Bachelorarbeit

zum Thema

Optimierung und Weiterentwicklung eines MIMO-Audiodemonstrators

Vorgelegt der Fakultät für Ingenieurwissenschaften der Universität Duisburg-Essen

von

Gengqi Liu

Duisburger Str. 451

45478 Mülheim an der Ruhr

Matrikelnummer 3145652

Erstgutachter/Erstgutachterin:	<Titel> <Vorname> <Name>
Zweitgutachter/Zweitgutachterin:	<Titel> <Vorname> <Name>
Studiengang:	Electrical and Electronic Engineering (ISE) (B.Sc.)
Studiensemester:	Wintersemester 25/26

Abstract

This thesis investigates ...

Contents

1 Introduction

Efficient data transmission over frequency-selective and multipath channels has long challenged digital communication systems. One widely adopted solution to this problem is *Orthogonal Frequency-Division Multiplexing* (OFDM), which mitigates inter-symbol interference (ISI) by distributing a high-rate data stream over a large number of orthogonal narrowband subcarriers and by employing a cyclic prefix [1, 2]. Due to its robustness against multipath propagation, OFDM has become a key modulation technique for broadband communication systems.

In parallel, *Multiple-Input Multiple-Output* (MIMO) technology exploits multiple transmit and receive antennas to increase channel capacity or improve link reliability without requiring additional bandwidth or transmit power. Fundamental theoretical works have shown that MIMO systems can significantly enhance spectral efficiency by exploiting spatial degrees of freedom [3, 4]. By combining MIMO with OFDM, MIMO-OFDM systems are able to jointly exploit spatial, frequency, and temporal diversity, making them particularly suitable for broadband and frequency-selective channels.

As a result of these advantages, MIMO-OFDM has been widely adopted in modern wireless communication standards, including IEEE 802.11n/ac/ax and cellular systems such as LTE and 5G [5]. Consequently, MIMO-OFDM represents not only a topic of high theoretical relevance, but also a core technology in practical communication systems. This importance makes it an essential subject in communication engineering education.

Beyond conventional radio-frequency applications, MIMO-OFDM has also been investigated for use in acoustic transmission environments, most notably in underwater acoustic communication systems. Acoustic channels are characterized by limited available bandwidth, low propagation speed, severe multipath propagation, and pronounced Doppler effects, all of which pose significant challenges for reliable high-rate communication [6]. Several studies have demonstrated that MIMO-OFDM can substantially improve spectral efficiency and robustness in such environments when appropriate synchronization and channel estimation techniques are employed [7, 8].

In the context of communication education, experimental platforms based on radio-frequency hardware are often complex and costly, and they typically offer limited possibilities for intuitive perception of the transmission process. An alternative approach is provided by acoustic MIMO demonstrators, in which loudspeakers and microphones are used instead of conventional radio transmitters and receivers. This approach enables low-cost experimentation while allowing the transmitted signals to be directly audible, thereby enhancing the intuitive understanding of signal processing algorithms.

Within the EIT study program, an existing acoustic MIMO demonstrator has been developed to illustrate fundamental concepts of MIMO signal transmission and reception. The system is implemented in MATLAB and realizes a complete signal chain from signal generation and modulation to acoustic transmission, reception, and baseband processing. The original system was designed for earlier audio hardware and employed MATLAB's GUIDE framework for the graphical user interface. While the demonstrator already enables the presentation of basic MIMO concepts, its structure and functionality exhibit limitations with respect to modern hardware interfaces, software maintainability, and extensibility.

In particular, the replacement of the original audio hardware by a modern multichannel audio interface requires adaptations in signal generation, acquisition, and channel mapping. Furthermore, the GUIDE-based graphical user interface does not support a

modular and extensible design suitable for future developments. From an algorithmic perspective, the existing demonstrator provides only limited insight into receiver-side signal processing and lacks comprehensive visualization of key performance metrics such as channel characteristics, spatial degrees of freedom, channel rank, signal-to-noise ratio (SNR), and error vector magnitude (EVM).

Against this background, the objective of this bachelor thesis is the systematic optimization and further development of the existing acoustic MIMO demonstrator. This includes adapting the software to the new hardware platform, redesigning the graphical user interface using MATLAB App Designer, and optimizing receiver-side signal processing algorithms. In addition, new analysis and visualization features are introduced to enhance the demonstrator's functionality and to provide a clearer and more comprehensive illustration of MIMO-OFDM principles for educational and experimental purposes.

2 Theoretical Background

2.1 Radio Propagation Channels and Motivation for OFDM

In practical communication environments, signal propagation between transmitter and receiver is rarely limited to a single line-of-sight path. Instead, transmitted signals are reflected, diffracted, and scattered by surrounding objects, resulting in multiple propagation paths with different delays and attenuations. This phenomenon is commonly referred to as multipath propagation. Under the assumption that the channel remains constant over the observation interval, the propagation channel can be modeled as a linear time-invariant (LTI) system [9].

In this case, the channel is fully characterized by its channel impulse response (CIR), which can be expressed as

$$h_c(t) = \sum_{l=1}^{L} h_{c,l}\, \delta(t - \tau_l), \tag{1}$$

where $h_{c,l}$ and τ_l denote the complex-valued path coefficient and propagation delay of the l-th multipath component, respectively, and L represents the total number of propagation paths. The corresponding channel transfer function (CTF) in the frequency domain is obtained by the Fourier transform of the CIR and is given by

$$H_c(\omega) = \int_{-\infty}^{\infty} h_c(t)\, e^{-j\omega t}\, \mathrm{d}t = \sum_{l=1}^{L} h_{c,l}\, e^{-j\omega\tau_l}. \tag{2}$$

This representation highlights the frequency-selective nature of multipath channels and is commonly used in the analysis of multicarrier transmission systems [2].

A statistical characterization of the multipath channel is provided by the power delay profile (PDP), defined as the expected squared magnitude of the CIR:

$$P_{\mathrm{PDP}}(\tau) = \mathbb{E}\{|h_c(\tau)|^2\}. \tag{3}$$

The PDP describes how the received signal energy is distributed over different delays and allows the definition of the maximum excess delay $\tau_{\max}$. Based on this parameter, the coherence bandwidth B_c of the channel can be approximated as

$$B_c \approx \frac{1}{\tau_{\max}}. \tag{4}$$

If the signal bandwidth B is much smaller than B_c, the channel can be regarded as frequency-flat. Conversely, if B exceeds B_c, the channel exhibits frequency-selective fading [9].

In the time domain, frequency-selective channels give rise to inter-symbol interference (ISI), since delayed replicas of previously transmitted symbols overlap with the current symbol. In single-carrier transmission systems, the mitigation of ISI typically requires sophisticated time-domain equalization techniques whose complexity increases significantly with the channel delay spread [2]. This complexity motivates the use of alternative transmission schemes that can more efficiently cope with frequency-selective channels.

The fundamental idea of multicarrier transmission is to decompose a wideband frequency-selective channel into a set of narrowband subchannels, each of which experiences approximately flat fading. Orthogonal Frequency Division Multiplexing (OFDM) represents a practical and efficient realization of this concept. By transmitting data symbols in parallel over a large number of orthogonal subcarriers, OFDM significantly increases the symbol duration on each subcarrier and thereby reduces the impact of ISI [1].

In OFDM systems, the subcarrier spacing Δf is chosen as

$$\Delta f = \frac{1}{T_s}, \tag{5}$$

where T_s denotes the useful OFDM symbol duration. This choice ensures orthogonality among the subcarriers under ideal synchronization conditions. To preserve subcarrier orthogonality in the presence of multipath propagation, a cyclic prefix (CP) is inserted at the beginning of each OFDM symbol. If the CP length is greater than or equal to the maximum channel delay spread, linear convolution with the channel impulse response is transformed into circular convolution. As a result, the frequency-domain channel matrix becomes diagonal, enabling simple one-tap equalization on each subcarrier [9, 1].

For discrete-time baseband modeling, the multipath channel is commonly represented as a finite impulse response (FIR) filter, also referred to as a tapped delay line model. This discrete-time representation provides the mathematical foundation for the implementation of OFDM systems using inverse discrete Fourier transform (IDFT) and discrete Fourier transform (DFT) operations, and it forms the basis for most practical OFDM transmitters and receivers.

2.2 Principles of OFDM Transmission

Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier transmission technique whose fundamental idea is to decompose a wideband frequency-selective channel into a set of mutually orthogonal narrowband subchannels. By doing so, each subchannel can be approximated as a frequency-flat channel, which significantly reduces the complexity of receiver-side equalization.

In practical systems, the implementation of OFDM relies on discrete-time signal processing, in particular on the discrete Fourier transform (DFT) and its fast algorithm (FFT). Compared to early multicarrier schemes based on analog oscillators, OFDM enables the realization of orthogonal subcarriers through digital signal processing, making it highly suitable for practical communication systems [1, 9].

2.2.1 Discrete-Time OFDM Signal Model

Consider a block of complex-valued data symbols

$$\{X[k]\}_{k=0}^{N-1},$$

to be transmitted over N subcarriers within one OFDM symbol. These frequency-domain symbols are mapped onto orthogonal subcarriers and transformed into the time domain by means of an inverse discrete Fourier transform (IDFT). The resulting discrete-time OFDM signal can be expressed as

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X[k]\, \mathrm{e}^{\mathrm{j}2\pi kn/N}, \quad n = 0, 1, \ldots, N-1. \tag{6}$$

The normalization factor $1/\sqrt{N}$ ensures that the average signal power is preserved during the transformation. In practice, the IDFT operation is efficiently implemented using the inverse fast Fourier transform (IFFT).

The orthogonality of the subcarriers is guaranteed by an appropriate choice of the subcarrier spacing. Let T_s denote the useful OFDM symbol duration. The subcarrier spacing is then defined as

$$\Delta f = \frac{1}{T_s}. \tag{7}$$

Under ideal synchronization conditions, this choice ensures that the demodulation of one subcarrier does not introduce interference from other subcarriers.

2.2.2 Cyclic Prefix and Circular Convolution

When an OFDM signal is transmitted over a multipath channel, the transmitted signal undergoes linear convolution with the channel impulse response. Without further measures, this linear convolution destroys the orthogonality among subcarriers and leads to inter-symbol and inter-carrier interference.

To avoid this effect, a cyclic prefix (CP) is inserted at the beginning of each OFDM symbol. The CP is generated by copying the last N_{CP} samples of the OFDM symbol and appending them to its front.

If the CP length satisfies

$$N_{\mathrm{CP}} \geq L_h - 1, \tag{8}$$

where L_h denotes the length of the discrete-time channel impulse response, the linear convolution can be transformed into circular convolution after CP removal. In this case, the received discrete-time signal can be written as

$$y[n] = x[n] \circledast h[n] + w[n], \tag{9}$$

where $\circledast$ denotes circular convolution and $w[n]$ represents additive noise [2].

2.2.3 Frequency-Domain Representation and One-Tap Equalization

Applying the discrete Fourier transform to the received OFDM symbol yields a frequency-domain signal model of the form

$$Y[k] = H[k]\, X[k] + W[k], \quad k = 0, 1, \ldots, N-1. \tag{10}$$

This expression shows that, under ideal conditions, the equivalent channel in the frequency domain exhibits a diagonal structure. Each subcarrier is affected only by the corresponding frequency-domain channel coefficient $H[k]$, which allows independent processing of each subcarrier.

A simple equalization method is the zero-forcing (ZF) equalizer, given by

$$\hat{X}[k] = \frac{Y[k]}{H[k]}, \tag{11}$$

provided that $H[k] \neq 0$. In the presence of noise, more advanced techniques such as minimum mean square error (MMSE) equalization can be employed to achieve a better trade-off between noise enhancement and distortion [2].

2.2.4 Advantages and Limitations of OFDM

The main advantage of OFDM lies in its ability to efficiently combat frequency-selective fading with relatively low receiver complexity. By converting a frequency-selective channel into multiple parallel flat-fading subchannels, OFDM enables simple frequency-domain equalization and supports flexible allocation of spectral resources [1].

Despite these advantages, OFDM also exhibits certain limitations. The superposition of many subcarriers results in a high peak-to-average power ratio (PAPR), which imposes stringent linearity requirements on the transmit power amplifier. Furthermore, OFDM systems are sensitive to synchronization errors, such as carrier frequency offset and timing misalignment. These impairments destroy subcarrier orthogonality and introduce inter-carrier interference (ICI), which must be carefully addressed in practical system implementations and experimental demonstrators [9].

2.3 Fundamentals of MIMO Systems

2.3.1 Motivation for Multi-Antenna Communication

In conventional single-input single-output (SISO) communication systems, system performance is primarily limited by channel fading and available bandwidth. In multipath propagation environments, random fluctuations in signal amplitude and phase may lead to deep fades and a significant degradation of link reliability.

Multi-antenna communication techniques address these limitations by introducing multiple antennas at the transmitter and/or receiver, thereby providing additional spatial degrees of freedom. Multiple-input multiple-output (MIMO) systems exploit these spatial dimensions to improve link reliability or increase data rates without requiring additional bandwidth or transmit power. As a result, MIMO has become a key technology in modern broadband communication systems.

2.3.2 Classification of Multi-Antenna Systems

Depending on the number of transmit and receive antennas, multi-antenna systems can be classified into four main categories: single-input single-output (SISO), single-input multiple-output (SIMO), multiple-input single-output (MISO), and multiple-input multiple-output (MIMO). While SIMO and MISO systems are primarily employed to achieve diversity or array gain, full MIMO systems are capable of supporting different transmission modes, including diversity-oriented and multiplexing-oriented schemes.

In practical system design, the choice of a specific MIMO transmission mode depends on the targeted performance objective, such as improved reliability or increased spectral efficiency.

2.3.3 Transmission Modes and Gains in MIMO Systems

Compared to single-antenna systems, MIMO systems offer several potential advantages. However, different MIMO transmission modes emphasize different performance objectives. In this work, *spatial multiplexing* is considered the primary focus, while other transmission modes are introduced for comparison and contextual understanding.

Spatial Multiplexing (Focus of This Work) Spatial multiplexing is one of the most prominent transmission modes in MIMO systems. Its key principle is the simultaneous transmission of multiple independent data streams over the same time and frequency resources. Under favorable channel conditions, spatial multiplexing enables an almost linear increase in data rate with the number of antennas, making it particularly attractive for improving spectral efficiency.

In spatial multiplexing systems, independent data streams are transmitted from different antennas and superimposed in the spatial channel. At the receiver, multi-antenna signal processing algorithms are employed to separate these streams. Typical detection techniques include zero-forcing (ZF), minimum mean square error (MMSE) detection, and successive interference cancellation schemes such as the Vertical Bell Labs Layered Space-Time (V-BLAST) architecture.

Since the acoustic MIMO demonstrator considered in this thesis aims at algorithmic analysis and visualization, spatial multiplexing allows for an intuitive illustration of channel rank, spatial degrees of freedom, and receiver detection performance. Therefore, it is selected as the primary transmission mode implemented and analyzed in this work.

Diversity Transmission and Alamouti Coding In addition to spatial multiplexing, MIMO systems can also be operated in diversity-oriented modes to enhance link reliability. A prominent example is the Alamouti space-time block code, which achieves full diversity gain for two transmit antennas while maintaining a simple linear receiver structure.

Unlike spatial multiplexing, Alamouti coding does not aim to increase the data rate but rather improves robustness against fading through redundant transmission. In this work, Alamouti transmission is mainly considered as a reference scheme to highlight the trade-off between reliability and throughput in different MIMO transmission modes.

Eigenmode Transmission and Beamforming If channel state information is available at the transmitter, the MIMO channel matrix can be decomposed into independent spatial subchannels by means of eigenvalue or singular value decomposition. This approach, commonly referred to as eigenmode transmission or beamforming, enables capacity-optimal transmission under ideal conditions.

However, eigenmode transmission requires accurate channel knowledge at the transmitter and involves increased implementation complexity. Therefore, this transmission mode is not considered for practical implementation in the demonstrator, but is briefly introduced as part of the theoretical background.

2.3.4 MIMO Channel Model and Matrix Representation

For analytical purposes, MIMO systems are commonly described using a matrix-based baseband equivalent channel model. Assuming N_{T} transmit antennas and N_{R} receive antennas, the MIMO system can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{12}$$

where $\mathbf{x} \in \mathbb{C}^{N_{\mathrm{T}} \times 1}$ denotes the transmit signal vector, $\mathbf{y} \in \mathbb{C}^{N_{\mathrm{R}} \times 1}$ the receive signal vector, $\mathbf{H} \in \mathbb{C}^{N_{\mathrm{R}} \times N_{\mathrm{T}}}$ the channel matrix, and $\mathbf{n}$ additive noise.

This model captures the fundamental property of MIMO systems, namely that multiple transmitted signals are linearly superimposed by the spatial channel and jointly processed at the receiver.

2.3.5 Channel Rank and Spatial Degrees of Freedom

An essential parameter in MIMO systems is the rank of the channel matrix. The channel rank determines the number of independent spatial data streams that can be transmitted simultaneously and is bounded by

$$\mathrm{rank}(\mathbf{H}) \leq \min(N_{\mathrm{T}}, N_{\mathrm{R}}). \tag{13}$$

In spatial multiplexing systems, the channel rank directly limits the achievable multiplexing gain. A full-rank channel allows the transmission of the maximum number of independent data streams, whereas a reduced rank leads to a loss of spatial degrees of freedom. Consequently, the channel rank plays a central role in the analysis of spatial multiplexing performance and serves as a key metric in the evaluation of MIMO demonstrators.

2.3.6 Summary and Relation to Subsequent Chapters

This section introduced the fundamental concepts of multi-antenna communication with a focus on spatial multiplexing as the primary transmission mode. Diversity-oriented schemes such as Alamouti coding and eigenmode transmission were discussed for comparison. The matrix-based MIMO channel model and the concept of channel rank were presented as essential tools for analyzing spatial multiplexing systems.

In the following sections, these concepts are combined with OFDM transmission, leading to a per-subcarrier MIMO signal model and forming the basis for channel estimation, receiver-side detection algorithms, and performance evaluation in MIMO-OFDM systems.

2.4 MIMO-OFDM Signal Model

2.4.1 Conceptual Combination of MIMO and OFDM

As introduced in the previous sections, OFDM enables the decomposition of a frequency-selective wideband channel into a set of parallel frequency-flat subchannels, while MIMO systems exploit spatial degrees of freedom by employing multiple transmit and receive antennas. The combination of these two techniques results in a MIMO-OFDM system, which has become the standard transmission scheme for modern broadband communication systems [1, 10].

The key idea of MIMO-OFDM is to apply a narrowband MIMO channel model independently to each OFDM subcarrier. Since OFDM converts a frequency-selective channel into multiple approximately flat-fading subchannels, the complex wideband MIMO channel can be represented as a collection of parallel narrowband MIMO channels in the frequency domain [2, 9]. This per-subcarrier modeling approach significantly simplifies the analysis and receiver design of broadband MIMO systems.

2.4.2 Discrete-Time Transmit Model

Consider a MIMO-OFDM system with N_{T} transmit antennas and N_{R} receive antennas employing N subcarriers per OFDM symbol. In spatial multiplexing mode, N_{S} independent data streams are transmitted simultaneously, where

$$N_{\mathrm{S}} \leq \min(N_{\mathrm{T}}, N_{\mathrm{R}}).$$

On the k-th subcarrier, the transmit signal can be represented by the vector

$$\mathbf{x}[k] \in \mathbb{C}^{N_{\mathrm{T}} \times 1},$$

whose elements correspond to the complex-valued symbols transmitted from the individual antennas on that subcarrier. After OFDM modulation, including IFFT operation and cyclic prefix insertion, the time-domain signals of all transmit antennas are emitted simultaneously and propagated through the spatial channel.

2.4.3 Per-Subcarrier MIMO Channel Model

At the receiver, after cyclic prefix removal and FFT processing, the received signal on the k-th subcarrier can be expressed as the vector

$$\mathbf{y}[k] \in \mathbb{C}^{N_{\mathrm{R}} \times 1}.$$

Under ideal synchronization conditions, the MIMO-OFDM system on each subcarrier is described by the linear model

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{x}[k] + \mathbf{n}[k], \tag{14}$$

where $\mathbf{H}[k] \in \mathbb{C}^{N_{\mathrm{R}} \times N_{\mathrm{T}}}$ denotes the frequency-domain MIMO channel matrix on the k-th subcarrier and $\mathbf{n}[k]$ represents additive noise.

This expression constitutes the standard per-subcarrier signal model used for the analysis of MIMO-OFDM systems and is widely adopted in the literature and textbooks on multicarrier and multi-antenna communications [2, 10, 9]. It highlights that OFDM transforms the convolutional wideband MIMO channel into a set of independent linear MIMO systems in the frequency domain.

2.4.4 Structure and Physical Interpretation of the Channel Matrix

Each element $H_{ij}[k]$ of the channel matrix $\mathbf{H}[k]$ represents the complex frequency response between the j-th transmit antenna and the i-th receive antenna on the k-th subcarrier. The channel matrix therefore captures the combined effects of multipath propagation, antenna configuration, and the spatial characteristics of the propagation environment.

In spatial multiplexing systems, the rank of $\mathbf{H}[k]$ determines the number of independent spatial data streams that can be transmitted on the k-th subcarrier. A full-rank channel matrix allows the system to fully exploit spatial degrees of freedom, whereas a rank-deficient channel limits the achievable spatial multiplexing gain. As a consequence, variations of the channel rank across subcarriers directly influence the frequency-dependent performance of MIMO-OFDM systems [10].

2.4.5 Detection Problem in MIMO-OFDM Systems

In spatial multiplexing mode, the primary task of the receiver is to recover the transmit vector $\mathbf{x}[k]$ from the received signal $\mathbf{y}[k]$. This task is commonly referred to as MIMO detection.

Due to the coupling of multiple data streams through the channel matrix $\mathbf{H}[k]$, MIMO detection constitutes a multidimensional signal separation problem. Depending on the employed algorithm, different trade-offs between computational complexity and detection performance arise. Linear detection schemes, such as zero-forcing (ZF) and minimum mean square error (MMSE) detection, offer low complexity, while successive interference cancellation techniques, including the Vertical Bell Labs Layered Space-Time (V-BLAST) algorithm, generally achieve improved performance at the cost of increased complexity [10].

In MIMO-OFDM systems, the detection process is typically performed independently on each subcarrier, which allows the wideband detection problem to be decomposed into a set of parallel narrowband MIMO detection problems.

2.4.6 Summary and Relation to Subsequent Chapters

This section established a unified signal model for MIMO-OFDM systems based on a per-subcarrier representation. The structure and physical interpretation of the frequency-domain channel matrix were discussed, and the role of channel rank in spatial multiplexing systems was highlighted. The resulting signal model provides the theoretical foundation for channel estimation, equalization, and MIMO detection algorithms, which are addressed in the subsequent sections.

2.5 Channel Estimation and Equalization in MIMO-OFDM Systems

2.5.1 Role of Channel Estimation in MIMO-OFDM

As established in the previous section, the received signal on the k-th subcarrier of a MIMO-OFDM system can be expressed as

$$\mathbf{y}[k] = \mathbf{H}[k]\mathbf{x}[k] + \mathbf{n}[k]. \tag{15}$$

In order to reliably recover the transmit vector $\mathbf{x}[k]$, knowledge of the corresponding channel matrix $\mathbf{H}[k]$ is required at the receiver. Consequently, channel estimation constitutes a key component of MIMO-OFDM receivers, and its accuracy has a direct impact on equalization and detection performance [2, 10].

Due to the OFDM structure, channel estimation can be performed independently on each subcarrier in the frequency domain, which significantly reduces computational complexity compared to time-domain approaches for wideband systems.

2.5.2 Training-Based Channel Estimation

One of the most fundamental channel estimation approaches is based on known training sequences or preambles transmitted prior to data transmission. By exploiting the knowledge of the transmitted symbols, the receiver can directly estimate the channel response from the received signal.

For a MIMO-OFDM system, assuming that a known training symbol matrix $\mathbf{X}[k]$ is transmitted on the k-th subcarrier, the received signal can be written as

$$\mathbf{Y}[k] = \mathbf{H}[k]\mathbf{X}[k] + \mathbf{N}[k]. \tag{16}$$

If $\mathbf{X}[k]$ is full rank and known at the receiver, a least-squares (LS) estimate of the channel matrix is given by

$$\hat{\mathbf{H}}[k] = \mathbf{Y}[k]\mathbf{X}^{-1}[k]. \tag{17}$$

This approach is simple to implement and provides reliable performance at sufficiently high signal-to-noise ratios. Therefore, training-based channel estimation is well suited for experimental platforms and teaching-oriented demonstrators [9].

2.5.3 Pilot-Based Channel Estimation

In practical systems with time-varying channels, relying solely on preamble-based estimation is often insufficient. To enable continuous channel tracking, pilot symbols are periodically embedded into the transmitted data stream.

In OFDM systems, pilots can be arranged in the time domain, frequency domain, or both. The channel response is first estimated at the pilot positions and subsequently interpolated across data subcarriers. In MIMO-OFDM systems, pilot design must additionally ensure orthogonality among transmit antennas to avoid pilot contamination and to maintain channel identifiability [1].

2.5.4 Frequency-Domain Equalization

Once an estimate $\hat{\mathbf{H}}[k]$ of the channel matrix is available, the receiver compensates for channel-induced distortions by means of equalization. Owing to the cyclic prefix, the equivalent frequency-domain channel exhibits a per-subcarrier matrix structure, which allows equalization to be performed independently on each subcarrier.

In spatial multiplexing mode, equalization aims to construct a matrix $\mathbf{W}[k]$ such that the estimate

$$\hat{\mathbf{x}}[k] = \mathbf{W}[k]\mathbf{y}[k] \tag{18}$$

approximates the original transmit vector $\mathbf{x}[k]$ as accurately as possible.

2.5.5 Linear Equalization: ZF and MMSE

The most commonly employed equalization techniques are linear equalizers, in particular zero-forcing (ZF) and minimum mean square error (MMSE) equalization.

The ZF equalizer completely eliminates inter-stream interference by inverting the channel matrix, yielding

$$\mathbf{W}_{\mathrm{ZF}}[k] = \hat{\mathbf{H}}^{-1}[k], \tag{19}$$

provided that the channel matrix is invertible. While ZF equalization achieves perfect interference suppression in the absence of noise, it may significantly amplify noise under unfavorable channel conditions.

In contrast, MMSE equalization accounts for noise by balancing interference suppression and noise enhancement. The MMSE equalization matrix is given by

$$\mathbf{W}_{\mathrm{MMSE}}[k] = \left(\hat{\mathbf{H}}^{\mathrm{H}}[k]\hat{\mathbf{H}}[k] + \sigma_n^2\mathbf{I}\right)^{-1}\hat{\mathbf{H}}^{\mathrm{H}}[k], \tag{20}$$

where σ_n^2 denotes the noise variance. MMSE equalization generally outperforms ZF at low and moderate signal-to-noise ratios [10].

2.5.6 Successive Interference Cancellation and V-BLAST

To further improve detection performance in spatial multiplexing systems, successive interference cancellation (SIC) techniques can be applied. A prominent example is the Vertical Bell Labs Layered Space-Time (V-BLAST) architecture.

The core idea of V-BLAST is to detect the data stream with the highest post-detection signal-to-noise ratio first, subtract its contribution from the received signal, and subsequently detect the remaining streams. By iteratively reducing inter-stream interference, V-BLAST achieves improved performance compared to purely linear detection methods, while maintaining manageable computational complexity [10].

In MIMO-OFDM systems, V-BLAST detection is typically performed independently on each subcarrier, preserving the modular structure of OFDM-based receivers.

2.5.7 Performance Metrics for Receiver Evaluation

Following channel estimation, equalization, and detection, quantitative performance metrics are required to evaluate receiver quality. One of the most fundamental measures is the signal-to-noise ratio (SNR), defined as the ratio between signal power and noise power:

$$\mathrm{SNR} = \frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}. \tag{21}$$

In MIMO-OFDM systems, SNR can be defined at different stages of the receiver chain and provides an important indication of channel conditions and operating points.

Another widely used metric is the error vector magnitude (EVM), which quantifies the deviation between received symbols and their ideal reference positions. For the k-th subcarrier, the error vector is defined as

$$\mathbf{e}[k] = \hat{\mathbf{x}}[k] - \mathbf{x}[k]. \tag{22}$$

The root-mean-square EVM is then given by

$$\mathrm{EVM} = \sqrt{\frac{\mathbb{E}\{\|\mathbf{e}[k]\|^2\}}{\mathbb{E}\{\|\mathbf{x}[k]\|^2\}}}. \tag{23}$$

Unlike bit error rate (BER), EVM directly reflects symbol-level distortions and is particularly sensitive to residual interference, noise, and synchronization errors. Therefore, EVM is well suited for analyzing modulation quality and receiver performance in experimental and demonstrator-based systems [2].

In addition, constellation diagrams provide an intuitive visualization of received symbol distributions in the complex plane and complement quantitative performance metrics. For spatial multiplexing systems, the rank of the channel matrix further serves as an important indicator of available spatial degrees of freedom and achievable multiplexing performance [10].

2.5.8 Summary and Relation to System Evaluation

This section discussed channel estimation and equalization techniques for MIMO-OFDM systems, including training-based and pilot-based estimation as well as ZF, MMSE, and V-BLAST detection methods. Furthermore, relevant performance metrics such as SNR, EVM, constellation diagrams, and channel rank were introduced. These concepts form the theoretical basis for the performance evaluation and experimental analysis presented in subsequent chapters.

3 Methodology: System Adaptation and Development

This chapter describes the concrete methods used to achieve the three core objectives defined in the thesis assignment: (i) adapting the existing software package to a new multi-channel audio interface, (ii) porting and restructuring the GUI from GUIDE to MATLAB App Designer, and (iii) optimizing and extending the receiver signal-processing chain for MIMO-OFDM demonstrations. The focus is on reproducible implementation details: how the audio hardware is interfaced from MATLAB, how data and parameters are propagated through the application, and how the physical-layer processing pipeline is organized and parameterized.

3.1 System-level workflow and design principles

The acoustic MIMO demonstrator follows a closed-loop transmission and reception workflow that is explicitly designed for algorithm analysis and teaching-oriented experimentation. In a typical experiment, the user first configures a set of physical-layer (PHY) parameters via the graphical user interface (GUI). These parameters determine the waveform structure and signal processing at the physical layer, including OFDM framing, modulation order, MIMO transmission mode, channel estimation strategy, and equalization method. Based on this configuration, a multi-channel transmit waveform is generated, played back through the audio interface and loudspeaker array, recorded by a multi-channel microphone setup, and subsequently processed at the receiver to recover the transmitted payload and evaluate system performance.

In the context of this work, the term *physical-layer (PHY) parameters* refers to all configuration parameters that directly affect waveform generation and receiver signal processing at the sample and symbol level. This includes, among others, the FFT size and cyclic prefix length of the OFDM system, the modulation order, the number of transmit and receive channels, the selected MIMO transmission scheme, as well as the channel estimation and equalization algorithms. These parameters jointly define the mathematical signal model described in Chapter 2 and are therefore grouped under the PHY abstraction commonly used in digital communication systems [2, 10].

Unlike a strict real-time communication prototype, the demonstrator is intentionally designed for *offline* or *nearline* receiver processing. While the transmission and recording of audio signals are performed in a streaming fashion, the receiver signal processing is executed only after a complete frame has been acquired. Consequently, no hard real-time deadlines are imposed on the execution of synchronization, channel estimation, equalization, or decoding algorithms. This design choice allows the use of computationally intensive processing steps—such as FFT-based OFDM demodulation, singular value decomposition in Eigenmode operation, or successive interference cancellation in

V-BLAST—without risking system instability. At the same time, it provides a robust and reproducible experimental environment that is particularly well suited for teaching and algorithm comparison.

The audio input/output loop itself is nevertheless required to operate reliably during streaming. Playback and recording are implemented using frame-based processing with a configurable frame length. In this context, *underruns* and *overruns* may occur if the host system is temporarily unable to supply playback samples or retrieve recorded samples fast enough. Such effects are well known in non-real-time audio systems running on general-purpose operating systems and are primarily influenced by buffer sizes, computational load, and operating system scheduling [11, 12]. In the demonstrator, underruns and overruns are explicitly monitored and reported but do not abort the experiment. This behavior reflects the educational focus of the system: maintaining continuity of operation and transparency about system limitations is prioritized over strict real-time guarantees.

A central architectural principle of the software design is the strict separation between GUI orchestration and signal processing functionality. The GUI layer is responsible for user interaction, parameter validation, execution control, and visualization of results. In contrast, waveform generation, audio input/output, and receiver processing are implemented as standalone MATLAB functions that operate on explicitly defined input structures. Communication between the GUI and the processing layer is realized through well-defined parameter and result structures, rather than through implicit shared state. This modular organization improves code readability and maintainability, simplifies debugging, and allows individual components—such as channel estimators, equalizers, or visualization modules—to be extended or replaced without affecting the overall system structure. As a result, the demonstrator can evolve alongside future hardware upgrades and algorithmic extensions while retaining a clear and robust software architecture.

3.2 Objective 1: Adapting the software to the new multi-channel audio interface

The first objective of this thesis is the adaptation of the existing software package to a fundamentally updated hardware platform. In contrast to the previous demonstrator setup, which relied on a collection of loosely integrated measurement and audio components, the new system is built around a modern, fully integrated multi-channel audio interface in combination with higher-quality loudspeakers and microphones. This hardware upgrade represents a qualitative improvement of the demonstrator rather than a limiting constraint, but it also changes several implicit assumptions made by the legacy software.

3.2.1 Hardware platform upgrade and its implications

The core of the new setup is a Focusrite Scarlett 18i20 audio interface, which provides multiple synchronized analog input and output channels using a single device clock. From a MIMO signal processing perspective, this is a key advantage, as it ensures sample-accurate synchronization between all transmit and receive channels. Compared to the previous solution, which required external power supplies and separate signal routing via a National Instruments BNC-2110 interface, the new platform significantly reduces wiring complexity and potential sources of channel mismatch.

In addition to the audio interface, the demonstrator employs Genelec 8010 active loudspeakers and t.bone SC140 condenser microphones. The Genelec loudspeakers are

Table 1: Comparison between the legacy and the upgraded hardware platforms used in the acoustic MIMO demonstrator.

Aspect	Legacy setup	Upgraded setup
System integration	NI BNC-2110 with external power supply	Integrated multi-channel audio interface
Clock synchronization	Implicit, wiring-dependent	Hardware-level common device clock
ADC/DAC quality	Basic measurement grade	High-quality audio-grade converters
Loudspeakers	Generic consumer-grade	Genelec 8010 near-field monitors
Microphones	Generic microphones	t.bone SC140 condenser microphones
Linearity and distortion	Limited, noticeable nonlinear effects	Improved linearity and reduced distortion
Software assumptions	Fixed and partially hard-coded	Fully parameterized and configurable

Table 2: Software-level audio I/O parameters defining the interface between MATLAB and the audio hardware.

Parameter	Meaning
f_s (`fs`)	Audio sampling rate (common device clock)
N_T (`Nt`)	Number of transmit channels (loudspeakers)
N_R (`Nr`)	Number of receive channels (microphones)
L (`frameLen`)	Frame length (samples per I/O call)
`deviceName`	Driver/device identifier (platform-dependent)

designed for near-field monitoring and exhibit a largely linear frequency response and low distortion within the relevant audio band. This improves the reproducibility of the acoustic transmission channel and reduces hardware-induced nonlinear effects that would otherwise mask the behavior of MIMO-OFDM algorithms. The t.bone SC140 microphones provide a comparatively low noise floor and consistent sensitivity characteristics across channels, which is particularly important for multi-channel channel estimation and spatial processing.

Overall, the upgraded hardware platform offers improved linearity, channel consistency, and synchronization accuracy. As a result, observed impairments in the received signal are more closely related to the propagation environment and the selected signal processing algorithms, rather than to limitations of the hardware itself.

3.2.2 Software-level audio I/O interface and channel mapping

To interface the upgraded hardware from MATLAB, the software employs the `audioPlayerRecorder` System object, which supports synchronous multi-channel playback and recording on a single device [11]. This interface exposes a small set of software-level parameters that define the contract between the signal processing code and the audio hardware, including the sampling rate, the number of active transmit and receive channels, and the frame length used for block-based streaming.

In the implementation (`runScarlettMimoIO`), the playback and recording channel mappings are explicitly defined as `1:Nt` and `1:Nr`, respectively. This explicit mapping replaces the implicit and partially hard-coded assumptions of the legacy software and ensures that experimental configurations remain reproducible across different devices and operating systems.

3.2.3 Frame-based streaming, underrun/overrun monitoring, and fallback mode

Audio playback and recording are performed using a frame-based streaming approach. The transmit signal $\mathbf{x}_{\text{tx}} \in \mathbb{R}^{N_{\text{samples}} \times N_T}$ is segmented into frames of length L and streamed through the audio interface. Each I/O call returns a recorded frame $\mathbf{y}_{\text{rx}} \in \mathbb{R}^{L \times N_R}$, together with diagnostic indicators for underruns and overruns. These events occur when the host system is temporarily unable to supply or retrieve audio data at the required rate, a well-known effect in non-real-time audio systems running on general-purpose operating systems [11, 12].

In the demonstrator, underruns and overruns are explicitly monitored and reported, but they do not abort the experiment. This design choice reflects the educational focus of the system: maintaining continuity of operation and transparency about I/O quality is preferred over enforcing strict real-time constraints. If the audio device cannot be opened, the software automatically falls back to an ideal loopback mode, in which the received signal is copied directly from the transmitted signal. This fallback mode enables debugging and algorithm development without requiring access to the physical hardware.

3.2.4 Amplitude protection and traceability via audio logging

To prevent clipping and hardware saturation, the transmit signal is subjected to amplitude protection prior to playback. The signal is constrained to real-valued samples, normalized to unit peak amplitude, and scaled by a fixed linear gain to provide additional headroom. For traceability and reproducibility, the transmitted and received multi-channel audio signals are saved as waveform files. These recordings allow offline inspection of the raw audio data and facilitate repeatable evaluation of system behavior across different experiments.

3.3 Objective 2: Porting and restructuring the GUI using MATLAB App Designer

3.3.1 Motivation for migration and architectural objectives

The original version of the MIMO audio demonstrator relied on MATLAB GUIDE for graphical user interface development. Since MATLAB GUIDE has been deprecated and removed from recent releases, migration to MATLAB App Designer became necessary to ensure long-term maintainability. [13, 14].

The GUI was redesigned as an orchestration layer for the physical-layer signal processing chain, rather than a direct replication of the legacy interface. In particular, the GUI was intended to:

- manage all system and PHY parameters in a centralized and consistent manner,
- control the execution order of transmission, reception, and processing steps, and

- provide controlled access to intermediate results and performance metrics.

As a result, the GUI becomes an integral part of the experimental methodology rather than a passive visualization tool.

3.3.2 Event-driven workflow and separation of concerns

The GUI is implemented as a class-based MATLAB App Designer application. User interactions are handled through event-driven callbacks, which explicitly separate control logic from signal-processing algorithms. A key methodological principle is that GUI callbacks do not perform numerical signal processing themselves. Instead, they coordinate the workflow by validating prerequisites, assembling parameter structures, invoking processing functions, and storing the resulting data.

This design is exemplified by the `StartProcessingButtonPushed` callback. When triggered, the callback first checks whether a received signal is available, then collects all relevant system parameters and user-selected algorithm options into a single structured variable, `procParam`. This structure is subsequently passed to the receiver processing function `Signalverarbeitung_app`, which executes the complete PHY processing chain.

All outputs of the receiver processing are stored in the centralized application state variable `app.RxAnalysisData`. This variable acts as the single source of truth for subsequent visualization steps, such as constellation diagrams, EVM plots, BER displays, and recovered payload previews. By enforcing this centralized storage concept, the GUI avoids hidden dependencies between callbacks and ensures reproducible access to processing results.

3.3.3 Global parameter management and state consistency

All system parameters are maintained as private properties of the app object. These include OFDM parameters (FFT length, cyclic prefix length, number of blocks), MIMO configuration (number of transmit and receive channels, transmission mode), modulation and coding settings, as well as frequency and sampling-rate parameters.

Each GUI control element (numeric edit fields and drop-down menus) is connected to a dedicated `ValueChangedFcn` callback. These callbacks perform basic input validation and immediately synchronize the corresponding internal property. This approach ensures that the application state remains consistent at all times, and that parameter changes are propagated throughout the system in a controlled manner.

For example, changes in modulation order or antenna configuration automatically trigger a recomputation of the recommended number of OFDM blocks. Similarly, changes to the antenna configuration invalidate previously estimated EigenMode channel information, which is cleared from the app state to enforce a new channel sounding procedure.

3.3.4 GUI-driven configuration of transmission and reception methods

The GUI provides explicit controls for selecting transmission schemes, channel estimators, and equalization strategies. Supported MIMO modes include spatial multiplexing, Alamouti coding, V-BLAST, and EigenMode transmission. Channel estimation and equalization options are selected via drop-down menus and stored as string identifiers within the app state.

Table 3: Representative GUI features supporting methodical experimentation.

Feature group	Implemented functionality
System parameters	FFT size, CP length, N_T/N_R, carrier and sampling rate
Transmission modes	Spatial multiplexing, Alamouti, EigenMode, V-BLAST
Receiver options	Channel estimation and equalization selection
Execution control	Separate triggers for transmission and Rx processing
Analysis tools	Channel responses, constellations, EVM, BER, rank metrics

Table 4: Key fields of `procParam` used for receiver processing.

Field	Description
`rxSignal`	Recorded signal matrix $[N_{\text{samp}} \times N_R]$
`fs`	Sampling rate (f_s)
`iNfft,iNg,iNb`	OFDM parameters (FFT, CP, block length)
`iNoTxAnt,iNoRxAnt`	Number of transmit and receive channels
`iModOrd`	Modulation order (bits per symbol)
`mimoMode`	Selected MIMO transmission scheme
`channelEstimator`	Channel estimation method
`equalizerMode`	Equalization/detection method
`DatenTyp,SendeDatei`	Payload type and original data

Instead, they are forwarded unchanged to the processing functions via structured parameter passing. This approach allows the receiver implementation to remain independent of the GUI while still enabling flexible experimentation with different algorithmic configurations.

3.3.5 Structured parameter interface via `procParam`

The structured variable `procParam` defines a stable interface between the GUI layer and the signal-processing backend. It encapsulates all information required for receiver processing, including the recorded waveform, system parameters, algorithm selections, and payload metadata. This design avoids implicit dependencies and allows the processing functions to be tested independently of the GUI.

For EigenMode operation, additional matrices obtained from channel sounding, such as precoding and decoding matrices and singular values, are included in `procParam`. This ensures that the receiver processing stage has access to all necessary channel-dependent information without relying on global variables.

3.3.6 Methodological implications

The event-driven GUI design enforces a clear separation between user interaction and signal-processing logic. This architecture supports systematic experimentation under well-defined conditions and allows individual components of the processing chain to be extended or replaced without modifying the GUI logic itself.

3.4 Objective 3: Algorithmic design and optimization of the end-to-end PHY pipeline

This section describes the algorithmic structure of the implemented physical-layer (PHY) processing chain. The focus is on the design methodology and implementation choices rather than on performance evaluation, which is deferred to later chapters. The system is implemented as an end-to-end acoustic MIMO–OFDM demonstrator with multiple transmission modes and extensive intermediate observability for teaching and debugging purposes.

3.4.1 Design objectives and architectural principles

The PHY processing pipeline is designed according to three main principles. First, the system must operate robustly under practical acoustic audio I/O conditions, where unknown delays, resampling artifacts, and carrier-frequency offsets are unavoidable. Second, a unified OFDM front-end is maintained across all MIMO modes (spatial multiplexing, V-BLAST, Alamouti, and Eigenmode), such that synchronization and OFDM processing are shared while mode-specific operations are isolated. Third, the implementation explicitly exposes intermediate signals and parameters (e.g., synchronization metrics, channel responses, equalized symbols, rank, and EVM) to support didactic analysis and debugging within the graphical user interface.

3.4.2 Transmit-side processing: payload framing and symbol generation

Transmit signal generation is orchestrated by `generateTxSequence_app` and finalized by a mode-specific transmitter function. A maximum number of payload bits per frame is determined by the modulation order, number of transmit antennas, number of OFDM blocks, and FFT size. Payloads shorter than this budget are zero-padded, while longer payloads are truncated to a single frame.

Payload bits are reshaped into groups of $iModOrd$ bits and converted to symbol indices prior to QAM modulation. Different bit-order conventions are intentionally applied: image payloads use a left-most-significant-bit convention to match image packing routines, while text payloads retain a legacy right-most-significant-bit convention to preserve compatibility with an existing decoding path.

QAM modulation is performed using unit-average-power constellations, followed by a normalization factor of $1/\sqrt{N_T}$ to ensure that total transmit power remains independent of the number of active transmit antennas.

3.4.3 Control information and header handling

For text transmission, a control header containing payload length information is inserted in a highly robust manner. Specifically, the transmitter overwrites the leading symbol positions of the serialized QAM stream with BPSK-modulated header bits. This design prioritizes reliable header recovery even under severe channel conditions. In contrast, for image transmission the control information is embedded directly into the QAM bitstream and no symbol overwriting is performed.

Implementation note: although convolutional coding is applied to the control header at the transmitter, the current receiver implementation does not perform convolutional decoding. Instead, the true payload length is passed as metadata from the transmitter to

the receiver. This approach ensures correct payload reconstruction in the demonstrator but does not yet represent a fully autonomous header-decoding design. A complete system would require explicit Viterbi decoding and header parsing at the receiver.

3.4.4 Training and synchronization signal design

Two known training signals are inserted to stabilize receiver processing. A Chu (CAZAC-like) preamble of length $iNfft$ is used for frequency-domain channel estimation. The preamble is transmitted as dedicated OFDM symbols and enables per-subcarrier channel estimation at the receiver.

In addition, a Schmidl–Cox synchronization symbol without cyclic prefix is inserted before the payload. This symbol consists of two identical halves in the time domain and enables robust frame-start detection and coarse carrier-frequency offset estimation. The same principle is applied across all MIMO modes, although the exact symbol generation differs slightly between implementations.

3.4.5 Mode-specific OFDM framing and MIMO mapping

After QAM mapping, the OFDM frame is constructed according to the selected MIMO mode. For spatial multiplexing and V-BLAST, the transmitter inserts one preamble OFDM symbol per transmit antenna before each data subframe. A block of trailing zero OFDM symbols is appended to support noise power estimation at the receiver. Cyclic prefix insertion is implemented robustly, including the edge case where the prefix length exceeds the FFT size.

For Alamouti transmission, frequency-domain Alamouti encoding is applied, producing two time slots per data block. A fixed frame structure with five OFDM symbols per block is used, consisting of antenna-specific preambles, two Alamouti data slots, and a null symbol. This deterministic structure simplifies receiver-side parsing.

For Eigenmode transmission, frequency-domain precoding is applied using precomputed per-subcarrier precoding matrices. After precoding, the OFDM framing follows the same structure as spatial multiplexing, ensuring a common receiver front-end.

3.4.6 Receiver front-end: downconversion, synchronization, and OFDM extraction

Receiver processing begins with passband-to-baseband conversion, where each recorded audio channel is independently downconverted, filtered, and resampled. Minor length mismatches caused by resampling are resolved by truncation to a common minimum length across channels.

Frame synchronization is performed independently per receive channel using a Schmidl–Cox metric. The earliest detected frame start across all channels is selected to ensure consistent MIMO processing. A small empirical safety margin is applied to compensate for practical alignment uncertainties. Coarse carrier-frequency offset estimates are applied via complex exponential compensation.

Implementation note: the carrier-frequency offset estimation and compensation follow a legacy scaling convention that has been empirically validated for the demonstrator. A future refinement should unify the CFO definition and compensation formula within a single normalized framework.

Following synchronization, the received baseband signal is segmented into OFDM blocks, reshaped into a three-dimensional array, and stripped of its cyclic prefix. This common OFDM front-end is shared across all MIMO modes.

3.4.7 Channel estimation and equalization

Mode-specific receiver functions perform channel estimation and equalization. Noise power is estimated using trailing zero OFDM symbols, and a per-subcarrier signal-to-noise ratio proxy is derived.

Channel estimation is performed using the known preamble symbols. Two estimators are implemented: a zero-forcing (least-squares) estimator based on direct division by the preamble spectrum, and a scalar MMSE estimator that incorporates the estimated noise power. Channel impulse responses are additionally computed via inverse FFT for inspection and visualization.

Equalization is carried out on a per-subcarrier basis. Linear zero-forcing and MMSE equalizers are implemented for spatial multiplexing. For V-BLAST operation, successive interference cancellation is applied using a per-subcarrier detection order and hard-decision feedback. Eigenmode reception applies equalization to the effective precoded channel and optionally includes singular-value-based scaling.

A decision-directed common phase error correction stage is applied after equalization to mitigate residual phase rotation that is not removed by coarse frequency-offset compensation.

3.4.8 Symbol demapping and payload reconstruction

After equalization, symbols are serialized and demapped to bits. Image payloads are uniformly demapped using M-QAM, while text payloads apply BPSK demodulation to the header region followed by QAM demodulation of the payload. Bit-order conventions are matched to the transmitter configuration.

Text reconstruction is performed by removing header bits, truncating to the known payload length, enforcing byte alignment, and converting bytes to ASCII characters. Image reconstruction follows an analogous inverse mapping.

Current limitation: although convolutional coding is supported at the transmitter, the receiver currently performs no channel decoding. Consequently, the coding mode influences the transmitted bitstream but does not yet provide full coding gain. Integrating channel decoding is identified as a necessary extension for a complete PHY implementation.

3.4.9 Reproducibility and traceability within the processing chain

To ensure that the described methods can be systematically analyzed and reproduced, the implementation incorporates explicit traceability mechanisms at several stages of the processing chain. Transmit signal generation uses a fixed random seed, ensuring that symbol mapping and padding behavior remain identical across repeated runs with the same parameters.

Raw transmit and receive waveforms are logged to disk, enabling offline inspection of the complete audio I/O path independently of subsequent processing steps. In addition, the receiver stores a structured snapshot containing decoded payloads, error metrics,

and intermediate PHY results such as synchronization metrics, channel estimates, and equalized symbols.

These measures support controlled experimentation and allow individual processing stages to be verified in isolation, which is essential for both systematic evaluation and iterative refinement of the implemented algorithms.

3.5 Chapter summary

This chapter has presented the methodological foundation of the developed acoustic MIMO–OFDM demonstrator. The focus was placed on concrete implementation choices rather than abstract performance claims, covering hardware interfacing, GUI restructuring, and the detailed design of the transmit and receive processing chains.

By combining a unified OFDM front-end with mode-specific MIMO processing and extensive internal observability, the system forms a flexible and reproducible experimental platform. This structure enables systematic variation of parameters and transmission modes while preserving stable and traceable operation.

Based on the methods introduced here, the following chapter presents experimental results obtained with the implemented system, without interpretation. These results are subsequently analyzed and discussed in the context of the design choices made in this chapter.

References

[1] R. v. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Artech House, 2000.

[2] J. G. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 5 ed., 2001.

[3] E. Telatar, "Capacity of multi-antenna gaussian channels," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[4] G. J. Foschini, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–335, 1998.

[5] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[6] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 34–43, 2007.

[7] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Mimo-ofdm for high-rate underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 4, pp. 634–644, 2009.

[8] M. Stojanovic, "Ofdm for underwater acoustic communications: Adaptive synchronization and sparse channel estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[9] L. Häring, "Ofdm – orthogonal frequency division multiplexing," 2025. Lecture slides, Chair of Communication Systems, University of Duisburg-Essen.

[10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[11] "audioplayerrecorder." MathWorks Documentation. Accessed 2026-01-22.

[12] "Audio i/o: Buffering, latency, and throughput." MathWorks Documentation. Accessed 2026-01-22.

[13] "guide (removed) create or edit ui file in guide." MathWorks Documentation. Accessed 2026-01-22.

[14] "Goodbye guide, hello app designer: Evolving your matlab apps." MathWorks Blog, 2025. Accessed 2026-01-22.