealities exist. Here, the focus is on the localization unit. Each localization system must be calibrated, and this calibration is one of the steps used to compensate the non-idealities of the hardware components that compose the system. The location system can be considered as a single-input multiple-output system (cf. Sect. 2). In practice, individual channels of a multi-channel receiver within the localization unit are different in their electrical length and attenuation. This deviation from the model affects the subsequent localization. In addition, in the case of wideband localization, these impacts depend on frequency. Moreover, mutual coupling of the an-

tenna elements have a negative impact on the angle estimation within the joint angle and delay estimation (JADE); thus, it is advisable to consider coupling in the model (Friedlander and Weiss, 1991). In addition to the aforementioned effects, other challenges, including detection of the desired signal from the continuously sampled received data, must also be addressed efficiently. Furthermore, it is necessary to estimate the channel from the incoming data; numerous approaches have been proposed for this purpose. In Cozzo and Hughes (2003), expectation maximization-based channel estimation is considered; other approaches for channel estimation can be found in Pal (1992), Yang et al. (2001), Van der Veen et al. (1995) and Biguesh and Gershman (2006). The channel information lays the foundation for JADE, and thus, for the localization of an object such as a vulnerable road user.

The aforementioned challenges must be effectively implemented. In contrast to previous studies, this study focuses on a practical implementation of data preprocessing required for JADE, focusing on a safety critical vehicle-based localization system used in road traffic applications.

The paper is organized as follows: first, a transponder-based localization system for protecting vulnerable road users is presented. Here, the boundary conditions are defined, such as the required throughput of the localization defined by the application. Next, we study the localization unit. System errors, which occur in a practical application in contrast to theoretical considerations, are identified. These errors can be taken into account within the signal processing. For this purpose, a preprocessing step will be introduced to account for the chosen identified non-idealities. This preprocessing step is the focus of this paper. Subsequent localization methods can build on this step. A hardware architecture for compensating these non-idealities is presented. A field programmable gate array (FPGA) is used as the hardware platform. We discuss the key components as well as resource requirement and run time of the architecture. The architecture are assessed in terms of the safety critical application. Finally, a localization method is applied which is based on the presented fixed-point preprocessing. Thus it is possible to make a statement on the impact of fixed-point implementation on the subsequent localization result.

2 Problem formulation

The urban environment is described by a multipath propagation channel, as shown in Fig. 1. The channel model is derived from van der Veen et al. (1997) and Jaafar et al. (2005), and is represented by a complex channel matrix \mathbf{H} , which contains informations about the position of objects. For convenience, we consider a single-user mode here; however, this case is without loss of generality because each user is placed in its own time slot. The access time is denoted by t_a . One user is equipped with a transponder, which responds to a localization request via the localization unit. The transponder

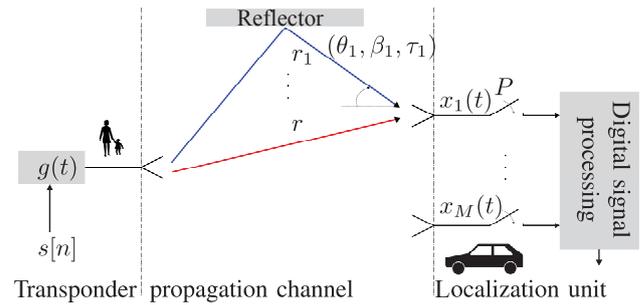


Figure 1. Multipath propagation channel for location applications in the urban environment.

unit transmits a digital modulated sequence $s[n]$ of n symbols over the wireless channel. The received baseband signal $x_m(t)$ at an m th element antenna array at time t is given by the convolution of the transmitted digital sequence $s[n]$ with the channel impulse response $\mathbf{h}(t)$. The channel impulse response includes r discrete paths. Each path is parameterized by a path attenuation β , a path delay τ and a direction of arrival Θ .

The sampled received signal is represented in a compact form by

$$\mathbf{X} = \mathbf{H} \cdot \mathbf{S} + \mathbf{N}, \quad (1)$$

where $\mathbf{X} \in \mathbb{C}^{M \cdot P \times N}$, $\mathbf{H} \in \mathbb{C}^{M \cdot P \times L}$, and $\mathbf{S} \in \mathbb{R}^{L \times N}$ contain the transmitted sequence in the form of a Toeplitz matrix. P denotes an oversampling factor and L denotes the physical channel length. Additive white Gaussian noise (AWGN) is modeled by \mathbf{N} . A direct estimate of the channel matrix is given by

$$\hat{\mathbf{H}} = \mathbf{X} \cdot \mathbf{S}^\dagger, \quad (2)$$

where $(\cdot)^\dagger$ denotes the pseudo-inverse of the signal matrix. This approach ignores noise, but is still chosen because it can be implemented easily in hardware. Furthermore the effect of the noise terms is reduced by averaging several snapshots. The pseudo-inverse of the signal matrix is required because, in general, $L \neq N$. Equation (2) lays the foundation for further considerations.

Finally, the system requirements are considered. The system should be designed for at least 20 users, as such a number reflects the urban environment. In addition, an update of 50 Hz is assumed, which results in an access time of $t_a = 1$ ms. The data preprocessing must be performed at least at this time slot. The subsequent localization can be performed in the background while other users are queried.

3 System design

This section presents the system design of a cooperative sensor system, focusing on an application out of the localization field. First, a block diagram of the system is shown.

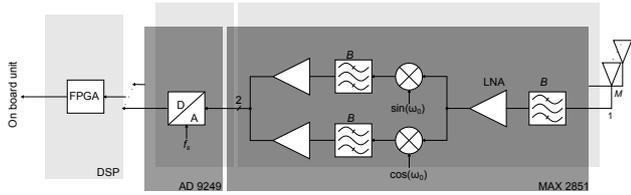


Figure 2. Block diagram of a localization unit as a part of a cooperative sensor system for protection of vulnerable road users.

The function of each component is summarized, and differences with respect to a theoretical consideration are identified. Finally, an efficient preprocessing implementation is presented. Figure 2 shows a block diagram of the localization unit of the localization system, operating at a center frequency of $f_0 = 5.8$ GHz. The application follows the *IEEE 802.11p* standard, which is intended for traffic applications.

On a localization request, the transponder unit responds with a digital modulated transmission sequence $s[k]$. This is received by an uniform linear antenna array. The elements are arranged at a distance $\frac{\lambda_0}{2}$. The radio frequency signal is converted to the baseband. In this case, a complex demodulation approach is used. The sampling frequency f_s is reduced in contrast to the sample rate, which is required at an intermediate frequency (IF) demodulation. The baseband signals are sampled synchronously. Finally, the sampled signals are passed to a digital signal processing unit, which is realized by an FPGA of the Virtex-6 family (Xilinx, 2012b).

In contrast to the theoretical considerations, the localization unit consists of non-ideal components. Generally, antenna elements do not have omnidirectional characteristics. In addition, coupling exists between individual antenna elements (Weiss and Friedlander, 1989). The radio frequency front end results in frequency-selective phase and amplitude errors due to individual delays and attenuation of individual receiving channels. The ADC discretizes the baseband signal in amplitude and time (12 bit, 65 MSps). Due to a finite word length, quantization errors occur. The DSP operates with a finite word length as well (processing word width of 12 bit). The fixed-point arithmetic is more efficient than the floating-point arithmetic in terms of resources. These non-idealities impact the quality of a further localization result, and thus, must be considered in a preprocessing approach.

Next, an implementation of data preprocessing for a model-based parameter estimation is presented, which lays the foundation for further model-based estimation methods.

Figure 3 shows an implementation of the detector unit. The objective is the separation of the relevant information from the sampled data stream x_{ref} ; this process is denoted by step 1. The concept is taken from Schaffer (2014).

The detection is based on a sample-wise comparison of a received input sequence $x_{ref}[k]$ with a copy of the digital modulated transmission sequence $s[k]$. First, the samples are evaluated according to their signs: $x_{ref}[k] > 0 \rightarrow 1$ and

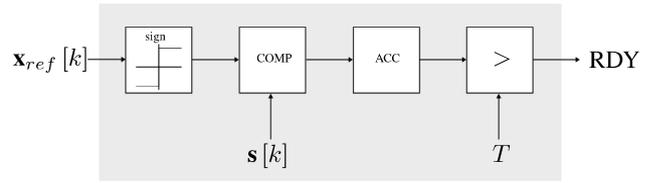


Figure 3. Block diagram of an implementation of a detector for sequence separation for a localization system.

$x_{ref}[k] \leq 0 \rightarrow 0$. The comparator COMP compares the rated input sequence with the copy of the digital modulated transmission sequence. The number of matches is accumulated in the accumulator ACC. When this sum exceeds a threshold T , the relevant information is detected, which is indicated by the output RDY. The transmit sequence is assumed to be known. This step is performed in real time. The implementation effort is limited to simple logic elements. One AND-logic is necessary to detect the sign based on the first bit of the sampled data stream (sign bit), and the other is used as a comparator COMP. The accumulator is a simple full adder composed of basic logic elements. The comparison of the sum with the threshold value T is also implemented using the AND-logic blocks.

Next, the calibration of the physical system is considered, denoted by step 2. Figure 4 presents an implementation of such a calibration in the case of wideband signals. The objective is the consideration of frequency-selective phase and amplitude errors, that result from the nature of a real receiving channel compared to the theoretical one. Generally, a distinction between computing calibration weights ($EN = 0$) and the actual calibration process ($EN = 1$) is made; this selection is provided by a multiplexer. Both parts are marked. First, the focus is on computing calibration weights denoted by step 2.1; these weights are summarized in a calibration matrix $C \in \mathbb{C}^{M \times K}$. In terms of the application, this calibration matrix is not critical during run time. We assume that thermal fluctuations can be neglected during run time. The localization system is in thermal equilibrium. In addition, it is assumed that the drift of the local oscillator can be neglected as well. Therefore, a resource-saving approach, instead of a run time saving approach is chosen. For this purpose, a second multiplexer is used. The control input CH_{sel} is used to select a single channel. Thus, it is possible to calculate the weights sequentially, channel by channel. Sampled values are available in the frequency domain. Frequency-selective amplitude and phase values are determined using the Coordinate Rotation Digital Computer (*CORDIC*) algorithm (Volder, 1959). This algorithm is an efficient implementation of an iterative algorithm for computing trigonometric functions and enables the formation of the magnitude of a complex number. The calculated calibration weights refer to a reference channel, which is initially set arbitrarily (here $x_{ref} = CH_1$). Finally, the determined values are converted into component repre-

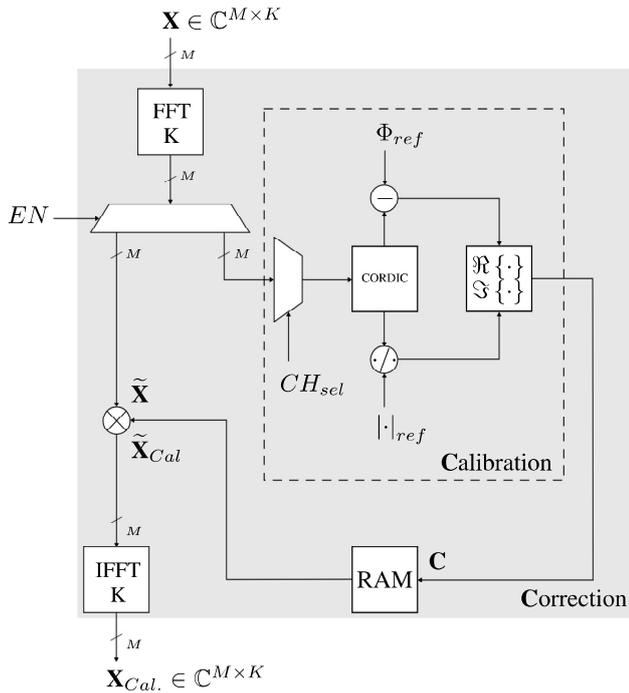


Figure 4. Block diagram of an implementation of a wideband calibration for a localization system.

sentation of complex numbers; this conversion facilitates further processing.

The calibration matrix is stored in the random-access memory (RAM). Next, the actual calibration process is considered, denoted by step 2.2. Applying calibration weights \mathbf{C} on an input data stream is critical in run time. A parallel processing is proposed. First, the uncalibrated values \mathbf{X} are transformed (FFT) (Xilinx, 2012a). The weighted frequency bins can be transformed back simultaneously. Different approaches exist for an efficient implementation. The Cooley–Tukey algorithm (Amirfattahi, 2013), also called the radix-2 algorithm, is the most common FFT algorithm. A further optimization is the use of a radix-p algorithm; an implementation of this algorithm is presented in Uzun (2003). Next, two architectures are compared: radix-2 and radix-4. The frequency-selective weighting is performed simultaneously with the output of the FFT-Block. An inverse FFT transforms the corrected data stream back into the time domain. It is advisable to use the same hardware resources for the FFT and its inverse. These parameters differ in the argument of the exponential by a sign. Thus, the real-time capability is ensured.

Finally, the channel matrix is estimated, denoted by step 3. The starting point forms Eq. (2). A channel matrix estimation by a single complex multiplier requires at least $2 \cdot M \cdot P \cdot L = 448$ DSP48E1 – blocks (cf. Virtex-6 (XC6VLX75T) 288 DSP48E1 available, for parameter see Sect. 4). The factor of 2 takes into account a semi-complex multiplier ($\mathbf{X}_{Cal.} \in \mathbb{C}$, $\mathbf{S}^\dagger \in \mathbb{R}$) consisting of two real multiplier blocks (DSP48E1).

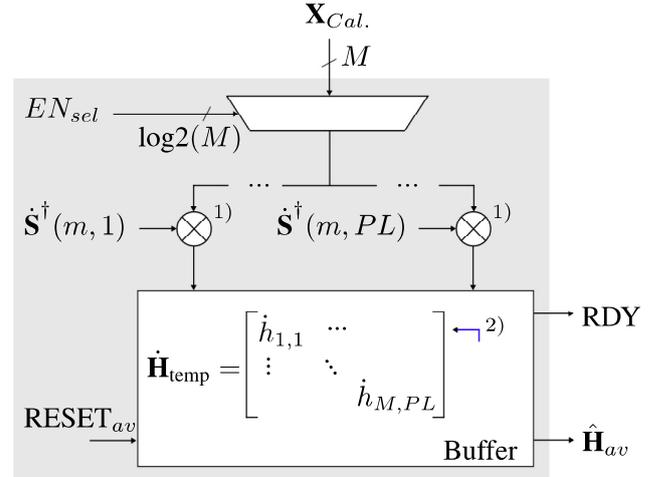


Figure 5. Block diagram of the implementation of an efficient channel matrix estimation approach for a localization system. (1) Complex hardware multiplier. (2) Line selection pointer.

At this point, a compromise is proposed. Figure 5 illustrates the block diagram of a semi-parallel implementation of the channel matrix estimation approach. The rearranged signal matrix $\mathbf{\dot{S}} \in \mathbb{R}^{P \cdot L \times P \cdot N}$ is assumed to be known. This exists in the form of a Toeplitz matrix. During run time, no calculation is necessary.

The required pseudo-inverse signal matrix $\mathbf{\dot{S}}^\dagger$ is initially stored in the RAM (for clarity, the control logic and RAM are not illustrated).

A multiplexer selects the m th row of $\mathbf{X}_{Cal.}$ by EN_{sel} . At the inputs of the complex multiplier block bank, the m th row $\mathbf{\dot{S}}^\dagger(m, :)$ is available. The multiplication result is written to the m th row of the buffered variable $\mathbf{\dot{H}}_{temp}$. The pointer (denoted by 2) specifies the location of the current calculation; this points to the m th row. The calculation result of each sample is accumulated in $\mathbf{\dot{H}}_{temp}$. According to this principle, the channel matrix is calculated sample-wise for one snapshot. There are different approaches for the implementation of a matrix multiplication on a parallel operating hardware platform (Mahendra Vucha, 2011; Qasim et al., 2010; Kumar and Tsai, 1991).

It is assumed that the complex fading amplitudes are constant during one snapshot, and that they vary from one snapshot to the next. This is valid for the assumed scenario. Thus, decorrelation is performed by averaging multiple channel matrices from different snapshots, which are uncorrelated in terms of complex fading. Alternatively, additional techniques must be applied, e.g., spatial – and temporal smoothing (Pillai and Kwon, 1989). These techniques require no additional hardware (except for a counter logic, which controls the input $RESET_{av}$ for resetting $\mathbf{\dot{H}}_{temp}$). After an initial set, the number of desired snapshots, $\mathbf{\dot{H}}_{temp}$ is reset by $RESET_{av}$ ($RDY \rightarrow 1$), which results in channel matrix $\hat{\mathbf{H}}_{av}$. This re-

Table 1. Performance evaluation for the presented implementation of a real-time capable data preprocessing for model-based parameter estimation.

		Step 1	Step 2.1	Step 2.2	Step 3
Resource ^{1,2}	Slices Register	384	7887	1067	18 917
	Slices LUT	64	6703	40	8449
	Block RAM	16	8	8	11
	DSP48E1	0	26	24	56
Runtime ^{3,2}		0.8 μ s	2.7 μ s	5.1 μ s	3.3 μ s

¹ Virtex-6 (XC6VLX75T); ² Radix-2 architecture; ³ $f_{\max} = 310$ MHz.

Table 2. Comparison of different FFT architectures in terms of a performance evaluation of the presented implementation of a real-time capable data preprocessing for model-based parameter estimation.

		radix-2	radix-4
Resource ¹	Slices Register	5293	12 408
	Slices LUT	4590	11 525
	Block RAM	8	18
	DSP48E1	24	72
Runtime ²		2.7 μ s	1.5 μ s

¹ Virtex-6 (XC6VLX75T); ² $f_{\max} = 310$ MHz.

sult forms the starting point for model-based algorithms for JADE.

4 System performance

In this section, the system performance is discussed. The results are summarized in Table 1. As an implementation platform, a *Virtex-6* FPGA (Xilinx, 2012b) is assumed. The input data stream is quantized by 12 bit in amplitude (ADC), and the internal processing word width is also assumed to be 12 bit. The results are specified for an operating frequency of $f_{\max} = 310$ MHz; this frequency is defined by the design. Furthermore a 64-bit training sequence is assumed. Generally, a linear relationship exists between the resource requirements and the input parameters M , P and L as well as latency. In the following the input parameters are assumed to be $M = 8$, $P = 4$ and $L = 7$, and the FFT is designed using a radix-2 architecture ($K = P \cdot N$).

In terms of the application, the system design is real time-capable. The FFT, which is used in step 2, determines the latency and the run time of the presented data preprocessing. If required, a part of the fast hardware multiplier DSP48E1 can be replaced by using look-up tables (LUTs). However, this approach has a decreased clock frequency f_{\max} .

In Table 2, different architectures of FFT are compared, focusing on the impact on resources and run time.

The speed advantage of using a radix-4 FFT architecture, compared to a radix-2, architecture results in an increased resource requirement, which increases disproportionately. In this case, the application cost and benefits are out of proportion, since the runtime requirement is complied with both architectures ($t_{\text{radix}} \ll 1$ ms).

5 Application

This section focuses on the application of the presented data preprocessing for the case of model-based parameter estimation. For this purpose, the incoming data \mathbf{X} are applied by frequency-selective phase and amplitude errors; these data are discrete in time and amplitude. One approach for source localization in a multipath environment is JADE-multiple signal classification (MUSIC). Similar to MUSIC, which was published by Schmidt in 1986 (Schmidt, 1986), the orthogonality between the noise subspace \mathbf{E}_n and the signal subspace \mathbf{E}_s of the correlation matrix $\hat{\mathbf{R}}_{hh} = \mathbf{h} \cdot \mathbf{h}^H$ is exploited, where H denotes the Hermitian operator. By the formation of the correlation matrix, there is an association between spatial samples in different times of the estimated channel matrix response vector $\hat{\mathbf{h}} = \text{vec}(\hat{\mathbf{H}})$. Assuming a space-time manifold $\mathbf{u}(\Theta, \tau)$, the two-dimensional spectral function results in

$$\hat{P}(\Theta, \tau) = \frac{\mathbf{u}^H \cdot \mathbf{u}}{\mathbf{u}^H \cdot \mathbf{E}_n \cdot \mathbf{E}_n^H \cdot \mathbf{u}}. \quad (3)$$

The local maxima provide information regarding the position $(\hat{\Theta}, \hat{\tau})$ of the object to be located. Figure 6 shows the two-dimensional spectral function of the JADE-MUSIC algorithm.

Data preprocessing is performed as described above. In this case, a direct path and an additional multipath are assumed. The object is located at $[-10^\circ, 2 \cdot T_s]$ and the multipath is located at $[20^\circ, 2.5 \cdot T]$. An eight-element antenna array and a 64-bit training sequence are assumed. The signal-to-noise ratio (SNR) is assumed to be 20 dB. This is consistent with the practice measurements. The multipath component is attenuated by 3 dB in terms of the direct path. In the case of a separate angle and delay estimation, a resolution of

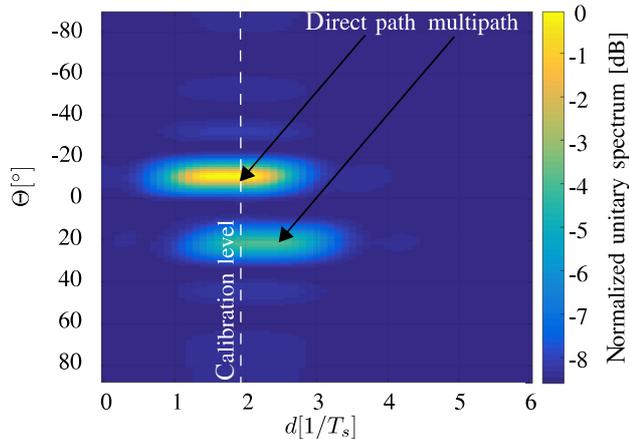


Figure 6. Spectral Function of JADE-MUSIC in the case of the presented data preprocessing approach.

the multipath component, i.e., an additional benefit of joint parameter estimation, is not possible. Thus, the paths can be resolved by using the additional angle information. The quantization by the presented design can be neglected. These results are obtained via a Monte Carlo simulation (500 runs) for the presented scenario above. In one case, the spectrum was computed using the presented fixed-point preprocessing; alternatively, a 64 bit floating-point implementation was used. The mean square errors (MSE) between these implementation variants are $\text{MSE}(\hat{\Theta}_{\text{fixed-point}}, \hat{\Theta}_{\text{floating-point}}) \approx 0$ and $\text{MSE}(\hat{d}_{\text{fixed-point}}, \hat{d}_{\text{floating-point}}) \approx 0$. For the spectrum a computation grid of $\Delta\Theta = 1.0^\circ$ and $\Delta d = 1.0\text{m}$ was assumed. Generally the impact of the modulation bandwidth of the sequence $s[n]$ or the impact of the number of antenna elements in terms of the resolution of the multipaths is much larger than the quantization errors obtained using the presented preprocessing step.

6 Conclusions

In this study, a real time-capable system design of the data preprocessing for a vehicle-based localization system used in road traffic applications was presented. As an implementation platform, an FPGA was used. The objective was the compensation of hardware imperfections compared to the theoretical consideration of a localization system. This preprocessing approach includes a separation of relevant information from the incoming data stream, a wideband calibration of the physical system and a channel estimation, as well as a decorrelation of the correlated signal components, which are contained in the incoming data stream because of a multipath assumption of the wireless channel. In this paper, the implementation aspects, including resources and run time aspects, were discussed. It was demonstrated that FFT, which is

used for a frequency-selective calibration, is computationally intensive. For this purpose, two architectures (radix-2 and radix-4) were compared. Depending on the requirements, a balance between the resources and run time was found to be required. Using the presented architecture real-time capability was achieved. Each snapshot incurred a processing time of approximately $9.2\mu\text{s}$, which was well below the requirements.

Due to the digital signal processing in fixed-point arithmetic (processing word width of 12 bit), quantization errors occurred. The used word width was found to be sufficient for a subsequent localization. The proposed implementation was demonstrated to be well-suited for the application of protecting vulnerable road users. As a next step, it will be useful to account for the errors originating from the antenna array, for example, the consideration of mutual coupling. This will also impact the channel matrix estimation and localization results.

7 Data availability

Data used in this paper is available upon request to Timo Patelczyk (timo.patelczyk@tum.de).

This work was supported by the German Research Foundation (DFG) and the Technische Universität München within the funding programme Open Access Publishing.

Edited by: J. Anders

Reviewed by: two anonymous referees

References

- Amirfattahi, R.: Calculation of Computational Complexity for Radix-2p Fast Fourier Transform Algorithms for Medical Signals, *J. Med. Signal. Sens.*, 3, 217–224, 2013.
- Biguesh, M. and Gershman, A. B.: Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals, *IEEE T. Signal. Proces.*, 54, 884–893, doi:10.1109/TSP.2005.863008, 2006.
- Cozzo, C. and Hughes, B. L.: Joint channel estimation and data detection in space-time communications, *IEEE T. Commun.*, 51, 1266–1270, 2003.
- Friedlander, B. and Weiss, A. J.: Direction finding in the presence of mutual coupling, *IEEE T. Antenn. Propag.*, 39, 273–284, doi:10.1109/8.76322, 1991.
- Jaafar, I., Amara, R., Boujemaa, H., and Siala, M.: Joint angle and delay estimation of point sources, *ICECS 2005, 12th IEEE International Conference on Electronics, Circuits and Systems*, Gammarth, Tunisia, 11–14 December 2005, 1–5, doi:10.1109/ICECS.2005.4633421, 2005.
- Kumar, V. and Tsai, Y.-C.: On synthesizing optimal family of linear systolic arrays for matrix multiplication, *IEEE T. Comput.*, 40, 770–774, doi:10.1109/12.90254, 1991.

- Mahendra Vucha, A. R.: Design and FPGA Implementation of Systolic Array Architecture for Matrix Multiplication, *Int. J. Comput. Appl.*, 26, 18–22, 2011.
- Pal, D.: Fractionally spaced semi-blind equalization of wireless channels, *Conference Record of The Twenty-Sixth Asilomar Conference on Signals, Systems and Computers*, 1992, Pacific Grove, CA, 26–28 October 1992, 642–645, doi:10.1109/ACSSC.1992.269117, 1992.
- Pillai, S. U. and Kwon, B. H.: Forward/backward spatial smoothing techniques for coherent signal identification, *IEEE T. Acoust. Speech.*, 37, 8–15, 1989.
- Schaffer, B.: Transponderbasierte Abstandsmessung für Straßenverkehrsanwendungen, *Dissertation Technische Universität München, Reihe Elektrotechnik*, Verlag Dr. Hut, 140 pp., ISBN 978-3-8439-2040-7, 2014.
- Schmidt, R. O.: Multiple emitter location and signal parameter estimation, *IEEE T. Antenn. Propag.*, 34, 276–280, 1986.
- Qasim, S. M., Telba, A. A., and AlMazroo, A. Y.: FPGA Design and Implementation of Matrix Multiplier Architectures for Image and Signal Processing Applications, *IJCSNS Int. J. Comput. Sci. Network Security*, 10, 168–176, 2010.
- Uzun, I.: FPGA implementations of fast fourier transforms for real-time signal and image processing, *2003 IEEE International Conference on Field-Programmable Technology (FPT)*, Tokyo, Japan, 15–17 December 2003, 102–109, doi:10.1109/FPT.2003.1275737, 2003.
- Van der Veen, A.-J., Talwar, S., and Paulraj, A.: Blind estimation of multiple digital signals transmitted over FIR channels, *IEEE Signal Proc. Lett.*, 2, 99–102, 1995.
- van der Veen, A.-J., Vanderveen, M., and Paulraj, A.: SI-JADE: an algorithm for joint angle and delay estimation using shift-invariance properties, *First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, Paris, France, 16–18 April 1997, 161–164, doi:10.1109/SPAWC.1997.630193, 1997.
- Volder, J. E.: The CORDIC trigonometric computing technique, *IRE T. Electron. Comput.*, 330–334, 1959.
- Weiss, A. and Friedlander, B.: Array shape calibration using sources in unknown locations—a maximum likelihood approach, *IEEE T. Acoust. Speech.*, 37, 1958–1966, doi:10.1109/29.45542, 1989.
- Xilinx: LogiCORE IP Fast Fourier Transform, Xilinx, 8.0 Edn., Application note DS808, available at: http://www.xilinx.com/support/documentation/ip_documentation/ds808_xfft.pdf, 2012a.
- Xilinx: Virtex-6 Family Overview, Xilinx, 2.4 Edn., Application note DS150, available at: www.xilinx.com/support/documentation/data_sheets/ds150.pdf, 2012b.
- Yang, B., Letaief, K. B., Cheng, R. S., and Cao, Z.: Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modeling, *IEEE T. Commun.*, 49, 467–479, doi:10.1109/26.911454, 2001.