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 presents the design and implementation of a multi-channel MIMO-OFDM audio demonstrator intended for teaching and experimental analysis of physical-layer communication concepts. The work focuses on adapting an existing software framework to a modern multi-channel audio interface, restructuring the graphical user interface using MATLAB App Designer, and implementing a configurable MIMO-OFDM receiver chain that exposes relevant intermediate processing results.

The demonstrator enables end-to-end transmission of payload data, such as text, over an acoustic multi-input multi-output channel using spatial multiplexing and OFDM modulation. Frame-based audio playback and recording are employed to ensure stable operation, while receiver processing includes timing synchronization, carrier-frequency-offset compensation, channel estimation, MIMO equalization, symbol demapping, and payload reconstruction.

Beyond payload recovery, the system provides access to a range of physical-layer observables, including synchronization metrics, estimated channel impulse and transfer functions, equalized symbol constellations, channel rank measures, and error vector magnitude (EVM). These quantities allow systematic evaluation of system behavior and facilitate intuitive visualization of MIMO-OFDM principles.

Experimental results obtained with a 4-transmit, 8-receive spatial multiplexing configuration using 4-QAM modulation demonstrate stable operation and reproducible analysis outputs. While channel coding is not yet integrated at the receiver side, the implemented framework provides a solid basis for further extensions and serves as an effective platform for education-oriented experimentation and analysis of MIMO-OFDM systems.

Contents

1 Introduction 4

2 Theoretical Background 5
2.1 Radio Propagation Channels and Motivation for OFDM . . . 5
2.2 Principles of OFDM Transmission . . . 6
2.2.1 Discrete-Time OFDM Signal Model . . . 7
2.2.2 Cyclic Prefix and Circular Convolution . . . 7
2.2.3 Frequency-Domain Representation and One-Tap Equalization . . . 7
2.2.4 Advantages and Limitations of OFDM . . . 8
2.3 Fundamentals of MIMO Systems . . . 8
2.3.1 Motivation for Multi-Antenna Communication . . . 8
2.3.2 Classification of Multi-Antenna Systems . . . 8
2.3.3 Transmission Modes and Gains in MIMO Systems . . . 9
2.3.4 MIMO Channel Model and Matrix Representation . . . 10
2.3.5 Channel Rank and Spatial Degrees of Freedom . . . 10
2.3.6 Summary and Relation to Subsequent Chapters . . . 10
2.4 MIMO-OFDM Signal Model . . . 10
2.4.1 Conceptual Combination of MIMO and OFDM . . . 10
2.4.2 Discrete-Time Transmit Model . . . 11
2.4.3 Per-Subcarrier MIMO Channel Model . . . 11
2.4.4 Structure and Physical Interpretation of the Channel Matrix . . . 11
2.4.5 Detection Problem in MIMO-OFDM Systems . . . 12
2.4.6 Summary and Relation to Subsequent Chapters . . . 12
2.5 Channel Estimation and Equalization in MIMO-OFDM Systems . . . 12
2.5.1 Role of Channel Estimation in MIMO-OFDM . . . 12
2.5.2 Training-Based Channel Estimation . . . 13
2.5.3 Pilot-Based Channel Estimation . . . 13
2.5.4 Frequency-Domain Equalization . . . 13
2.5.5 Linear Equalization: ZF and MMSE . . . 13
2.5.6 Successive Interference Cancellation and V-BLAST . . . 14
2.5.7 Performance Metrics for Receiver Evaluation . . . 14
2.5.8 Summary and Relation to System Evaluation . . . 15

3 Methodology: System Adaptation and Development 15
3.1 System-level workflow and design principles . . . 15
3.2 Objective 1: Adapting the software to the new multi-channel audio interface 16
3.2.1 Hardware platform upgrade and its implications . . . 16
3.2.2 Software-level audio I/O interface and channel mapping . . . 17
3.2.3 Frame-based streaming, underrun/overrun monitoring, and fallback mode . . . 18
3.2.4 Amplitude protection and traceability via audio logging . . . 18
3.3 Objective 2: Porting and restructuring the GUI using MATLAB App Designer . . . 18
3.3.1 Motivation for migration and architectural objectives . . . 18
3.3.2 Event-driven workflow and separation of concerns . . . 19
3.3.3 Global parameter management and state consistency . . . 19
3.3.4 GUI-driven configuration of transmission and reception methods . . 19

3.3.5 Structured parameter interface via `procParam` 20
3.3.6 Methodological implications 20
3.4 Objective 3: Algorithmic design and optimization of the end-to-end PHY pipeline 21
3.4.1 Design objectives and architectural principles 21
3.4.2 Transmit-side processing: payload framing and symbol generation . 21
3.4.3 Control information and header handling 21
3.4.4 Training and synchronization signal design 22
3.4.5 Mode-specific OFDM framing and MIMO mapping 22
3.4.6 Receiver front-end: downconversion, synchronization, and OFDM extraction 22
3.4.7 Channel estimation and equalization 23
3.4.8 Symbol demapping and payload reconstruction 23
3.4.9 Reproducibility and traceability within the processing chain 23
3.5 Chapter summary 24

4 Results **24**
4.1 Overview of reported results 24
4.2 Recorded transmit and receive signals 25
4.3 Timing synchronization results 25
4.4 Channel estimation outputs: CIR and CTF 26
4.5 Equalized symbol-domain results: constellations and EVM 27
4.5.1 Equalized constellations per spatial stream 27
4.5.2 EVM-based quality metrics 28
4.6 Channel rank and conditioning results 28
4.6.1 Rank over subcarriers and rank histogram 29
4.6.2 Minimum singular value and condition-number proxy 29
4.7 End-to-end payload reconstruction and bit error results 30
4.7.1 Recovered text output 30
4.7.2 Bit error count and BER (GUI indicators) 30
4.8 Key metrics summary (tool-reported snapshot) 31
4.9 From results to discussion 31

5 Discussion **32**
5.1 Reliable decoding despite non-ideal symbol quality 32
5.2 Constellation rotation and amplitude contraction 32
5.3 Frame timing uncertainty and CIR alignment 33
5.4 Spatial multiplexing performance and channel rank 33
5.5 EVM as a complementary performance metric 33
5.6 End-to-end payload recovery and uncoded BER 34
5.7 Limitations and opportunities for improvement 34

6 Conclusion **34**
6.1 Summary of achieved objectives 35
6.2 Key outcomes and insights 35
6.3 Limitations of the current implementation 35
6.4 Outlook and future work 36

A Extended result figures for additional operating modes **37**

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.

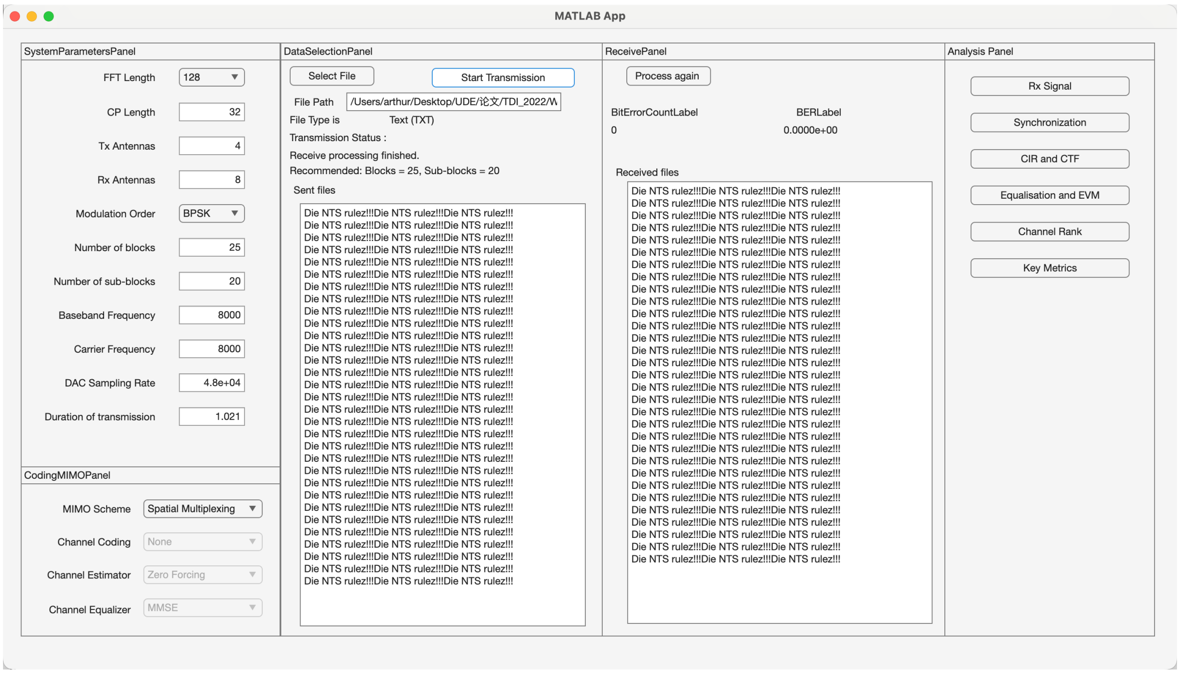
4 Results

4.1 Overview of reported results

Building on the theoretical background and the implementation methodology described in the previous chapters, this project results in a stable and repeatedly executable MIMO audio demonstration that enables end-to-end transmission of payload data (text) over a multi-channel acoustic link. In addition to successful payload recovery, the system provides access to a range of physical-layer observables that characterize the transmission and reception process, including synchronization metrics, channel estimates, equalized symbols, rank-related measures, and error vector magnitude (EVM).

Unless stated otherwise, the results presented in this chapter correspond to the **main operating mode** of the demonstrator, which is configured as follows:

- Transmission mode: **Spatial Multiplexing (SM)**
- Acoustic array configuration: **4 loudspeakers (Tx) and 8 microphones (Rx)**
- Modulation scheme: **4-QAM** ($M = 4$, corresponding to `iModOrd = 2`)
- Channel estimation: **Zero-Forcing** estimator (“ZF”)
- Equalization: **Minimum Mean Square Error** (MMSE) equalizer
- Channel coding: **disabled**



FFT Length	128
CP Length	32
Tx Antennas	4
Rx Antennas	8
Modulation Order	BPSK
Number of blocks	25
Number of sub-blocks	20
Baseband Frequency	8000
Carrier Frequency	8000
DAC Sampling Rate	4.8e+04
Duration of transmission	1.021
MIMO Scheme	Spatial Multiplexing
Channel Coding	None
Channel Estimator	Zero Forcing
Channel Equalizer	MMSE

File Path /Users/arthur/Desktop/UDE/论文/TDI_2022/W

File Type is Text (TXT)

Transmission Status :

Receive processing finished.

Recommended: Blocks = 25, Sub-blocks = 20

Sent files

```
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
```

BitErrorCountLabel	BERLabel
0	0.0000e+00

Received files

```
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
Die NTS rulez!!!Die NTS rulez!!!Die NTS rulez!!!
```

Figure 1: Default UI main interface for 4T8R modulated via 4QAM in SM mode

4.2 Recorded transmit and receive signals

The demonstrator provides direct access to the recorded acoustic transmit and receive signals, enabling inspection of the waveform content at different signal representations. For each microphone channel, the recorded receive signal can be examined over the full recording interval, and—when the corresponding transmit sequence is available—the transmitted loudspeaker signals can be inspected for reference.

In addition to the time-domain representation, the system exposes frequency-domain magnitude spectra for all channels, allowing the occupied bandwidth and channel-dependent spectral shaping introduced by the acoustic path and the audio front-end to be observed. Furthermore, a baseband spectral view centered around the configured carrier frequency is provided by extracting and re-centering the spectral region of interest. This compact representation highlights the effective transmission band used by the MIMO-OFDM signal and serves as a consistency check for modulation, upconversion, and recording.

4.3 Timing synchronization results

Frame timing synchronization is evaluated using a Schmidl–Cox-style repeated training structure. Figure 2 summarizes the synchronization observables obtained for the main operating mode across all eight receive channels (microphones).

For each microphone, three quantities are reported as functions of the delay index d: the magnitude of the correlation sum $|P(d)|$, the corresponding energy term $R(d)$, and the normalized timing metric $M(d)$. These quantities are shown in a consistent three-row layout, with columns corresponding to individual receive channels.

Across all microphones, a dominant peak is observed at a common delay index. This peak is clearly visible in $|P(d)|$ and $R(d)$, and results in a pronounced maximum in the timing metric $M(d)$. The alignment of the peak position across all receive channels indicates that a consistent frame start can be detected despite channel-dependent amplitude

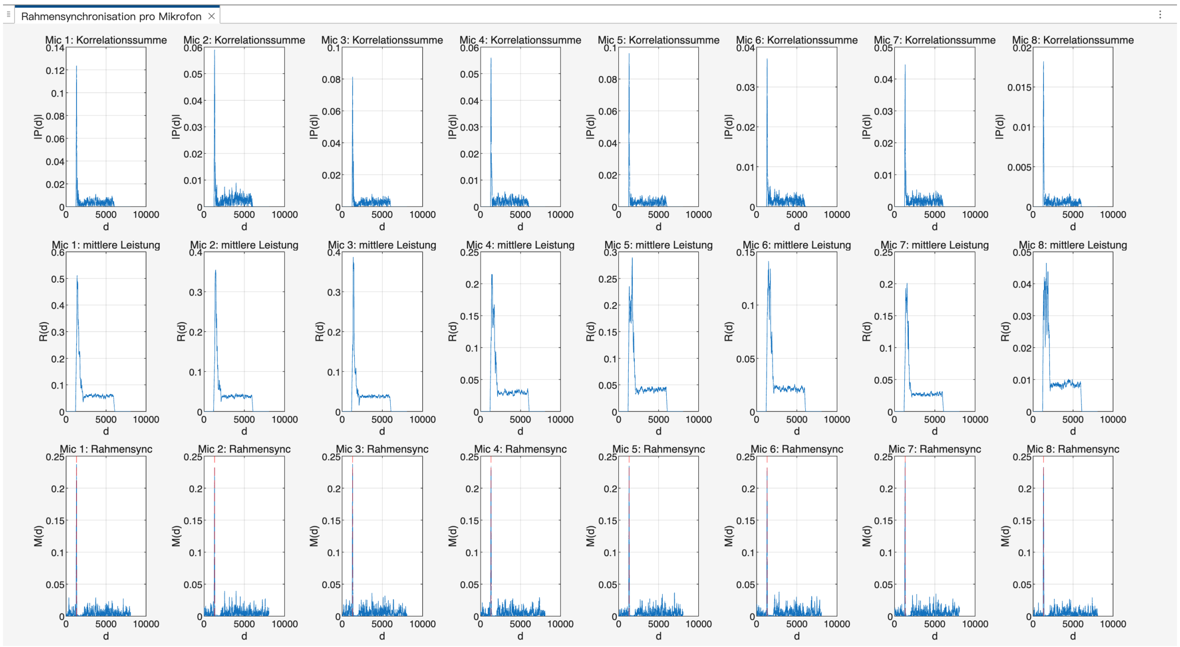


Figure 2: Synchronization observables per receive channel for the main operating mode (SM, 4-QAM, 4 transmit loudspeakers and 8 receive microphones). Top row: correlation magnitude $|P(d)|$; middle row: energy term $R(d)$; bottom row: normalized timing metric $M(d)$.

variations.

While the absolute magnitudes of $|P(d)|$ and $R(d)$ differ between microphones, reflecting different received signal powers and acoustic channel conditions, the location of the synchronization peak remains stable. In the implementation, the detected peak index (`idxMax`) is extracted per channel and used to derive a common frame start for subsequent OFDM block extraction and MIMO processing.

4.4 Channel estimation outputs: CIR and CTF

After frame synchronization and cyclic-prefix removal, the receiver estimates the frequency-domain channel transfer function (CTF) from the preamble symbols. The corresponding time-domain channel impulse response (CIR) is then obtained via an inverse FFT. For each transmit–receive channel pair, both representations are evaluated for the same selected subframe and visualized jointly.

Figure 3 presents the estimated channel responses for the main operating mode. For a fixed transmit channel, the results are shown across all eight receive microphones. The upper row displays the magnitude of the CIR, $|h[n]|$, as a function of the delay tap index, while the lower row shows the magnitude of the CTF, $|H[k]|$, across OFDM subcarriers on a logarithmic (dB) scale.

In the CIR plots, a small number of dominant taps can be observed at low delay indices for all receive channels, followed by a rapidly decaying tail. The relative strength and exact tap distribution vary across microphones, reflecting channel-dependent propagation conditions. For visualization purposes, the CIRs are normalized to enable direct comparison between different receive channels.

The corresponding CTF plots exhibit pronounced frequency selectivity, with deep

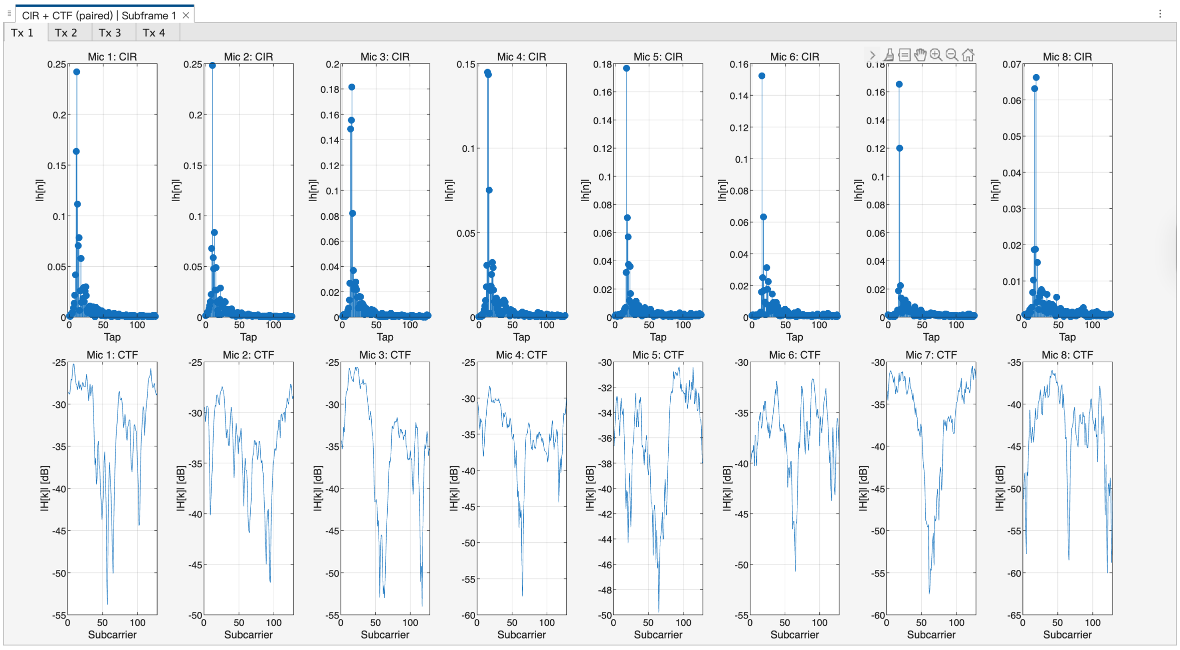


Figure 3: Estimated channel representations for the main operating mode (SM, 4-QAM, 4 transmit loudspeakers and 8 receive microphones). Top row: magnitude of the channel impulse response $|h[n]|$; bottom row: magnitude of the channel transfer function $|H[k]|$ in dB. Results are shown for the same selected subframe and grouped by transmit channel.

notches and peaks that differ between receive channels. These spectral variations are consistent with the multipath structure observed in the CIR representations and illustrate the frequency-dependent nature of the acoustic MIMO channel. Together, the paired CIR and CTF views provide complementary insight into the same estimated channel realization in the time and frequency domains.

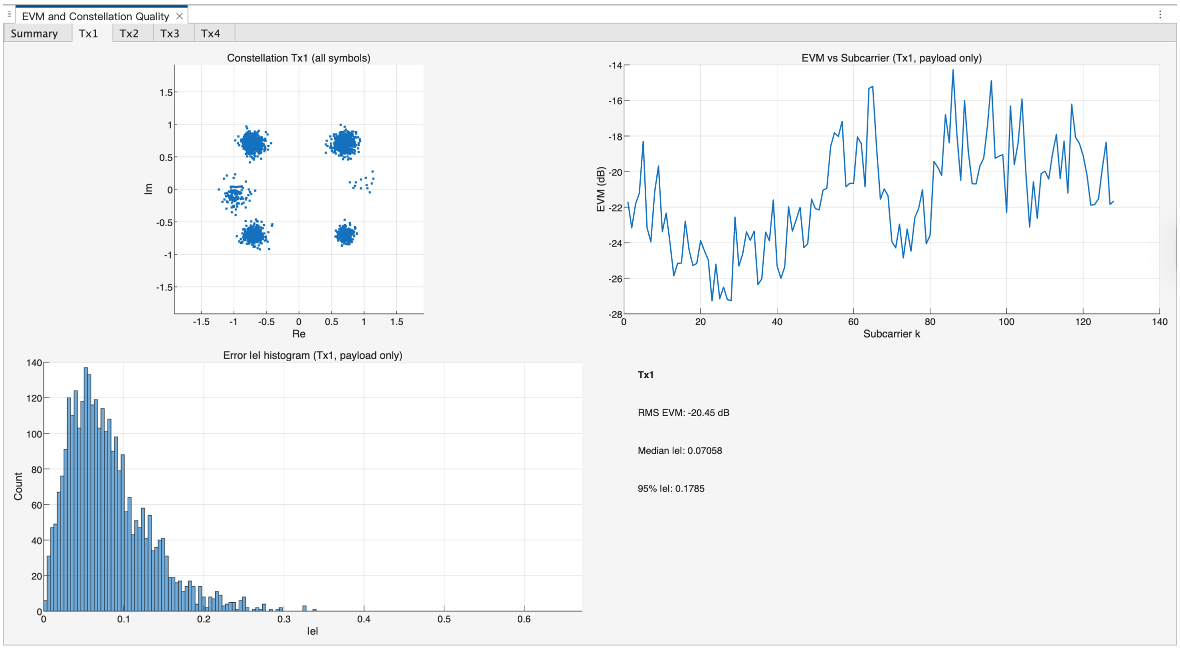
4.5 Equalized symbol-domain results: constellations and EVM

After MIMO detection and linear equalization, the receiver provides access to the equalized complex-valued symbols for each spatial stream. For the main operating mode (SM, four transmit loudspeakers and eight receive microphones, 4-QAM), the results are organized per transmit stream and evaluated in both the complex symbol domain (constellation plots) and the error domain using EVM-based metrics. All results reported in this section are derived from payload symbols only; symbol positions associated with header or control information are explicitly excluded from the EVM computation.

4.5.1 Equalized constellations per spatial stream

Figure 4 shows the equalized symbol constellations for the four spatial streams corresponding to the four transmit loudspeakers. Each constellation aggregates symbols across all active subcarriers and OFDM blocks of the selected frame.

The four QAM clusters are clearly identifiable in all streams, indicating successful separation of the spatial layers by the receiver. At the same time, the apparent spread of the clusters differs between streams, reflecting stream-dependent channel and noise conditions after equalization. The constellations are shown without any post-processing



Tx1

RMS EVM: -20.45 dB

Median |e|: 0.07058

95% |e|: 0.1785

Figure 4: Equalized symbol constellations per spatial stream for the main configuration (SM, four transmit loudspeakers, eight receive microphones, 4-QAM).

such as decision-directed refinement, so the observed dispersion directly corresponds to the output of the linear MMSE equalizer.

4.5.2 EVM-based quality metrics

To quantify the quality of the equalized symbols, the error vector magnitude (EVM) is evaluated with respect to the nearest ideal 4-QAM constellation points. Figure 5 summarizes the EVM results in three complementary representations.

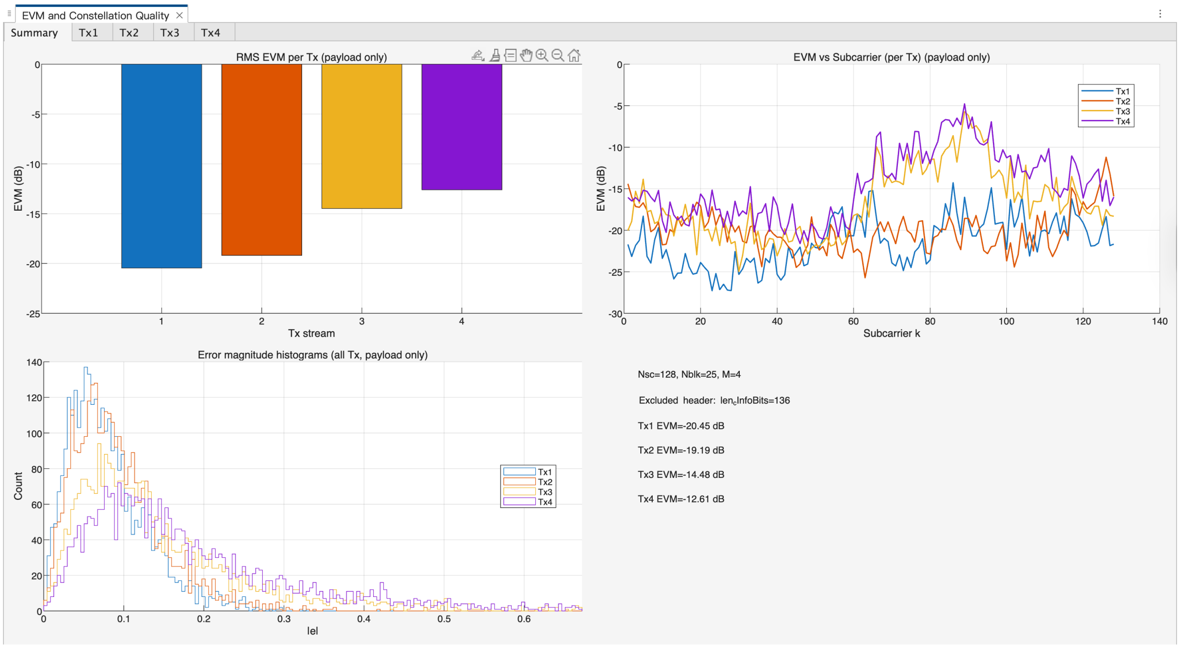
First, the RMS EVM per spatial stream provides a compact scalar measure of symbol quality after equalization. In the shown example, the four streams exhibit different RMS EVM levels, consistent with the stream-dependent dispersion observed in the constellation plots.

Second, the EVM is reported as a function of subcarrier index for each stream. This representation highlights frequency-selective variations across the OFDM band and allows spatial streams to be compared on a per-subcarrier basis.

Third, histograms of the error magnitude $|e|$ illustrate the statistical distribution of symbol errors for each stream. These histograms provide additional insight into the spread and tail behavior of the error distribution beyond a single RMS value.

4.6 Channel rank and conditioning results

To characterize the spatial properties of the estimated acoustic MIMO channel, the demonstrator evaluates rank- and conditioning-related metrics of the frequency-domain channel matrix $H[k]$ for each subcarrier. Figure 8 summarizes these results for the main operating mode.



```
Nsc=128, Nblk=25, M=4

Excluded  header:  len_cInfoBits=136

Tx1 EVM=-20.45 dB

Tx2 EVM=-19.19 dB

Tx3 EVM=-14.48 dB

Tx4 EVM=-12.61 dB
```

Figure 5: EVM evaluation for the main configuration (SM, four transmit loudspeakers, eight receive microphones, 4-QAM): RMS EVM per stream, EVM versus subcarrier index, and error-magnitude histograms (payload symbols only).

4.6.1 Rank over subcarriers and rank histogram

The numerical rank of the channel matrix $H[k] \in \mathbb{C}^{N_r \times N_t}$ is computed for each subcarrier using a singular-value-based criterion with a fixed tolerance. The top-left panel of Figure 8 shows the rank as a function of the subcarrier index.

For the considered configuration with $N_t = 4$ transmit channels and $N_r = 8$ receive channels, the rank is equal to four for all subcarriers. This is further confirmed by the rank histogram shown in the top-right panel, where all observations fall into a single bin centered at rank four. No rank deficiency is observed across the evaluated bandwidth.

4.6.2 Minimum singular value and condition-number proxy

In addition to rank, two auxiliary measures are reported to assess the numerical conditioning of the channel matrices. The bottom-left panel of Figure 8 shows the minimum singular value $\sigma_{\min}(H[k])$ as a function of subcarrier index, expressed in dB. While $\sigma_{\min}(H[k])$ varies across frequency, it remains finite for all subcarriers, consistent with the full-rank observation.

The bottom-right panel shows a condition-number proxy defined as

$$\mathrm{cond}(H[k]) = \frac{\sigma_{\max}(H[k])}{\sigma_{\min}(H[k])},$$

plotted on a logarithmic scale. The condition number exhibits frequency-dependent variations, indicating that although the channel remains full rank, the relative separation between singular values changes across subcarriers.

Together, these plots provide a compact summary of the spatial degrees of freedom and numerical conditioning of the estimated MIMO channel over frequency.

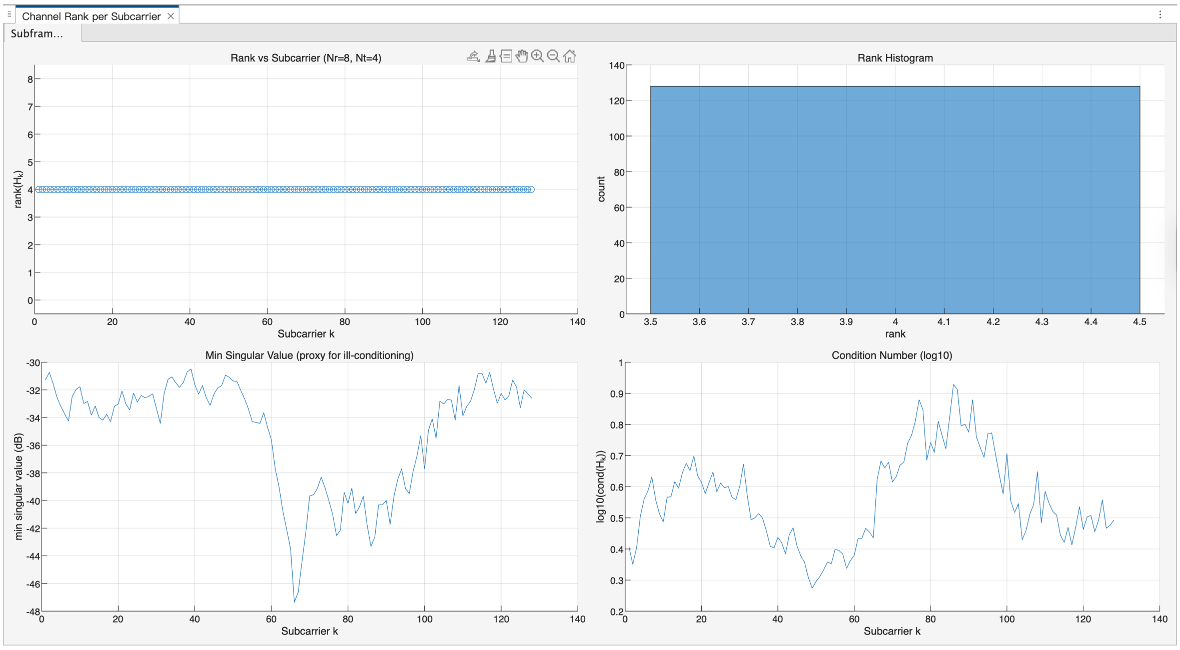


Figure 6: Channel rank and conditioning measures for the main operating mode. Top: rank versus subcarrier index and corresponding rank histogram. Bottom: minimum singular value (in dB) and condition-number proxy versus subcarrier.

4.7 End-to-end payload reconstruction and bit error results

Besides intermediate PHY observables, the demonstrator also provides an end-to-end view of whether the payload can be reconstructed successfully. After demapping, the recovered payload is presented in the GUI (text preview area), and—if the original transmitted payload is available as reference—a bitwise comparison is performed to obtain the bit error count and the corresponding BER. In the main configuration reported in this chapter (SM, 4QAM, no channel coding), these indicators reflect the uncoded link performance.

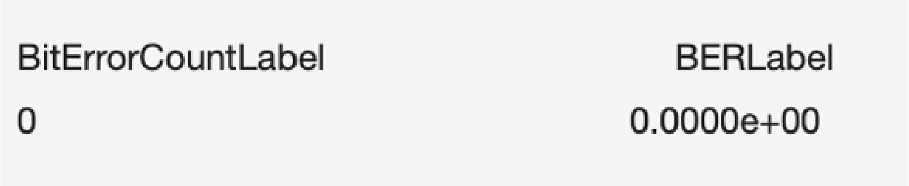
4.7.1 Recovered text output

In text mode, the reconstructed payload is formed by converting the recovered bitstream back to ASCII characters. In the current implementation, this includes (i) optional removal of control/header bits when such a header is configured, (ii) truncation to the known payload length when available, (iii) enforcing byte alignment, and (iv) mapping bytes to characters. The decoded string is displayed directly in the GUI after receiver processing, providing immediate qualitative confirmation of successful reconstruction.

4.7.2 Bit error count and BER (GUI indicators)

When a reference payload is available (i.e., the transmitted text is stored alongside the run), the demonstrator computes the bit error count and BER using `bitFehlerRaten_app`. For text payloads, the function converts both the transmitted and reconstructed strings to 8-bit ASCII sequences via `de2bi(.,8)`, reshapes them into a linear bitstream, and then applies `biterr` on the common prefix length:

$$\mathrm{BER} = \frac{N_{\mathrm{err}}}{N_{\mathrm{comp}}}, \qquad N_{\mathrm{comp}} = \min\{N_{\mathrm{tx}}, N_{\mathrm{rx}}\}.$$



BitErrorCountLabel	BERLabel
0	0.0000e+00

Figure 7: GUI indicators for end-to-end bit error evaluation: bit error count and BER for one representative run.

Table 5: Placeholder for the key-metrics snapshot exported from the demonstrator for one representative run.

Metric	Value
N_{FFT}	(to be inserted)
N_g (CP length)	(to be inserted)
Modulation (M)	(to be inserted)
N_t / N_r	(to be inserted)
Frame start used	(to be inserted)
iSNR (if stored)	(to be inserted)
EVM per stream (if stored)	(to be inserted)
Rank mean/min/max (if stored)	(to be inserted)

The resulting values are presented in the GUI via two numeric indicators (bit error count and BER). Figure 7 shows an example of this UI output for a representative run in which the displayed values are zero.

Implementation note (to be reconciled): Channel coding is currently not enabled in the receiver chain. While a coding mode parameter exists on the transmitter side in the broader project, the reported BER is therefore an uncoded BER. A coded BER evaluation requires consistent integration of receiver-side channel decoding.

4.8 Key metrics summary (tool-reported snapshot)

For rapid inspection and teaching-oriented demonstrations, the GUI provides a consolidated “Key Metrics” view that lists selected parameters and computed metrics from the receiver processing. Typical entries include FFT and CP sizes, modulation order, the number of transmit/receive channels, the selected frame start index, and (when stored) SNR-related quantities, EVM per stream, and rank summary statistics.

Implementation note (to be reconciled): Some metrics are mode-dependent and may be stored under different field names or only in specific receiver branches. The “Key Metrics” view therefore reports available quantities for the selected mode and run.

4.9 From results to discussion

This chapter has presented the set of observable outputs produced by the MIMO audio demonstrator under its main operating configuration. These results establish a concrete basis for further analysis, as they expose the behavior of synchronization, channel estimation, spatial characteristics, and symbol-level quality in a real acoustic MIMO setting. In the following chapter, these observations are examined in more detail and compared across operating modes to discuss their implications for system performance and robustness.

5 Discussion

Chapter 4 reported the observable results obtained from the implemented MIMO audio demonstrator under its main operating configuration. These results demonstrated stable end-to-end transmission and provided detailed insight into synchronization behavior, channel estimation quality, spatial multiplexing characteristics, symbol-domain distortion, and end-to-end payload recovery. The purpose of the present chapter is to interpret these observations by relating them to the concrete implementation choices described in Chapter 3 and to the physical constraints of the acoustic transmission environment.

Rather than evaluating idealized performance limits, the discussion focuses on explaining specific non-ideal phenomena visible in the results, identifying their underlying causes in the signal-processing chain and system setup, and outlining potential directions for improvement.

5.1 Reliable decoding despite non-ideal symbol quality

A central observation in the reported results is that reliable payload decoding is achieved even when symbol-domain quality indicators, such as constellation compactness and error vector magnitude (EVM), are clearly non-ideal. In multiple runs, text payloads are decoded correctly with zero or very low bit error rates, while the corresponding constellation diagrams exhibit noticeable spreading and the measured EVM remains relatively high.

This apparent discrepancy can be directly explained by the structure of the implemented receiver and the chosen modulation scheme. In the main operating mode, 4-QAM modulation is used in combination with hard-decision demapping based on nearest-neighbor detection. As long as the equalized symbols remain within the correct decision regions, moderate distortion does not result in bit errors. Consequently, BER may remain zero even when constellation spreading is clearly visible. This behavior is well known in digital communication systems and highlights that BER alone is an insufficient indicator of link quality, particularly in uncoded systems [2].

The use of MMSE equalization further contributes to this effect. Unlike zero-forcing equalization, MMSE explicitly trades residual interference for reduced noise amplification. As a result, the equalized symbols are not forced onto ideal constellation points, which manifests as increased EVM even when symbol decisions remain correct. In this sense, the observed EVM values are consistent with the chosen equalization strategy rather than an indication of malfunction.

5.2 Constellation rotation and amplitude contraction

Several runs exhibit constellation patterns that are rotated with respect to the ideal reference or show a contraction of symbol clusters toward the origin. These effects can be traced back to specific implementation choices in the receiver chain.

Carrier-frequency-offset (CFO) estimation and compensation are performed during the synchronization stage based on the Schmidl–Cox metric. In the current implementation, this compensation is applied once as a static correction prior to OFDM block extraction. No subsequent pilot-assisted phase tracking or decision-directed phase-locked loop is implemented. As a result, residual phase offsets and slow phase drift can persist, leading to an apparent rotation of the constellation. Such behavior is well documented in OFDM systems when only coarse CFO correction is applied [1].

Amplitude contraction of constellation points can be explained by the interaction between channel estimation, normalization, and MMSE equalization. While channel estimates are normalized for visualization purposes, the equalizer itself does not enforce explicit unit-gain normalization per spatial stream. MMSE equalization preserves relative amplitude scaling across streams and subcarriers, particularly in ill-conditioned channel realizations. Consequently, residual amplitude mismatch remains visible in the equalized symbols, causing the constellation to appear compressed toward the origin.

5.3 Frame timing uncertainty and CIR alignment

The estimated channel impulse responses (CIRs) reveal that the dominant tap position is not perfectly aligned across all runs and subframes, but may shift slightly within the available delay window. This behavior is closely related to the frame-start selection strategy used in the receiver.

Although the Schmidl–Cox timing metric produces clear and distinct peaks for all receive channels, the exact peak indices differ slightly between microphones due to channel-dependent propagation delays and noise. The current implementation selects a single global frame start index based on the earliest reliable detection across channels. This approach ensures consistent OFDM block alignment for all receive channels, but it does not guarantee optimal FFT window placement for each individual channel.

As a result, the FFT window may not be centered identically with respect to the dominant CIR energy for all channels, leading to small shifts in the estimated CIR and corresponding variations in the CTF. More advanced synchronization schemes could mitigate this effect by performing per-channel fine timing adjustment or by selecting the FFT window to maximize energy concentration within the cyclic prefix interval [15].

5.4 Spatial multiplexing performance and channel rank

The channel-rank analysis provides valuable insight into the spatial degrees of freedom available in the acoustic MIMO channel. In the considered configuration with four transmit loudspeakers and eight receive microphones, the estimated channel rank approaches the maximum value of four for a large portion of subcarriers. This indicates that the channel can, in principle, support four parallel spatial streams, which is consistent with the successful operation of spatial multiplexing in the reported experiments.

However, the rank is not uniformly maximal across all subcarriers. Subcarrier-dependent reductions in rank, accompanied by small minimum singular values and elevated condition numbers, indicate locally ill-conditioned channel realizations. These effects are characteristic of frequency-selective MIMO channels and imply that some spatial dimensions are weakly excited at certain frequencies. In such cases, spatial streams become more sensitive to noise and estimation errors, which directly impacts symbol-domain quality and EVM, even when the overall rank remains high [10].

5.5 EVM as a complementary performance metric

The reported EVM results provide a continuous and informative performance measure that complements discrete metrics such as BER. Subcarrier-resolved EVM curves closely mirror the frequency selectivity observed in the estimated CTF, illustrating the strong link between channel frequency response and symbol-domain distortion.

The exclusion of header symbols modulated with BPSK from the EVM computation improves interpretability by ensuring that the reported values reflect the quality of the QAM payload symbols only. This distinction is particularly important in mixed-modulation systems, where header and payload symbols are subject to different detection characteristics.

Overall, the EVM analysis confirms that while the system operates reliably at the bit level, significant symbol-domain distortion remains, leaving clear room for improvement through enhanced channel estimation, phase tracking, and coding.

5.6 End-to-end payload recovery and uncoded BER

The successful reconstruction of text payloads demonstrates the internal coherence of the implemented receiver chain across all processing stages, from synchronization and channel estimation to equalization and demapping. When reference payload data is available, the reported bit error counts and BER values reflect the combined effect of channel conditions, MIMO detection, and the absence of channel coding.

While uncoded transmission limits the achievable error performance, the results confirm that reliable payload recovery is feasible under the chosen operating conditions. This outcome is consistent with the demonstrator's primary objective of illustrating MIMO-OFDM concepts rather than achieving near-capacity performance.

5.7 Limitations and opportunities for improvement

Several limitations of the current implementation are intentionally acknowledged. Although channel coding options exist at the transmitter side, receiver-side channel decoding is not yet integrated in a unified manner. As a result, coding gain is not realized, and BER results correspond to uncoded transmission only.

Furthermore, auxiliary metrics such as CFO estimates and SNR-related quantities are not consistently stored across all processing branches, reflecting the evolutionary nature of the software development. From a system-design perspective, integrating pilot-assisted phase tracking, refined synchronization strategies, and receiver-side channel decoding would constitute natural next steps toward improved performance and completeness.

Finally, it must be emphasized that the acoustic transmission medium itself imposes fundamental constraints. Loudspeaker directivity, microphone placement, partial obstruction, and room reflections introduce spatial asymmetries and time variations that are difficult to control. These effects limit the extent to which ideal MIMO-OFDM behavior can be reproduced and must be considered an inherent part of any practical MIMO-audio demonstrator.

6 Conclusion

This thesis addressed the development of a MIMO-OFDM audio demonstrator with the dual purpose of enabling hands-on experimentation and supporting teaching-oriented analysis of modern physical-layer concepts. Starting from an existing software framework, the work focused on adapting the system to new hardware constraints, restructuring the user interface, and extending the receiver processing chain to expose meaningful and interpretable intermediate results.

6.1 Summary of achieved objectives

The thesis objectives defined in Chapter 1 have been met as follows.

Objective 1: Adaptation to a multi-channel audio interface. The demonstrator was successfully adapted to operate with a modern multi-channel audio interface, supporting simultaneous playback and recording across multiple loudspeaker and microphone channels. A stable streaming setup was achieved using frame-based audio I/O, with explicit handling of buffering, latency, and synchronization constraints. The resulting system enables repeatable transmission and reception of broadband OFDM waveforms in an acoustic environment, forming a reliable experimental basis for further signal processing and analysis.

Objective 2: Migration and restructuring of the graphical user interface. The legacy GUIDE-based GUI was migrated to MATLAB App Designer and restructured to clearly separate user interaction from signal-processing logic. The resulting application provides intuitive configuration of physical-layer parameters, mode selection, and execution control, while exposing a rich set of visualization and analysis tools. This modular design improves maintainability and extensibility, and aligns the software structure with current MATLAB development practices.

Objective 3: Implementation and analysis of a MIMO-OFDM receiver chain. A configurable receiver processing pipeline was implemented, including timing synchronization, carrier-frequency-offset compensation, channel estimation, MIMO equalization/detection, symbol demapping, and payload reconstruction. The receiver supports spatial multiplexing operation with multiple estimation and equalization strategies and exposes intermediate quantities such as synchronization metrics, CIR/CTF estimates, equalized symbols, channel rank, and EVM. These outputs allow systematic evaluation of MIMO-OFDM behavior and form the core analytical contribution of the demonstrator.

6.2 Key outcomes and insights

The experimental results demonstrate that the implemented system operates stably in its main configuration (spatial multiplexing, 4T8R, 4-QAM, zero-forcing channel estimation, MMSE equalization, no channel coding) and reliably supports end-to-end text transmission. Beyond payload recovery, the system provides a coherent set of physical-layer observables whose behavior is consistent with theoretical expectations for frequency-selective MIMO channels.

In particular, the combination of channel estimation, rank analysis, and symbol-domain quality metrics illustrates how spatial multiplexing performance varies across subcarriers and spatial streams. The availability of such intermediate results is especially valuable in an educational context, where understanding system behavior is often more important than optimizing absolute performance figures.

6.3 Limitations of the current implementation

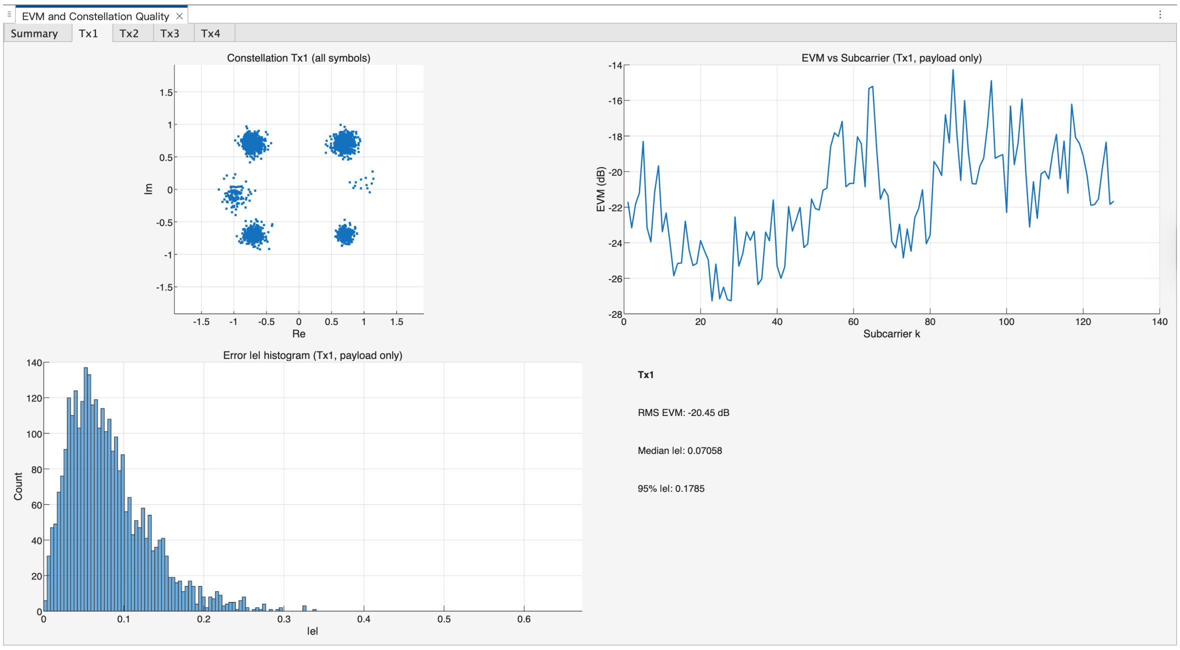
Several limitations of the present implementation have been identified. Although channel coding options exist conceptually, receiver-side channel decoding is not yet integrated in a consistent manner, and the reported BER results therefore correspond to uncoded transmission. In addition, certain auxiliary metrics (e.g., consolidated CFO or SNR reporting) are not uniformly stored across all processing branches, reflecting the incremental development process.

These limitations do not compromise the validity of the reported results but indicate areas where further consolidation and refinement would improve completeness and comparability.

6.4 Outlook and future work

Future work could extend the demonstrator in several directions. Integrating channel decoding would enable systematic coded performance evaluations and closer alignment with standard communication-system models. Additional MIMO transmission schemes, such as Alamouti and eigenmode transmission, could be analyzed more comprehensively using the existing framework. Finally, enhanced automation of measurement logging and result export would facilitate larger experimental studies and further strengthen the demonstrator's value as a teaching and experimentation platform.

Concluding remark. Overall, this thesis demonstrates that a carefully designed audio-based MIMO-OFDM demonstrator can serve as an effective bridge between theoretical communication concepts and practical experimentation. By emphasizing transparency, stability, and analytical accessibility, the developed system provides a solid foundation for both education and further research-oriented extensions.



Tx1

RMS EVM: -20.45 dB

Median |e|: 0.07058

95% |e|: 0.1785

Figure 8: Channel rank and conditioning measures for the main operating mode: rank versus subcarrier and rank histogram (top), minimum singular value and a condition-number proxy versus subcarrier (bottom).

A Extended result figures for additional operating modes

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.

[15] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.